License Use Management

Using License Use Management Runtime for Windows

Version 4.6

SH19-4347-03

License Use Management

Using License Use Management Runtime for Windows

Version 4.6

SH19-4347-03

Note

Before using this information and the product it supports, be sure to read the general information under “Notices”.

ISO 9001 Certification

This product was developed using an ISO 9001 certified quality system. Certification has been awarded by Bureau Veritas Quality International (BVQI), (Certification No. BVQI-92053). BVQI is a world leader in quality certification and is currently recognized by more than 20 accreditation bodies.

Fourth Edition (November 2000)

This major revision obsoletes and replaces SH19-4347-02. The major changes are described in ″Summary of Changes″. In the hard copy version of this book, technical changes are marked by a vertical line in the left margin. In the .HTM version, post-publication technical changes appear in brown.

This edition applies to Version 4.6.0 of IBM License Use Management Runtime for Windows, a part of License Use Management for Windows, Program Number 5697-B23, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments.

Address your comments to:

License Use Management Information Development
Rome Tivoli Lab
IBM Italia S.p.A.
Via Sciangai, 53
00144 Rome
Italy

Fax Number : (+39) 06 5966 2077

Internet ID: ROMERCF at VNET.IBM.COM

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Chapter 3. Installing and Maintaining LUM Runtime 49

Chapter 4. Configuring LUM Runtime . . . 69

Chapter 5. Daily Tasks 95

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
P.O. Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709-2195
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
IBM
NetFinity
SystemView
VisualAge
OS/2

LicensePower and iFOR are registered trademarks of Isogon Corp.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the U.S. and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Other company, product, and service names may be trademarks or service marks of others.

About This Book

This book describes License Use Management Runtime for Windows (*LUM Runtime*). It contains information about how to set up:

- The servers where product licenses are to be installed
- The clients that use the products
- An environment that can support network licensing

It also explains how to manage the licenses for the products you install.

Before reading this guide, read the documentation supplied with the licensed product.

Who Should Use This Book

This book is intended for:

- The system administrator who is responsible for setting up and administering the license management environment. This person must have general knowledge of:
  - Windows
  - The NetBIOS, TCP/IP, or IPX transport protocol
  - The network environment
- The end users who run the software products and the LUM Runtime product. End users require only general knowledge of the Windows operating environment.

How This Book is Organized

This book is organized into the following sections:

Chapter 1. Introduction provides an overview of License Use Runtime features and benefits, describes supported license types, license policies, and license-enabling models, and presents some simple scenarios of the use of licensed products.

Chapter 2. Planning Network Licensing provides the basic NCS concepts you need to set up your network and manage licenses.

Chapter 3. Installing and Maintaining LUM Runtime explains how to install, uninstall, and upgrade License Use Runtime components.

Chapter 4. Configuring LUM Runtime provides scenarios for configuring License Use Runtime, administering product licenses, and exercising customer-controlled policies, using the License Use Runtime tools.

Chapter 5. Daily Tasks describes daily tasks, such as administering product licenses and exercising customer-controlled policies.

Chapter 6. Command Line Interface documents the License Use Runtime command line interface (CLI).

Chapter 7. Tips and Hints explains how to better take advantage of License Use Runtime.

Chapter 8. Troubleshooting explains how to improve performance and handle problems, should they arise when you use license-enabled products.

Appendix A. LUM Runtime Configuration File provides reference information on the configuration file.

Appendix B. Using the Nodelock File explains how to edit a nodelock file.

Appendix C. Operating Systems and License Use Management Support identifies the operating systems supported by License Use Management and, where appropriate, the conditions of their support.

The glossary explains terms used in this manual.

Where to Find More Information

This section lists other sources of information for LUM Runtime.

README File

For changes to LUM Runtime that were made after the book was finished, see the README.ARK file in the download package from the LUM Web site.

HTML Files

To view an HTML file, open it in your Web browser.

The files are in the directory:

```
\ifor\win\bin\en_US
```

There are HTML versions of:

- LUM Runtime for Windows Message Reference:

  `lummsg.htm`
- This book (*Using License Use Management Runtime for Windows*):

```
lumusg.htm
```

A PDF version of this book is available for download from the IBM LUM Web site.

- *Using Application Developer's Toolkit* (if Application Developer's Toolkit is installed in your environment):

  ```
  lumtkt.htm
  ```

  This book describes how to license-enable software products for use with License Use Management.

  A PDF version of this book is available for download from the IBM LUM Web site.

LUM Web Site

For information and news about IBM License Use Management, and to download LUM Runtime publications and code, visit the IBM LUM Web site at:

```
http://www.software.ibm.com/is/lum
```

Notational Conventions

This book uses the following notation in text:

Bold Bold print indicates something you click, select, or type, such as a menu option, field, or push button.

Italics Italics are used for variables, for new terms in the text, and for emphasis.

`Monospacing` Monospacing is used for system messages and examples.

Summary of Changes

The fourth edition of this book incorporates late changes to Versions 4.5.5 through 4.5.9 of LUM Runtime, which were documented in the README.ARK and README.JCS files for those versions.

This section summarizes the changes made to this book to reflect new and changed function and support.

Additional operating systems supported
LUM Runtime supports the additional operating systems:

- IBM AIX Version 5.0 for IA-64
- Microsoft Windows 2000
- Sun Solaris 2.8

Operating systems no longer supported
OS/2 and Windows NT for Alpha are no longer supported.

New application programming interface for Java applications
A new application programming interface enables Java clients to interact directly with the license server without using the Java Client Support.

Administering groups of licenses
You can group license-enabled products and administer the group as a unique entity. Such a group is called a ″bundle.″

Pricing products by system capacity
Product pricing can be based on the number of users or on the characteristics of the system on which the products run, such as the number of processors or disks.

Displaying licenses and license user databases
You can run the basic license tool to display lists (`i4blt -l`) and product license status (`i4blt -s`), and can generate reports (`i4blt -r`) without having administrator privileges. These options are also available through the equivalent GUI actions.

Detecting a nonresponding server
You can modify the amount of time that a license-enabled application is to wait before a connection is lost.

Automatically cleaning up stale licenses
Stale licenses on configured servers are automatically cleaned up at start-up and during each refresh. The automatic clean-up is triggered by a configurable tag in the `i4ls.ini` file.

Testing high-availability licensing clusters

You can test a high-availability licensing (HAL) cluster that has been configured and activated by means of the `i4tv` command. This command tries to contact all servers that are in serving state, to request all 10 000 licenses of the HAL Test Product.

Improved cluster reliability

When a network fails and is then resumed, any cluster it contains is automatically restarted after a brief interval.

Improved reliability of license server databases

To improve the reliability of license server operations, the structure of the server databases has been modified, so that license information is separated from user information.

Annotation extension for simple nodelocked licenses

The length of the annotation field for simple nodelocked licenses enrolled in the nodelock file is extended to 255 characters.

Defining multiuse rules for the same accessor ID

Vendors can define their own multiuse rule based on a generic string that identifies the requestor of the license.

Enrolling multiple license certificates simultaneously

Using the basic license tool, you can now enroll more than one license certificate at a time.

Product-specific program

Vendors can specify in the license password the name of a program that is to run automatically when a license is enrolled.

Languages supported

LUM Runtime 4.6.0 supports only U.S. English. LUM Runtime 4.5.5 information files are not compatible with LUM Runtime 4.6.0.

Chapter 1. Introduction

License Use Management Runtime (referred to here as *LUM Runtime)* is part of IBM License Use Management, a combination of tools for software asset protection. The License Use Management tools enable software vendors and their customers to ensure that customers comply with the terms and conditions of license agreements. They check compliance through runtime monitoring of the usage of software assets.

License Use Management at a Glance

License Use Management consists of two products:

License Use Management Application Developer's Toolkit (LUM ADK)
Enables vendors to implement license management in an application. To implement license management, vendors embed API calls and API code in their products. The product is then described as *license-enabled*. Vendors can license-enable C-language programs, Java applications, and Java applets.

The LUM ADK offers the vendor great flexibility in:

- Level of control exercised by the license-enabled application
- Type of customer licensing environment for which the application is intended
- Implementation of policies

The LUM ADK is a priced product of IBM. The vendor who acquires the kit receives a copy of the License Use Management software, and gets royalty-free rights to redistribute License Use Management Runtime within the license-enabled application.

LUM Runtime
Enables users of license-enabled software to manage the licensing environment. The LUM Runtime software is free of charge and is available for download from the IBM License Use Management Web site:

`http://www.software.ibm.com/is/lum`

License Use Management Vendor Perspective

License Use Management benefits software vendors by enabling them to:

- Ensure that customers use software licenses within entitled limits
- Base product prices on actual usage
- Protect intellectual property from unauthorized use
- Increase overall revenue as customers acquire all the licenses they need

- Distribute software for a trial period with trial licenses that can be replaced by production licenses, thus minimizing distribution cost

License Use Management Customer Perspective

License Use Management benefits the customers of software vendors by enabling them to:

- Ensure that they have enough licenses to satisfy their business requirements and, at the same time, that they are not paying for more licenses than they need
- Base software charges within the enterprise on actual usage
- Demonstrate license use compliance to internal and external auditors
- Protect organizations from inadvertent violations of license agreements
- Change software assets to alternative pricing policies that the vendor offers

Basic Concepts of License Use Management

A *license*, in the context of License Use Management, is permission to use an instance of a licensed software product or service, according to the basis on which the vendor charges for the product or service. The term *license* in this context does not refer to the license agreement that governs use of and rights to a product.

License-Enabling a Product

Vendors deliver licenses to customers in the form of a *license password*. The password contains encrypted terms of the usage of the software product, such as:

- The number of licenses or concurrent copies of the product the customer can use
- The expiration date of the licenses
- The license type (see “License Types” on page 7)

In the license-enabling process, the vendor can:

- Select among the various license types
- Decide to distribute licenses one-by-one or in packages of multiple licenses from which individual licenses can be extracted (see “License Passwords” on page 5)
- Implement direct controls over the use of licenses, or make it possible for the customer to control use of licenses (see “Use-Control Levels” on page 5)
- Impose, or allow the user to impose, various types of control over administration of licenses (see “License Policies” on page 12)

When the product is installed, LUM Runtime checks that a license is available each time the product is run.

Nodelocked and Network Licenses

Vendors can supply licenses in two forms: *nodelocked* licenses or *network* licenses.

A *nodelocked* license is a license for a specified workstation (node). It is stored on the specified workstation, and the license-enabled product will run on only that workstation.

A *network* license is a license, stored on a server, that can be used by any client workstation. Many LUM Runtime clients can share the licenses for enabled products. The licenses are stored on one or more network license servers. Each client workstation must be connected to a server. When the user at a client starts a licensed program, LUM Runtime at the license server determines whether a license is available.

License-enabled Java products can have network licenses or nodelocked licenses. For network licenses that use Java Client Support, a Web server computer, rather than the end-user workstation on which the application or applet runs, can also serve as a client. See “Planning for Java Products that Use Java Client Support” on page 42.

Managing Licenses

LUM Runtime includes the *basic license tool*, which manages both nodelocked and network licenses. The basic license tool enables you to:

- Add licenses to or delete licenses from the server database
- Display information about the licenses installed
- Distribute the licenses among the license servers available on the network
- Reserve licenses for the exclusive use of certain users
- Generate reports on license usage and server events

The basic license tool has a graphical user interface (GUI) and a command-line interface (CLI). For more information about what the basic license tool does, see “Chapter 5. Daily Tasks” on page 95.

LUM Runtime Platforms

Table 1 shows which platforms LUM Runtime supports, and where to get the LUM Runtime code.

Table 1. LUM Runtime Platforms

AIX 5 for IA-64	LUM Runtime 4.6 base code is part of the base operating system, and is installed on every workstation when the operating system is installed. You can install Version 4.6 from the License Use Management CD-ROM or from the product package downloaded from the IBM LUM Web site: `http://www.software.ibm.com/is/lum`

AIX 4.3.3
AIX 4.3.2
AIX 4.3.1
AIX 4.3.0
	The following levels of LUM Runtime are part of their associated level of the AIX operating system. They are installed on each workstation when AIX is installed.
	`AIX Level      LUM Runtime Level`
	`4.3.3          4.5.5`
	`4.3.2          4.5.0`
	`4.3.1          4.0.1`
	`4.3.0          4.0.0`
	Optional packages and filesets can be installed from the AIX installation media.
	To upgrade to LUM Runtime Version 4.6, download the code from the IBM LUM Web site:
	`http://www.software.ibm.com/is/lum`

Windows® 2000 Windows NT® 4.0 Windows NT Server 4.0, Terminal Server Edition Windows NT Server 4.0 Windows 98 Windows 95 Sun Solaris 2.6, 2.7, and 2.8 HP-UX 10.20 and 11.0 Silicon Graphics IRIX 6.3, 6.4, and 6.5	LUM Runtime Version 4.6 can either be redistributed with the license-enabled product or be downloaded from the IBM LUM Web site: `http://www.software.ibm.com/is/lum`

License Passwords

Vendors provide license passwords (also known as *license keys*) with their products.

A license password is an encrypted character string. It contains information about the license, such as:

- The type of license (see “License Types” on page 7)
- The date when the licenses become active
- The date when the licenses expire

There are two types of password: simple and compound. A *simple password* represents one or more licenses. A *compound password* contains multiple simple passwords, each of which represents one or more licenses.

Note: The expiration date of simple licenses that are contained in a compound password must fall on or before the expiration date of their compound password.

Compound passwords can be used as follows:

- Vendors use compound passwords to distribute multiple licenses to their customers.
- The customer installs the compound password on the assigned license server. The customer extracts the licenses and distributes them, as required, to different license servers. (These servers are not specified in the compound password.)
- A sales representative extracts the licenses from the compound password and distributes them to customers.

The vendor includes the password, along with other information about the application, in the *enrollment certificate file* (ECF). You then make licenses available to users of the product by *enrolling* the product using the license password. See “Enrolling the Product” on page 97.

Use-Control Levels

Products can have one of the following predefined use control levels:

- Vendor-managed use control
- Customer-managed use control

Vendor-Managed Use Products

With vendor-managed use (VMU) products, the vendor manages compliance with the terms of the acquisition of the software product.

When you request licenses for a vendor-managed use product, the product vendor asks you to supply the unique identifier (target ID) of each workstation where you intend to install the product licenses. For nodelocked licenses, this is the identification of the workstation where the enabled product is to be installed; for network licenses, this is the network license server. You must also supply the number of licenses that you want to acquire. The vendor uses this information to create the password that you use to install and activate the licenses.

Vendors ship a vendor-managed use product typically with a simple password. They can also use compound passwords for this purpose.

If you want to change the terms and conditions of the contract (for example, to increase the number of licenses), you provide the vendor with similar information for each of the workstations on which you intend to install the licenses and get a new password.

A vendor can produce a vendor-managed use license password with target ID set to ANY, so that you can install it on any workstation. The vendor can deliver such a password with the product package without your specifying how many licenses you want to acquire. Instead, the vendor sets an upper limit, possibly very large, on the number of licenses you can install on each license server. By generating and delivering such a password, the vendor has decided not to perform the checks that are normally associated with vendor-managed use.

Customer-Managed Use Products

To provide vendors with greater flexibility in the way they deliver licensed software, LUM Runtime supports customer-managed use (CMU) control. With products that are enabled in this way, the vendor does not directly associate licenses with a particular license server (or group of license servers). The vendor does not set an upper limit on the number of licenses that you are entitled to use. Instead, it is your responsibility to set that upper limit, depending on the terms of the software product acquisition.

LUM Runtime provides you with the information on the usage of the enabled products, thereby helping you stay within the boundaries of the acquisition agreement. Transactions, such as enrollment, distribution of licenses, updates, and deletions, are logged in a tamper-proof LUM Runtime database.

Vendors typically ship a customer-managed use product with a compound password that you can use to extract and distribute the number of licenses you have acquired. They can also use simple passwords for certain types of licenses.

License Types

This section describes the types of license the vendor can select. When you receive a license-enabled product, check the product documentation to determine the license type.

Nodelocked Licenses

A nodelocked license restricts the use of a product to the workstation specified in the license. Vendors typically use nodelocked licenses for standalone applications, rather than for client/server applications.

For nodelocked licenses there are two license-enabling models: *non-runtime-based* and *runtime-based*. If a vendor chooses non-runtime-based enabling, the license-enabled product itself, rather than LUM Runtime manages use of the nodelocked license. The password for such a product is stored in a *nodelock file*. When you start the application, it checks the nodelock file to ensure you have a valid license. If the vendor chooses runtime-based enabling, management of the nodelocked license is performed by the nodelocked license server on the local workstation. You then use the basic license tool to view and update information about the nodelocked licenses. See “License-Enabling Models” on page 16 for more information about license-enabling models.

Vendors can enable their products to use the following kinds of nodelocked licenses:
- Simple nodelocked licenses
- Concurrent nodelocked licenses
- Use-once nodelocked licenses
- Per-server licenses

Simple Nodelocked Licenses

A simple nodelocked license allows an unlimited number of simultaneous uses of the licensed application on the local workstation. Simple nodelocked licenses are valid only for vendor-managed use products. A word processor is a typical example of a product that uses nodelocked licenses.

Concurrent Nodelocked Licenses

The concurrent nodelocked license is local to the node where the application has been installed. It allows a limited number of simultaneous uses of the licensed application. A typical example of a concurrent nodelocked license is a client/server application. The application server is able to recognize the number of clients connected to it and ask for a license for each of them.

Vendors can use concurrent nodelocked licenses for both vendor-managed and customer-managed products.

When you enroll a customer-managed product, you must specify how many concurrent nodelocked licenses you have acquired for the product. The administrator can modify this number at any time.

Use-Once Nodelocked Licenses

A use-once nodelocked license permits a single use of a licensed product on a particular workstation. Every time the product is started, one license is consumed.

A typical use of use-once nodelocked licenses is to distribute promotional or demonstration versions of software.

Vendors also provide use-once nodelocked licenses to supplement concurrent nodelocked licenses during times when user demand for those products exceeds the number of available concurrent nodelocked licenses.

Vendors can use use-once nodelocked licenses for both vendor-managed and customer-managed products.

When you enroll the licenses for a customer-managed product, you must specify how many use-once nodelocked licenses you have acquired for the product. The administrator can modify this number at any time.

Per-Server Licenses

Per-server licenses are similar to concurrent nodelocked licenses, except that at any time, you can change them into per-seat licenses (see “Scenario 8: Switching from Per-Server to Per-Seat Licenses” on page 108).

Vendors use per-server/per-seat licenses to enable client/server applications constructed for multiple-server solutions. With both per-server and per-seat licenses, the server of a licensed client/server application can request licenses for its clients. The application clients need not be license-enabled.

With per-server licensing, each application server license is associated with a specific number of application clients. This represents the maximum number of application clients that may concurrently request services from that application server. The application client licenses are stored locally on the application server workstation and are granted temporarily to requesting application clients. Multiple application servers grant licenses independently of one another; if the same application client connects to more than one application server, the application client is granted more than one license. You should therefore probably use per-server licenses only in an environment where:

- Each application client connects to only a single application server, or
- Each application client uses the application infrequently for brief periods

When your environment grows in such a way that application clients are connecting to multiple application servers, you can convert your per-server licenses to per-seat. With per-seat licensing, unused application client licenses are kept in a central repository, which all the application servers share. They also share a central list of application clients to which a license has been assigned. When a license is assigned to an application client, the license remains assigned to the application client even when it is not using the product. If an application client connects to multiple application servers, it is assigned only one license.

Per-server licenses are valid only for customer-managed use products.

Network Licenses

Network licenses, rather than being restricted to a single workstation, are stored on a network license server and shared among multiple network license clients.

Vendors can enable their products to use the following kinds of network licenses:

- Concurrent licenses
- Reservable licenses
- Use-once licenses
- Per-seat licenses

Concurrent Licenses

A concurrent license is a network license that can be temporarily granted to run the licensed application on a client.

When the product is running, that license remains unavailable to other users of the product. When the product stops running, the license is returned to the server, where it becomes available to other users.

Concurrent licenses allow as many users to run a licensed application simultaneously as there are valid licenses for the product available from the network license servers in your licensing environment.

A typical use of concurrent licenses is for products with relatively expensive licenses that each user will use only some of the time. The customer orders fewer licenses than there are users to optimize use of the licenses. Such applications may be either client/server applications, for which the client is enabled, or non-client/server applications.

Vendors can use concurrent licenses for both vendor-managed and customer-managed products.

Reservable Licenses

A reservable license is a network license that you can reserve for the exclusive use of a user, a group, or a node. The reservation is for a specified time period. A reservable license that has been reserved is called a reserved license. A reservable license that has not been reserved is called an unreserved license.

When a reserved license is granted from the network, the license is stored on the workstation where the licensed application is running. Thereafter, the license can be used on the workstation, even if the workstation is disconnected from the network, until the reservation expires.

A typical use of reservable licenses is for the client part of a client/server application that is likely to run on a portable computer that is often disconnected from the network. Another typical use is for a compiler being used in software development. During a build process involving many compilations, it is more efficient to reserve a compiler license for a day or two than to make a separate request for a compiler license for every compilation

You can reserve some of the reservable licenses for an application and leave others unreserved. Unreserved licenses are treated like concurrent licenses.

Vendors can use reservable licenses for both vendor-managed and customer-managed products.

Use-Once Licenses

A use-once license is a network license that permits a single use of a licensed product during the time the license is valid. Every time the product is started, one license is consumed.

A typical use of use-once licenses is to distribute promotional or demonstration versions of software.

Vendors also provide use-once licenses to supplement concurrent licenses when user demand for those products exceeds the number of available concurrent licenses.

Vendors can use use-once licenses for both vendor-managed and customer-managed products.

Per-Seat Licenses

Vendors use per-server/per-seat licenses to enable client/server applications constructed for multiple-server solutions. With both per-server and per-seat licenses, the server of a licensed client/server application can request licenses for its clients. The application clients need not be license-enabled.

For per-seat licensing, unused application client licenses are kept in a central repository, which all the application servers share. They also share a central list of application clients to which a license has been assigned. When a license for a C or C++ application or for a Java application that uses the Java native interface is assigned to an application client, that assignment is permanent. If an application client connects to multiple application servers, it is assigned only one license.

You will probably want to use per-seat, rather than per-server, licenses in an environment where application clients connect to multiple application servers. (See also “Per-Server Licenses” on page 8.)

Per-seat licenses are valid only for customer-managed use products.

License Policies

Vendors can enable their products to implement various policy decisions regarding how licenses are managed.

Vendor Controlled Policies

The vendor can implement the license policies: capacity, try-and-buy, multiuse rules, product wait queues, license annotation, custom configuration, bundle, and product-specific program.

Capacity

With this policy vendors can comply with the terms and conditions of products that are priced based on certain characteristics of the system on which the product is running, not only on the number of users who are using the product. This includes, for example, the number of processors on which the application can run or the maximum size of the physical memory.

In such cases the vendor can specify a “capacity type” that defines the unit of measure for the internal counter (“capacity units”) associated to the license password. The capacity type is an integer value between 1 and 255; some of these values have a predefined value, others may have any vendor specific meaning.

The current predefined capacity types are:

- Online processors
- Configured processors
- Physical disks
- Physical memory (in MB)

When you use one of the predefined capacity types, the LUM library embedded in the license application can calculate the number of capacity units that are consumed by the application itself. In the other cases the vendor must calculate them before passing the results to the LUM library. In either case, the license is granted only if the capacity type of the password matches the capacity type specified in the request, and the capacity units are not exceeded. This applies to simple nodelock, concurrent nodelock, and concurrent network license types, under either vendor-managed or customer-managed use control.

Note: Not all the above capacity types are supported on all the platforms (see *Using License Use Management Application Developer’s Toolkit*).

Try-and-Buy

The vendor can enable a product with a special simple nodelocked license for customers to use during an evaluation period. The evaluation period (with duration set by the vendor) starts either when the product is enrolled or when the product is run for the first time.

Multiuse Rules

Multiuse rules define the conditions under which multiple invocations of a product require only a single license. These rules are applicable only to concurrent network, concurrent nodelocked, and per-server licenses.

The vendor can enable a product so that after a license has been granted to a particular user, group, node, job ID, or accessor ID, a second invocation of the product does not require a second license. An accessor ID is a generic string that can be used by the vendor to further identify the caller of the license request.

For example, if a user invokes a compiler repeatedly, a multiuse rule might specify that the second and subsequent invocations do not require additional licenses. Multiuse rules may be based on any combination of the following tests that the server performs when a concurrent license is requested:

- The request for a license is associated with the same user as a previous request.
- The request for a license is associated with the same group as a previous request. The vendor can also change the meaning of the “same group” rule to implement a vendor rule. For example, the vendor might implement a multiuse rule that applies when a request is associated with the same display as a previous request. Vendors can also modify the meaning of “same group” in other ways, to implement whatever multiuse rules they design. Any vendor-specific rule overrides the “same group” rule.
- The request for a license is associated with the same node as a previous request (applicable to concurrent licenses only).
- The request for a license is associated with the same job ID as a previous request.
- The request for a license is associated with the same accessor ID as a previous request.

Product Wait Queues

Some products with concurrent licenses may use wait queues.

When a user invokes such a product, and there are no licenses available, the product can be enabled to ask if the user wants to wait for a license. If the user responds affirmatively, the user is added to the wait queue on each LUM Runtime network license server that provides concurrent licenses for the product. User names are added to the wait queue in chronological sequence. When a license becomes available, it is granted to the first user in the queue.

License Annotation

License annotation is data that is defined and included as part of the license information when a license is created. When the license is granted, the data is passed to the enabled application for its own use. Licenses of any type can be annotated.

A typical use of license annotation is to create licenses that correspond to different configurations of the same product. Consider an application that has several optional priced features, all delivered as part of the product package. The vendor can create license annotations to define which options the customer has bought and, therefore, which features are accessible to the end user.

Custom Configuration

Vendors who want to offer selected combinations of products, tailored more precisely to the needs of users, can define custom configurations by adding functions and products to a base configuration.

You specify the required content of a custom configuration when you order the configuration. You can order a custom configuration for one *seat* or for a block of any number of identical seats. If you order a configuration for a block of seats, the quantity of each add-on function or product must equal the number of seats in the block.

Each custom configuration, whether for a single seat or for a block of two or more seats, is assigned a separate custom configuration license. A custom configuration license is a special case of a concurrent network license or a concurrent or simple nodelocked license that contains a unique serial number identifying that custom configuration. The single serial number and license for a block configuration helps you to manage your installed licenses more easily.

After initial installation of a custom configuration, you can better manage the evolution and growth of your configurations, by ordering additional “add-on” functions and products, as necessary. To retain a single serial number and license, however, any changes made to the custom configuration must be applied to all seats under that serial number.

Bundle

Sometimes it may be useful to treat a set of products as a unique entity. By defining *bundles*, licensing system administrators can manage the licenses referring to products as a unit.

The licensing system administrator can view all the bundles in a licensing system and see the licenses that are in bundles. You cannot delete individual licenses in a bundle, but you can delete the bundle.

Also, it is possible to verify the consistency of a bundle if all the licenses in the bundle have been enrolled.

Products in the bundle are linked by a common serial number, but each product is still represented by its own license password. The custom configuration policy, however, provides a way of bundling in which all the products are represented by a single license.

Product-Specific Program

Vendors who need to perform some product-specific actions at the moment of the license enrollment can define in the license password the name and path of a program that LUM Runtime will run as part of the enrollment process. The product is responsible for the existence of such a program on the workstation where the enrollment takes place. If LUM does not find the program the enrollment fails. This policy is only available for simple nodelock licenses under a non-runtime-based enabling (see “License-Enabling Models” on page 16).

Customer Controlled Policies

The customer can exercise the license policies: hard stop / soft stop, user access restriction, and per-server / per-seat switch.

Hard Stop/Soft Stop

The vendor can enable a product so that you can choose the behavior of the product when the end user starts it and no licenses are available.

If no license is available, one of two things can happen:

- The product does not start, and there is no way for the end user to go on (hard stop policy).
- The product starts (soft stop policy).

When you enroll a product enabled for hard stop / soft stop, the default is hard stop. To use soft-stop, you must enable a network license server with the correct option. You can use the basic license tool to change the policy to soft stop and back again. For details, see “Scenario 9: Using the Hard Stop / Soft Stop Policy” on page 109.

When the soft stop policy is set, LUM Runtime keeps track of the high-water mark. The high-water mark is the maximum number of licenses ever granted for a given product beyond the number of licenses that are enrolled for that product. You can see this number through the basic license tool, and you can reset it to 0. Use this number to help you decide the number of additional license keys you need. When the hard stop policy is selected, the number of in-use licenses cannot exceed the number of enrolled licenses, so the high-water mark is not maintained.

While for CMU policies the user can exploit the LUM Runtime hard stop/soft stop policy, VMU application vendors can decide to have a soft stop policy that is under the full control of the enabled application itself.

User Access Restriction

You can use the user file to control which users have access to licenses for specific products. The user file is a flat ASCII file that you create using a text editor. For each product in the file, there is a list of users. It lists either those who are allowed to use the product (in which case no one else can use it) or those who are not allowed to use it (in which case anyone else can use it).

For details, see “Scenario 10: Restricting User Access” on page 112.

Switching from Per-Server to Per-Seat Licenses

Vendors of client/server applications who choose per-server/per-seat licensing provide you with two enrollment certificates:

- The per-server certificate, containing a per-server password
- The per-seat certificate, containing a per-seat password

You have the option to start in per-server mode, and switch at any time to per-seat mode, or start directly in per-seat mode. Once the per-seat mode has been activated, it is not possible to go back to per-server mode.

For information to help you decide between per-server and per-seat, see “Per-Server Licenses” on page 8 and “Per-Seat Licenses” on page 11. For information about how to perform the switch, see “Scenario 8: Switching from Per-Server to Per-Seat Licenses” on page 108.

License-Enabling Models

The product vendor can create license-enabled products that use nodelocked or network licenses. The enablement of nodelocked licenses can be either non-runtime-based or runtime-based.

If the vendor chose non-runtime-based enabling (nodelocked licenses only), the product does not make use of LUM Runtime on the workstation where the product runs. Following the vendor’s installation instructions, you may be required to store the password for such a product in a vendor-selected nodelock file. When you start the application, it checks the nodelock file to ensure you have a valid license. It is not necessary for the nodelocked license server to be running for the license to be granted. Information about use of the product is not logged. You cannot use the basic license tool to view information or get reports about the product and its usage.

If the vendor chose runtime-based enabling for a product with nodelocked licenses, the product makes use of LUM Runtime on the workstation where the product runs. It does not require configuration unless the end user has special requirements. The password for such a product is stored in the nodelocked license database. When you start the application, it contacts the nodelocked license server, which checks its database to ensure you have a valid license. Information about use of the product is logged. You can use the basic license tool to view information or get reports about the product and its usage.

A network license-enabled product makes use of LUM Runtime on the workstation where the product runs and requires some limited configuration on that workstation. The licenses are stored on one or more network license servers. When the user at a client starts a licensed program, LUM Runtime at the license server determines whether or not a license is available.

Table 2 on page 18 summarizes the license-enabling models, license types, and license policies.

Table 2. License Types and Policies Available for LUM Products

License-Enabling Model	License Types Available	License Policies Available
Nodelocked* License-Enabled Products (Non-Runtime-Based Enabling)	Simple Nodelocked	Try-and-Buy License Annotation Custom Configuration Capacity Product-Specific Program
Nodelocked License-Enabled Products (Runtime-Based Enabling)	Simple Nodelocked	Try-and-Buy License Annotation Capacity Bundles Custom Configuration
	Use-Once Nodelocked	User Access Restriction License Annotation
	Concurrent Nodelocked, Per server	Hard Stop/Soft Stop† Multiuse Rules User Access Restriction License Annotation Capacity† Bundles
Network License-Enabled Products	Concurrent	Hard Stop/Soft Stop† Multiuse Rules User Access Restriction Product Wait Queues License Annotation Capacity† Bundles Custom Configuration
	Reservable	Hard Stop/Soft Stop (when unreserved) User Access Restriction License Annotation
	Use-Once	User Access Restriction License Annotation
	Per-Seat	Hard Stop/Soft Stop User Access Restriction Per-Server/Per-Seat switch License Annotation

* This applies only to C-language applications.

† Soft-stop does not work with capacity for customer-managed use licenses.

Scalable Installation

LUM Runtime consists of separate installable features, so that you can install exactly what you need on each workstation.

For example, in AIX 5.0 for IA-64, all required LUM Runtime features are automatically installed on every AIX workstation as part of AIX 5.0 for IA-64 installation. You can install optional features, such as the graphical user interface, either when you install AIX or later.

On Windows, you can select the appropriate features depending on the role the workstation is to play in your licensing environment. There are features for runtime, communications and online documentation. Vendors can, optionally, incorporate the communications feature into the installation images of their license-enabled products at the minimum level of installation and configuration the products require. Alternatively, they can specify that you download and install LUM Runtime.

When you configure LUM Runtime, the configuration tool recognizes which features are installed and presents only the options consistent with the installed feature.

The tool for configuring LUM Runtime license servers and clients (i4cfg) has a command-line interface and a graphical user interface on all platforms, and an alternative, interactive script interface on all UNIX platforms.

Creating Licenses

LUM Runtime includes a tool that creates product licenses for the use of vendors who create license-enabled products. Two uses of the license creation tool are:

- The tool enables vendors to create these kinds of passwords:
  - Test passwords, for use in testing while enabling a product
  - Production passwords, to deliver to customers

  Distribution of production passwords to customers depends on the use control level of the license-enabled products:

  For customer-managed use control products, the customer receives the license password together with the product package.

  For vendor-managed use control products, for IBM license-enabled products, the customer requests the license password from the IBM country software password distribution center. For non-IBM license-enabled products, the customer requests the license password from the vendor software password distribution center.

To create production passwords, vendors must acquire the license for this tool from IBM or from Isogon Corp. The address of Isogon Corp. is:

Isogon Corporation
330 Seventh Avenue
New York, New York 10001
U.S.A.
Tel: (+1) 212-376-3200
Fax: (+1) 212-376-3280

- The tool is also useful for vendor sales representatives, who can be provided by the vendor with a production compound password for a vendor-managed use product. The compound password contains many licenses, from which the sales representative extracts licenses for individual customers.

For details about how to use this tool, see "i4lct - License Creation Tool" on page 171.

Administering Licenses

The *basic license tool* is used to administer licenses.

| The basic license tool has a command-line interface and a graphical user
| interface on all platforms.

It enables you to:

- Manage all types of licenses.

  The administrator can use the basic license tool to manage nodelocked and network licenses.
- Add, update, or delete licenses.

  Add licenses to or delete licenses from the network license server or nodelocked licensed server database; update information about existing customer-managed use licenses.
- Display information about licenses.

  Display a notebook of information about the licenses installed for each product.
- Distribute licenses.

  Extract licenses from a compound password and distribute them among the network license servers available on the network.
- Reserve licenses.

  Manage the reservation of reservable licenses for the exclusive use of certain users.
- Manage multiple network and nodelocked license servers.

From any properly configured workstation, you can view and manage licenses that are installed on any network license server and on any nodelocked license server in the network. Working at a single administration site, you can manage all kinds of licenses on all workstations. The capability to manage licenses on nodelocked license servers is particularly useful for per-server and concurrent nodelocked licenses.

- Generate reports.

  Standard Event Report. Displays detailed information about significant events that occur on the license servers that you specify.

  License Request by Product Report. Displays statistical information about the use of the licenses of a product in the time interval that you specify. For each product, it reports the licenses that were requested, the licenses that were granted, and the percentage of rejections.

  License Request by User Report. Displays statistical information about the use of products by users in the time interval that you specify. For each user, it reports the licenses that were requested, the licenses that were granted, and the percentage of rejections.

  License Use by Product Report. Displays statistical information about the use of the licenses of a product in a specified time interval. For each product, it reports:

  – The maximum number of nodes that used licenses for the product at the same time

  – The maximum number of users that used licenses for the product at the same time

  – The average time the licenses were in use

  License Use by User Report. Displays statistical information about the use of the licenses by each user in a specified time interval. For each user, it reports the times the user requested licenses and the length of time the user kept the licenses in use.

  Customer-Managed Use Audit. Reports the following information for customer-managed use product transactions:

  – Vendor name

  – Product name

  – Product version

  – Administrator information

  – Time stamp

  – Number of licenses

  – Transaction type (for example, product enrolled, license distributed, license deleted, license updated, per-server/per-seat license migrated)

  - Signature stamp (user, group, and node)
  - Signature information
- Use the high-water mark.

  When the soft-stop policy is in effect, the high-water mark is recorded in the licensing database. The high-water mark is the maximum number of licenses ever granted for a given product beyond the number of licenses that are enrolled for that product. You can see this number through the basic license tool and can reset it to 0. This number assists you in deciding how many additional licenses (when customer-managed) or license keys (when vendor-managed) you need. When the hard stop policy is selected, the number of in-use licenses cannot exceed the number of enrolled licenses, so the high-water mark is not maintained.
- Set the threshold.

  You can set a threshold percentage of licenses. If more than the threshold percentage of licenses for a product are in use, messages about the level of usage are logged. There is a single threshold that applies to all vendor-managed products. You can change that threshold, and you can also set a separate threshold for each product.
- Exercise customer-controlled policies.

  You can switch between hard stop and soft stop, switch from per-server to per-seat, and manage the identifiers of application clients using per-seat licenses.
- Administer high-availability licensing (see “High-Availability Licensing”).

  Create clusters, activate and deactivate members of clusters, and show cluster status.
- Administer bundles.

  Display a list of bundles, display a list of bundle features, and delete bundles.

For help on using the basic license tool, see “Performing Basic Administration” on page 97.

High-Availability Licensing

High-availability licensing enables you to set up an environment in which there is a very high degree of certainty that concurrent licenses will be available, even if a network license server goes down.

When you use this option, you create a cluster of network license servers. A cluster is a group of from 3 to 12 network license servers that jointly serve vendor-managed concurrent licenses that are enrolled on the cluster rather than on an individual server.

You can create and administer a cluster, and administer high-availability licenses, from any workstation. However, no Windows 95 or Windows 98 network license server can be a member of a cluster.

While some servers in the cluster are serving licenses, one or more servers remain in reserve, ready to take over should an active server fail.

Each active server serves an equal share of the licenses enrolled on the cluster. When a server becomes unavailable and another server takes its place, responsibility for the licenses is automatically redistributed among active servers.

For high-availability licensing to work for a particular product, the product vendor must supply a password tied to a cluster rather than to an individual target server.

High-availability licensing works only with the IP protocol and does not support the product wait queue policy. Before you decide to use high-availability licensing for a product, make sure such a password is available from the product's vendor.

High-availability licensing is recommended only for users who are already experienced with managing individual license servers and who already have a stable licensing environment working.

For planning information, see "Planning Clusters for High-Availability Licensing" on page 42. For an example, see "Scenario 11: Creating and Managing a Cluster" on page 113.

Backing Up Databases and Files

On license servers, there is an automatic backup procedure for LUM Runtime databases and files. See "Using the Built-In Backup and Recovery Procedure" on page 185.

Working with Licensed Products

This section explains what happens when a user starts a licensed product, depending on how the product is enabled.

Only C, C++, and Java programs can be license-enabled.

In the figures and text in this section, references to the *enabled application* or *enabled applet* refer to the application or applet itself (which contains API calls) plus the embedded code that services the API calls (which carries out the described steps).

This section assumes that, if required:

- LUM Runtime is installed properly.
- LUM Runtime is configured properly.
- A Web server computer is set up properly for Java applications and applets that use Java Client Support (JCS).
- The network is running properly.

If not, what happens depends on how the vendor enabled the product. See the product documentation for details.

Central Registry License Server

Some of the scenarios in this section show the use of a *central registry license server*. The central registry is a repository of information that all the other network license servers can use. If you plan to install customer-managed use products with network licenses, or products with reservable licenses, you must identify one (and only one) central registry. Otherwise, the central registry license server is not required.

The central registry can contain:

- A record of all per-seat licenses
- The list of application clients to which per-seat licenses have been granted
- A record of soft stop license use
- The high-water mark
- Reserved licenses that have not yet been granted

Working with Nodelocked Licenses (Non-Runtime-Based Enabling)

Figure 1 shows what happens when an end user invokes a C or C++ application with nodelocked licenses for which the vendor chose non-runtime-based enabling. The licenses must be simple nodelocked licenses.

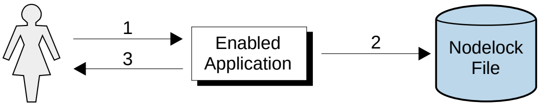


Figure 1. Using a Nodelocked License (Non-Runtime-Based Enabling)

1. The end user invokes the application.
2. The application checks the nodelock file to ensure a license is stored on the local system.
3. If there is a valid license in the nodelock file, the application runs. If not, depending on how the vendor enabled the application, it may return information to the end user, or it may run even with no license available.

For information about how to edit a nodelock file, see “Appendix B. Using the Nodelock File” on page 219.

Working with Nodelocked Licenses (Runtime-Based Enabling)

Figure 2 shows what happens when an end user invokes an application with nodelocked licenses for which the vendor chose runtime-based enabling. The licenses can be simple nodelocked, use-once nodelocked, or concurrent nodelocked licenses.

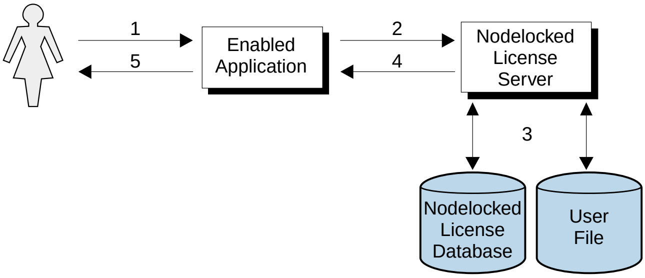


Figure 2. Using a Nodelocked License (Runtime-Based Enabling)

1. The end user invokes the application.
2. The application requests a license from the nodelocked license server on the local system.
3. The nodelocked license server checks that there is a valid license on the workstation and that this user is authorized to use it.

   If there is no nodelocked license but the application uses concurrent nodelocked licenses and implements the soft stop policy, the nodelocked license server checks for a soft stop license and checks the user file for authorization.
4. The nodelocked license server returns the status of the license request to the application.
5. If a license was found and granted, or if a soft stop license was granted, the application runs. If no license can be granted, depending on how the vendor enabled the product, the application may return information to the end user, or it may run even with no license available.

Working with Use-Once Licenses

Figure 3 shows what happens when an end user invokes an application with use-once licenses.

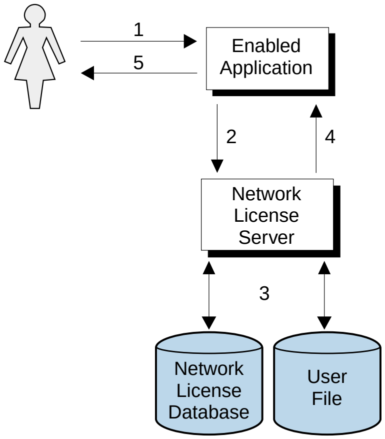


Figure 3. Using a Use-Once License

1. The user invokes the application.
2. The application requests a license from the network license server.
3. The network license server checks its license database for an available license and the user file for authorization.
4. The network license server returns the status of the request to the application. If a license was found and granted, the application runs, and one license is subtracted from the number of available use-once licenses.
5. If no license can be granted, depending on how the vendor enabled the product, the application may return information to the end user, or it may run even with no license available.

Figure 4 shows what happens when an end user invokes a Java application or applet with use-once licenses, using Java Client Support. In this case, the Web server computer, rather than the end user's workstation, serves as the network license client.

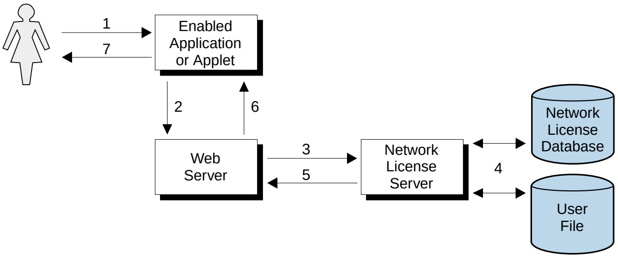


Figure 4. Using a Use-Once License with Java Client Support

1. The user invokes the application or downloads the applet through a Web browser.
2. The application or applet sends a license request to the Web server using the http protocol.
3. The Web server requests a license for the application or applet from the network license server.
4. The network license server checks its license database for an available license and the user file for authorization.
5. The network license server returns the status of the request to the Web server. If a license was found and granted, one license is subtracted from the number of available use-once licenses.
6. The Web server returns the status of the request to the application or applet, using the http protocol. If the status is OK, the application or applet runs.
7. If no license can be granted, depending on how the vendor enabled the application or applet, it may return information to the end user, or it may run even with no license available.

Working with Concurrent Licenses

Figure 5 shows what happens when an end user invokes an application with concurrent licenses.

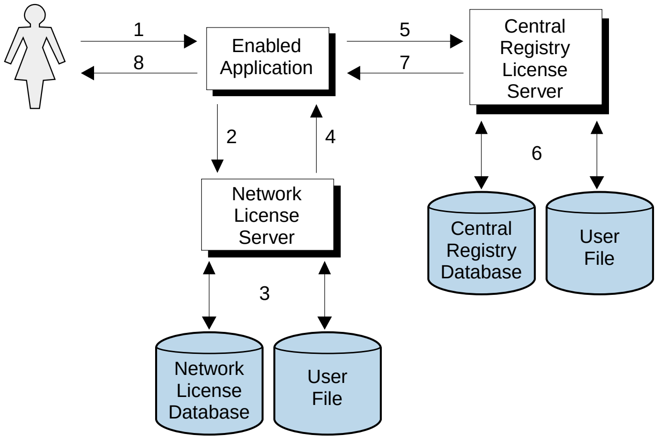


Figure 5. Using a Concurrent License

1. The user invokes the application.
2. The application requests a license from the network license server.
3. The network license server checks its license database for an available license and the user file for authorization.
4. The network license server returns the status of the request to the application. If a license was found and granted, the application runs.
5. If a network license was not found, and the application implements the soft stop policy, the application requests a soft stop license from the central registry license server.
6. The central registry license server checks its database for a soft stop license and the user file for authorization.
7. The central registry license server returns the status of the request to the application. If a soft stop license was granted, the application runs.
8. If no license can be granted, depending on the vendor enabled the product, the application may return information to the end user, or it may run even with no license available.

Figure 6 shows what happens when an end user invokes a Java application or applet that has concurrent licenses using Java Client Support. In this case, the Web server computer, rather than the end user’s workstation, serves as the network license client.

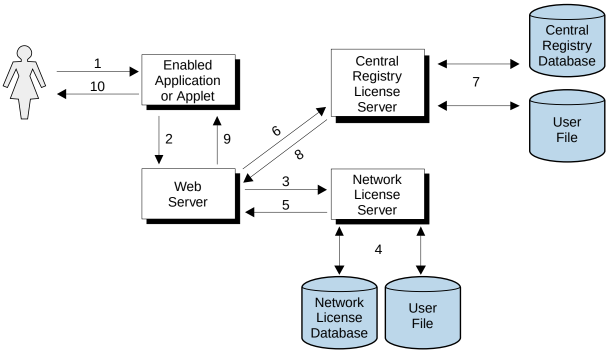

1
10
Enabled Application or Applet
2
9
Web Server
3
5
Network License Server
6
8
Central Registry License Server
7
Central Registry Database
User File
4
Network License Database
User File

Figure 6. Using a Concurrent License with Java Client Support

1. The user invokes the application or downloads the applet through a Web browser.
2. The application or applet sends a license request to the Web server using the http protocol.
3. The Web server requests a license for the application or applet from the network license server.
4. The network license server checks its license database for an available license and the user file for authorization.
5. The network license server returns the status of the request to the Web server. If a license was found and granted, the Web server returns a positive status to the application or applet, and it runs.
6. If no concurrent license was found, the Web server requests a soft-stop license from the central registry license server.
7. The central registry license server checks its database for a soft-stop license and the user file for authorization.
8. The central registry license server returns the status of the request to the Web server.
9. The Web server returns the status of the request to the application or applet, using the http protocol. If the status is OK, the application or applet runs.
10. If no license can be granted, depending on how the vendor enabled the application or applet, it may return information to the end user, or it may run even with no license available.

Working with Reservable Licenses

Figure 7 shows what happens when an end user invokes an application with reservable licenses.

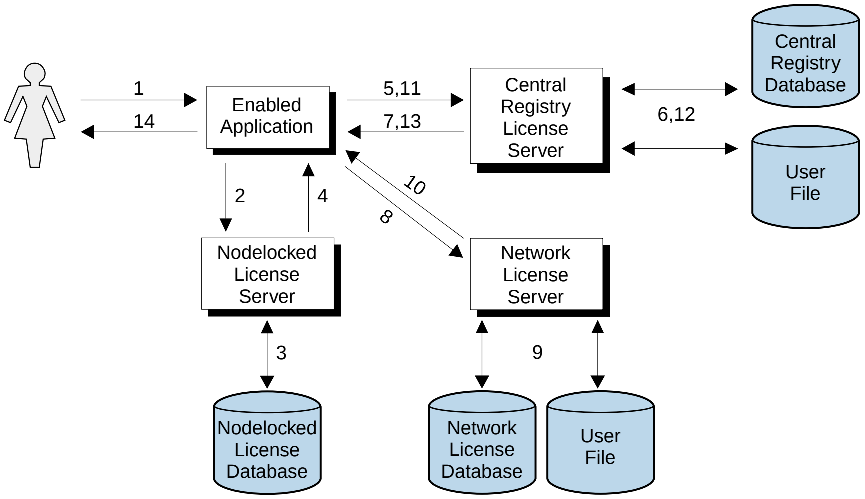

1
14
Enabled Application
5,11
7,13
Central Registry License Server
6,12
Central Registry Database
User File
2
4
10
8
Nodelocked License Server
Network License Server
3
9
Nodelocked License Database
Network License Database
User File

Figure 7. Using a Reservable License

1. The user invokes the application.
2. The application requests a reserved license from the nodelocked license server.
3. The nodelocked license server checks its database for a reserved license. This is a license that you reserved for the user. It was granted to the user and stored on the local workstation, in response to a previous request.
4. The nodelocked license server returns the status of the request to the application. If a license was found, the application runs.
5. If the nodelocked license server does not find a license, the application requests a reserved license from the central registry license server. This is a license that you have reserved for this user, group, or workstation.
6. The central registry license server checks its database for a reserved license and the user file for authorization.
7. The central registry license server returns the status of the request to the application. If a reserved license was found and granted, it is stored in the nodelocked license server’s database, and the application runs.
8. If a reserved license was not found, the application requests a reservable license from the network license server. This is a reservable license that you have not reserved for anyone.
9. The network license server checks its license database for a reservable license and the user file for authorization.

10. The network license server returns the status of the request to the application. If a reservable license was found and granted, the application runs.
11. If a reservable license was not found, and the application implements the soft stop policy, it requests a soft stop reservable license from the central registry license server.
12. The central registry license server checks its database for a soft stop reservable license and the user file for authorization.
13. The central registry license server returns the status of the request to the application. If a soft stop license was granted, the application runs.
14. If no license can be granted, depending on how the vendor enabled the product, the application may return information to the end user, or it may run even with no license available.

Figure 8 shows what happens when an end user invokes a Java application or applet with reservable licenses using Java Client Support. In this case:

- The Web server computer, rather than the end user's workstation, serves as the network license client.
- Reserved licenses, when granted, are not moved to the nodelocked license server.

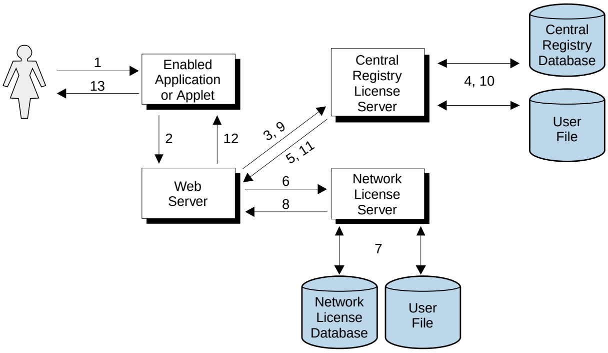


Figure 8. Using a Reservable License with Java Client Support

1. The user invokes the application or downloads the applet through a Web browser.
2. The application or applet requests a reserved license from the Web server using the http protocol.
3. The Web server requests a reserved license for the application or applet from the central registry license server. This is a license that you have reserved for this user, group, or workstation.

4. The central registry license server checks its database for a reserved license and the user file for authorization.
5. The central registry license server returns the status of the request to the Web server. If a reserved license was found and granted, the Web server returns a positive status to the application or applet, and it runs.
6. If a reserved license was not found, the Web server requests a reservable license from the network license server. This is a reservable license that has not been reserved.
7. The network license server checks its database for a reservable license and the user file for authorization.
8. The network license server returns the status of the request to the Web server.
9. If no reservable license was found, the Web server requests a soft-stop license from the central registry license server.
10. The central registry license server checks its database for a soft-stop license and the user file for authorization.
11. The central registry license server returns the status of the request to the Web server.
12. The Web server returns the status of the request to the application or applet, using the http protocol. If the status is OK, the application or applet runs.
13. If no license can be granted, depending on how the vendor enabled the application or applet, it may return information to the end user, or it may run even with no license available.

Working with Per-Server Licenses

Figure 9 shows what happens when an end user invokes an application with per-server licenses when per-seat has not been enabled. In the figure, the application server is license-enabled.

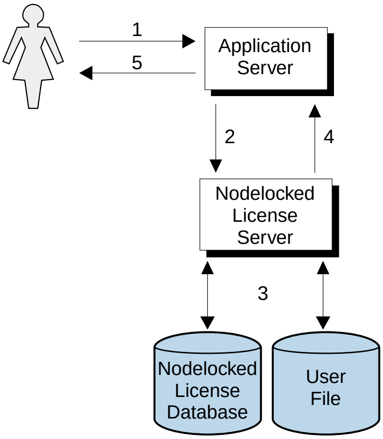


Figure 9. Using a Per-Server License

1. The application client user invokes the application.
2. The application server requests a per-server license from the nodelocked license server. This is a license that you have stored on the nodelocked license server.
3. The nodelocked license server checks the nodelocked license database for such a license and the user file for authorization. If no license is found, but the application implements the soft stop policy, the nodelocked license server checks for a soft stop license.
4. The nodelocked license server returns the status of the request to the application server. If a license was found, or if a soft stop license was granted, the application runs.
5. If no license can be granted, depending on how the vendor enabled the product, the application may return information to the end user, or it may run even with no license available.

Working with Per-Seat Licenses

Figure 10 shows what happens when an end user invokes an application with per-server / per-seat licenses when per-seat has been enabled. In the figure, the application server is license-enabled.

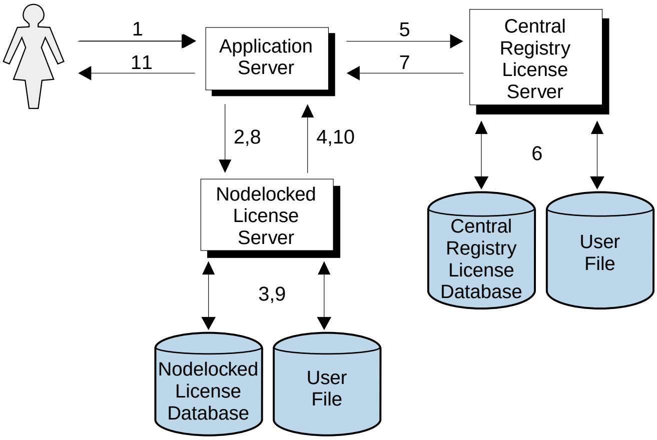

1
11
Application Server
5
7
Central Registry License Server
2,8
4,10
6
Nodelocked License Server
Central Registry License Database
User File
3,9
Nodelocked License Database
User File

Figure 10. Using a Per-Seat License

1. The application client user invokes the application.
2. The application server requests a per-seat license from the nodelocked. license server. This is a license that has already been granted to the user on a previous request and stored on the local workstation.
3. The nodelocked license server checks the nodelocked license database for such a license.
4. The nodelocked license server returns the status of the request to the application server. If a per-seat license was found, the application runs.
5. If no per-seat license was found on the nodelocked license server, the application server requests a per-seat license from the central registry license server.
6. The central registry license server checks whether a license is already being used by the requesting application client, possibly granted through another application server. In such a case the application can start without having a new license granted. Otherwise, the central registry license server checks whether a per-seat license is available. If so, it grants the license and records the application client identifier. If no per-seat license is found, but the application implements the soft stop policy, the central registry license server checks for a soft stop license.
7. The central registry license server returns the status of the request to the application server.

8. If a per-seat or soft stop license was granted, the application sends a shadow copy of the granted per-seat license to the nodelocked license server.
9. The nodelocked license server adds the shadow copy to the nodelocked license database.
10. The nodelocked license server returns the status of the request to the application server, and the application runs.
11. If no license can be granted, depending on how the vendor enabled the product, the application may return information to the end user, or it may run even with no license available.

Figure 11 shows what happens when an end user invokes a Java application or applet with per-server/per-seat licenses using Java Client Support, when per-seat has been enabled. In this case:

- The Web server computer, rather than the end user’s workstation, serves as the network license client.
- When a per-seat license is granted no shadow copy is stored on the nodelocked license server.

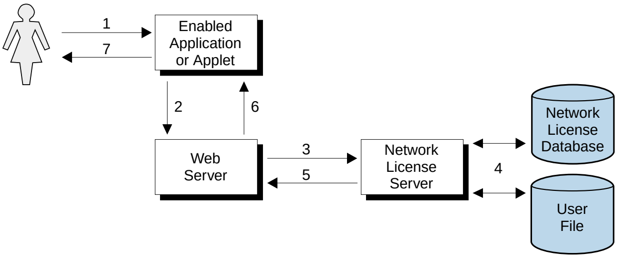


Figure 11. Using a Per-Seat License with Java Client Support

1. The user invokes the application or downloads the applet through a Web browser.
2. The application or applet sends a license request to the Web server. It requests a per-seat license from the Web server, using the http protocol.
3. The Web server requests a license for the application or applet from the central registry license server.
4. The central registry license server checks whether a license is already being used by the requesting application client, possibly granted through another application server. In such a case the application can start without having a new license granted. Otherwise, the central registry license server checks whether a per-seat license is available. If so, it grants the license and records the application client identifier. If no per-seat license is found,

but the application implements the soft-stop policy, the central registry license server checks for a soft-stop license. If a soft-stop license is available, it is granted.

5. The central registry license server returns the status of the request to the Web server.
6. The Web server returns the status of the request to the application or applet, using the http protocol. If the status is OK, the application or applet runs.
7. If no license can be granted, depending on how the vendor enabled the application or applet, it may return information to the end user, or it may run even with no license available.

Chapter 2. Planning Network Licensing

This chapter is for the administrator who is setting up an environment in which multiple client workstations can share licenses stored on one or more network license servers. Skip this chapter if:

- You are using only applications with nodelocked licenses, *and*
- You do not intend to use the basic license tool from one workstation to administer licenses on other workstations

Designing the network licensing environment that provides the best solution for your business requires careful and thoughtful planning. This chapter will assist you, as the system administrator, to plan the environment for network licensed products and to decide on the configuration options you need.

It is important that you allow enough time for planning, especially when using LUM Runtime in large networks or across subnetworks.

The decisions you need to make include:

- How many network license servers you will set up
- Which workstations, if any, will be the network license servers
- How you will distribute product licenses among the network license servers
- Which clients will have access to which servers
- How clients will locate the servers
- Whether all servers will serve all clients, or whether you will set up independent groups of servers and clients
- Which workstation will be the central registry (if required)
- Whether you will make use of LUM Runtime high-availability licensing; and, if so, which network license servers will be part of a cluster
- Whether and how you will make use of remote administration to administer from one workstation the network and nodelocked licenses on other workstations
- Which workstations will be used as Web servers for Java applications and applets

Selecting Your Servers

The license server system depends on a stable network. If name resolution and routing in a network are not running properly, then the network license servers, network license clients, and central registry license server may be unable to communicate properly.

In designating workstations to be network license servers or the central registry license server, keep the following criteria in mind:

- A license server should be a computer that stays on at all times. Workstations that are frequently unavailable or unreliable, such as those that are brought down often for testing or maintenance, are not good candidates.
- It is important to keep license servers for production environments separate from those for test environments.
- If you have multiple subnetworks, then ideally, the servers should be on the same subnetwork as the majority of clients that will run the licensed products. Accessing in another subnetwork, across a bridge or router, may not be quite as fast.

If the network spans subnetworks, you need to spread the licenses out among network license servers. Also, inside the same network, each client request for a license generates network traffic. Therefore, it can be useful to spread the application licenses across more than one network license server, and across multiple platforms. When a computer is down, the licenses assigned to the network license server on that system are unavailable, but licenses assigned to other network license servers remain available. Having several license servers on the network will help to prevent bottlenecks that result when many clients communicate with a single network license server.

The number of network license servers in the network should be proportional to the frequency of license requests rather than to the number of users. For example, suppose that a compiler and a word processor are both license-enabled. A single user running many short compilations will place a heavier load on the license server than many users each starting the word processor once.

Computers that function as network license servers or as the central registry can also run the license-enabled products. The license server software does not have a noticeable effect on the performance of products.

When you have identified the network license servers, and before you configure them, you must organize the servers into one or more groups. The servers in a group form an independent licensing environment and serve a common set of clients. You must also identify a central registry license server for the group, if it requires one. “Planning Direct Binding” on page 39 will help you to group your servers.

Network Computing System (NCS)

The network computing system (NCS) is a set of tools for distributed computing. Some of its tools are included in the LUM Runtime features.

The *Remote Procedure Call (RPC) Runtime Library* provides the calls that enable local programs to execute procedures on remote hosts. These calls transfer requests and responses between clients (the programs that call the procedures), and servers (the programs that execute the procedures). The RPC that is embedded in all license servers and in enabled products provides a common mechanism to support the request and acquisition of licenses.

Planning Direct Binding

Network license clients locate the network license servers by means of the *direct binding* mechanism.

With direct binding, you make a list of your network license servers and the central registry. The list is called the *direct binding servers list* in this chapter and in the configuration scenarios in “Setting Up Your Servers and Clients” on page 69.

During configuration of servers and clients, you specify the network addresses of all the servers on the list. The network addresses are stored on every server and every client in a configuration file.

All network license servers, and the central registry license server, listen for incoming communications on well-known ports:

Table 3. Communication Ports Used

Protocol	Ports	Operating Environments and Systems
NetBIOS	115 and 109	Windows NT, Windows 2000, and Windows Terminal Server
IPX	1515 and 10999	Windows 95, Windows NT, Windows 2000, and Windows Terminal Server
TCP/IP	1515 and 10999	Windows 95, Windows 98, Windows NT, Windows 2000, and Windows Terminal Server

The network license client uses these port numbers, together with the network addresses of the server systems that are specified in the configuration file, to locate and connect to the servers.

In addition to enabling clients to locate license servers, the direct binding mechanism makes it possible, from any license server, to use the basic license tool remotely to administer licenses on all the servers in the direct binding list. By adding nodelocked license servers to the direct binding servers list, you can administer licenses that are on remote nodelocked license servers.

Nodelocked license servers that are configured for remote administration listen on port 12999 if you use TCP/IP or IPX; 129 if you use NetBIOS. The basic license tool uses this port number, together with the network addresses of the nodelocked license servers, to locate and connect to the servers for remote administration.

In the same way, you can enable remote administration from a workstation configured only as a nodelocked license server. When you configure a nodelocked license server, you can create a direct binding servers list that contains all the license servers (both network and nodelocked) whose licenses you want to administer remotely.

Figure 12 shows an example of a network configuration that uses direct binding. The example shows a network license server and two nodelocked license servers in a network. From any of the license servers, it is possible to administer licenses on all the license servers.

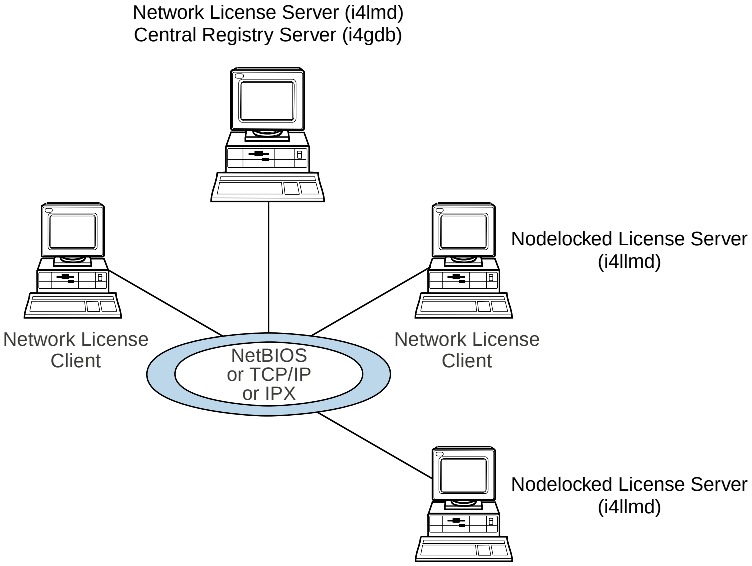

Network License Server (i4lmd)
Central Registry Server (i4gdb)
Nodelocked License Server (i4llmd)
Network License Client
NetBIOS or TCP/IP or IPX
Network License Client
Nodelocked License Server (i4llmd)

Figure 12. Direct Binding with Network License Servers and Nodelocked License Servers

Figure 13 on page 41 shows an example of a Java configuration. The example shows two network license servers, a network license client, and a network application client in a Java-enabled network. Server A communicates with the Java-enabled applet or application using the Hypertext Transfer Protocol (HTTP), if necessary being protected by a firewall.

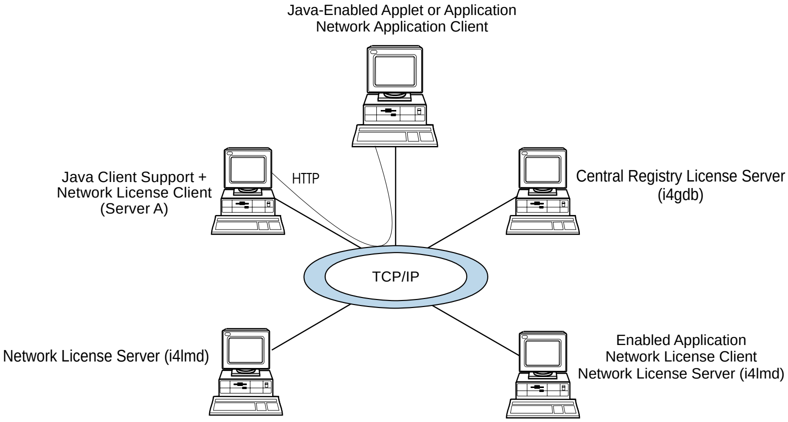

Java-Enabled Applet or Application
Network Application Client
Java Client Support +
Network License Client
(Server A)
HTTP
Central Registry License Server
(i4gdb)
TCP/IP
Network License Server (i4lmd)
Enabled Application
Network License Client
Network License Server (i4lmd)

Figure 13. Direct Binding with Java Client Support

Before you begin configuring workstations to use direct binding, be sure you have identified all the servers in your direct binding servers list. Then go to the configuration scenarios in “Setting Up Your Servers and Clients” on page 69. As you configure direct binding at each server and at each client, be sure you enter exactly the same list of servers.

Performance Notes:

- It is important that the direct binding servers list include all the servers, and that it exclude workstations that will not function as servers. If there are any extra workstations in the list, there will be a noticeable effect on performance.
- When a network license client configured for direct binding requests a license, and multiple servers have licenses for the product, the servers are checked for an available license in the sequence they were entered into the direct binding servers list during configuration of the client. Therefore, if you know how frequently specific network license clients will request licenses for specific products, you may be able to balance the workload of the servers by varying the sequence in which servers are defined at different clients.
- If you are certain that all the licenses requested by a particular network license client will be supplied by a subset of the servers, when you configure the client you may configure direct binding with just those servers, rather than all servers in the direct binding servers list, to improve performance. If you configure the client in this way, make sure that it is configured to communicate with the necessary servers; otherwise, it will not be able to get licenses.

Planning the Central Registry

The central registry license server process provides a mechanism for storing licensing information in a database common to all the servers. It is used for the administration of customer-managed use products and products with reservable licenses.

Only one central registry process must be running in a direct binding servers list. This ensures that the data is accurate and complete.

Notes:

1. You use the configuration tool to specify where to start the central registry.
2. Select the node where you will run the central registry carefully. After you place the central registry on a node, it cannot be moved.
3. The workstation running the central registry must be up and running in order to perform administration tasks on network customer-managed products or on products using reservable licenses.

Planning for Java Products that Use Java Client Support

Java Client Support (JCS) provides an alternative environment to the native Java APIs for running Java applications. Java Client Support is the only way to run Java applets.

If you plan to use license-enabled Java products using Java Client Support, you will need to set up one or more Web server computers. License-enabled Java applications and applets request licenses from the Web server. The Web server computer, in turn, serves as the network license client.

Planning Clusters for High-Availability Licensing

High-availability licensing is recommended for only those users who are already experienced with managing individual license servers and who already have a stable licensing environment working.

For high-availability licensing, you set up clusters of network license servers connected through TCP/IP. For concurrent licenses with vendor-managed use control only, the software vendor generates passwords that are bound to the cluster rather than to a single server. Some of the servers in a cluster serve licenses, while others wait in reserve to take over in case a serving server goes down. The servers that are serving at any time share equally the responsibility for the licenses that are bound to the cluster, and keep one another informed about the status of the licenses.

You can create and administer a cluster, and administer high-availability licenses, from any workstation. However, no Windows 95 or Windows 98 network license server can be a member of a cluster.

A network license server that is a member of a cluster can serve licenses that are bound to the server and participate as a member of a cluster at the same time.

Restrictions on Cluster Size and Composition

For security reasons, it is necessary to impose strict rules on the size and composition of clusters. Be very careful when you decide how many and which servers to put in a cluster. You will not be able to change your decisions after the fact and they will affect the size and composition of the cluster as long as it exists.

When you create a cluster (see “Scenario 11: Creating and Managing a Cluster” on page 113), you specify the initial number of servers in the cluster, and which servers they are. The initial number must be in the range 3 through 10. The first server assigned to the cluster is automatically activated; that is, it is available to participate in serving licenses as part of the cluster. You must explicitly activate the other members.

The initial number of servers dictates two important attributes of the cluster:

- The minimum number of servers that must be activated in the cluster for the cluster to work
- The maximum number of servers that can be added to the cluster in addition to the initial number

If you want to replace a server workstation that is one of the initial minimum number, to upgrade the hardware or to replace failing hardware, add a new server to the cluster. The number of new servers you can add, even to replace other servers, is limited.

Attention: The initial minimum number of servers must always be in the cluster; they must not be deactivated. If any is deactivated, the cluster ceases to serve licenses.

Passwords that are bound to a cluster are usable on only that cluster. If you find it necessary to delete a cluster and create a new one, or to create additional clusters, you will not be able to use existing passwords on the new cluster.

Note: To delete a cluster, deactivate all its members. When you deactivate a server, it must be up and running.

After a cluster has been created and its members have been activated, the number of activated members determines how many servers must be up and running for the cluster to function.

The relationships between these cluster attributes is shown in Table 4 .

The minimum number of servers up and running, as shown in the table, is the number of servers that serve licenses. All servers beyond that number are in reserve, waiting to take over if a serving server goes down.

Table 4. Number of Servers in a Cluster

Initial Number	Min. no. Activated for Cluster to Work	Max no. Added after Cluster Creation	Actual no. Activated	Min. no. Up and Running for Cluster to Work
3	2	1	2 3 4	2 2 3
4	4	2	4 5 6	3 3 4
5	4	1	4 5 6	3 3 4
6	6	2	6 7 8	4 4 5
7	6	1	6 7 8	4 4 5
8	8	2	8 9 10	5 5 6
9	8	1	8 9 10	5 5 6
10	10	2	10 11 12	6 6 7

Examples of Cluster Size Rules

Example 1: Initial number of servers is 3

The following rules apply:

- During the life of the cluster, you can add only one server to the cluster. This means you can add a new server with upgraded hardware, and deactivate one of the original three servers, only once during the life of the cluster. This scenario has the effect of replacing a server with an upgraded workstation.

Alternatively, you can add a fourth server to the cluster without deactivating any of the original three, thus increasing the cluster size to four servers. Again, you can add a server only once.

- No matter whether the cluster has three or four members, at least two members must be activated for the cluster to work.
- More than half of the activated servers must be up and running for the cluster to work.

Table 5 shows how the servers are deployed, depending on how many are activated, and how many are up and running.

Table 5. Example - Cluster with Three Initial Members

Number of Activated Members	**Number of Members Up and Running**	**Number of Members Serving Licenses**	**Number of Servers In Reserve**
2	2	2	0*
3	2	2	0*
	3	2	1
4	3	3	0*
	4	3	1

* When the number of servers in reserve is 0, there is no high-availability advantage

Example 2: Initial number of servers is 6

The following rules apply:

- During the life of the cluster, you can add two servers to the cluster. This means you can add two new servers with upgraded hardware, in effect replacing two servers with upgraded workstations.

  Alternatively, you can add one or two servers to the cluster without deactivating any of the original six, thus increasing the cluster size to seven or eight servers.
- Whether the cluster has six, seven, or eight members, at least six members must be activated for the cluster to work.
- More than half of the activated servers must be up and running for the cluster to work.

Table 6 on page 46 shows how the servers are deployed, depending on how many are activated and how many are up and running.

Table 6. Example - Cluster with Six Initial Members

Number of Activated Members	Number of Members Up and Running	Number of Members Serving Licenses	Number of Servers In Reserve
6	4	4	0*
	5	4	1
	6	4	2
7	4	4	0*
	5	4	1
	6	4	2
	7	4	3
8	5	5	0*
	6	5	1
	7	5	2
	8	5	3

* When the number of servers in reserve is 0, there is no high-availability advantage

Cluster Membership Considerations

If you use direct binding, be sure all network license clients that will use licenses bound to the cluster, and all servers that are members of the cluster, have all the servers of the cluster in their direct binding server list in order to exploit fully the high availability of licenses.

A server can be activated in only one cluster at any time. If you assign a server to a cluster and never activate it, or explicitly deactivate it, it can join a second cluster and be activated there. But in this case, the server cannot be activated in its original cluster, and no other server can be substituted in the original cluster. To reactivate the server in its original cluster, you must first deactivate it in the second cluster.

Note: Either disable or enable remote administration on all servers. If you enable or disable it on only some servers, you may have problems enrolling and removing licenses bound to the cluster.

Running the NetBIOS Socket Server

You need to run the NetBIOS socket server on any Windows NT or Windows 2000 workstation that is configured to communicate with an OS/2 workstation through NetBIOS. The OS/2 workstation must run the NetBIOS socket server as well. NetBIOS is not supported on Windows 95 or Windows 98.

To start the NetBIOS socket server:

1. If your workstation is configured as a server that provides licenses to OS/2 clients connected through NetBIOS, in `i4ls.ini` set:

   ```
   HasOS2Clients=yes
   ```

2. If your workstation needs to bind an OS/2 LUM Runtime server connected through NetBIOS, on the Direct binding page of the configuration tool:

   a. In the **Name** field, add the name of the OS/2 server in uppercase.

   b. Select the **OS/2 server** check box.

Verifying Network Connections

LUM Runtime provides the test verification tool (i4tv) to verify that license servers are running properly. For a detailed explanation of how to start the tool and how to use it, see “i4tv - Test Verification Tool” on page 161.

Chapter 3. Installing and Maintaining LUM Runtime

This chapter is intended for:

- The software vendor who needs to install LUM Runtime on Windows NT, Windows 2000, Windows Terminal Server, Windows 95, or Windows 98 for use in license-enabling a product
- The customer of a license-enabled product, if the vendor did not incorporate the LUM Runtime features into the product

This chapter explains:

- The LUM Runtime features you can select ("Selecting LUM Runtime Features" on page 50)
- LUM Runtime hardware and software requirements ("Hardware Requirements" on page 51 and "Software Requirements" on page 51)
- How to get the LUM Runtime code ("Obtaining the LUM Runtime Code" on page 53)
- How to install LUM Runtime interactively, using the installation GUI ("Installing LUM Runtime Interactively" on page 54)
- How to install LUM Runtime noninteractively ("Installing LUM Runtime Noninteractively" on page 63)
- How to install LUM Runtime from a code server ("Installation Using a Code Server" on page 65)
- How to uninstall LUM Runtime features ("Uninstalling LUM Runtime Features" on page 66)

Before You Install

Read this section to be sure you have everything you need and that you have made the necessary decisions before installing.

Note: To install LUM Runtime, you must log in as a user with administrator privileges. This ensures that the path and registry information is updated during the installation. If you try to install LUM Runtime from another user ID, the installation program will issue the message:

```
Installation stopped.  Only the user administrator can install LUM Runtime.
```

To determine whether LUM Runtime is already installed on your workstation, refer to the following files:

Communications feature

`c:\ifor\ls\conf\comver`

Runtime feature

`c:\ifor\ls\conf\runver`

where *c* is the drive letter.

Selecting LUM Runtime Features

LUM Runtime consists of these features:

- Runtime
- Communications
- Documentation

Before you install on any workstation, you must decide which features to install. Your selection of features depends on the role the workstation will play in your licensing environment, as shown in Table 7 on page 51. If your workstation will play more than one role in the licensing environment, be sure you install the features required for all the roles. See “Setting Up Your Servers and Clients” on page 69 for information about configuring your workstations to act in these roles.

Table 7. Required Features by Workstation Role

Role of the workstation	License Administration	Required Features
Nodelocked license server	Local administration only	Runtime
Nodelocked license server or Network license server or Network license client or Central registry license server	Local and remote administration	Runtime and Communications
Network license client	No administration	Communications
Network license client and Web server support for license-enabled Java applications and applets	No administration	Communications and LUM Java Client Support (Windows

Install the Documentation feature on any workstation to give you access to the online *Using License Use Management Runtime for Windows* and *Message Reference* books, in .HTM format for viewing with a Web browser.

Hardware Requirements

The hardware requirements for the various Windows platforms are shown in Table 8.

Table 8. Hardware Requirements

Platform	Minimum		Suggested	
	Processor	RAM	Processor	RAM
Windows 95 Windows 98 Windows NT Windows 2000 Windows Terminal Server	133 MHz Pentium	32 MB	233 MHz Pentium II or higher	64 MB

Before installing, make sure that there is enough disk space for the features, as shown in Table 9.

Table 9. Disk Space Requirements

Feature	Requirement
Runtime	18 MB
Communications	9 MB
Documentation	3 MB

Software Requirements

- One of the following operating systems:
  - Windows 2000
  - Windows NT 4.0 with Service Pack 4, 5, or 6
  - Windows NT 4.0 Terminal Server Edition with Service Pack 4, 5, or 6

  - Windows 95
  - Windows 98
- For all supported Windows platforms:
  - A device driver that complies with NDIS 4
  - The latest level of the device driver. To determine the latest level of IBM device drivers, go to:

    `http://www.networking.ibm.com`
  - The i4target executable. If LUM Runtime is not yet installed on the workstation, you can download i4target from the License Use Management Web site:

    `http://www.software.ibm.com/is/lum`

    or the vendor of a license-enabled product can supply it to you.
- Depending on your operating system or operating environment, one or more of the transport protocols shown in Table 10:

Table 10. Transport Protocols

Platform	**TCP/IP**	**NetBIOS**	**IPX**
Windows 2000	✓	✓	✓
Windows NT 4.0	✓	✓	✓
Windows NT 4.0 Terminal Server Edition	✓	✓	✓
Windows 95	✓		✓
Windows 98	✓		✓*

* Windows 98 Second Edition only

If the prerequisites are not met, the hardware-based target ID is not generated.

Protocol Configuration Requirements

After installation only the TCP/IP protocol is available. If you use the configuration tool command-line interface to configure, the tool handles protocol configuration. If you use the GUI to configure, then to use the protocols other than TCP/IP, edit the Registry and change the values of the following parameters from False to True:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Gradient\DCE\Configuration\IPXSupport
HKEY_LOCAL_MACHINE\SOFTWARE\Gradient\DCE\Configuration\NetBIOSSupport
```

If you use NetBIOS as the transport protocol:

- Make sure that the NetBEUI protocol is installed.
- On a Windows NT system, on the Services page of the Network window select the NetBIOS interface and modify its properties so that the LAN adapter number of the Network Route starting with Nbf is set to 0.
- On a Windows 2000 system, set the value of Nbf to 0 by manually editing the registry.

Obtaining the LUM Runtime Code

Download the LUM Runtime code from the Web:

1. Download the self-extracting EXE file from:

   `http://www.software.ibm.com/is/lum`

2. Enter the name of the file as a command. This automatically starts the interactive installation procedure.

Deciding How to Install

Before you install, you need to decide:

- Whether to install directly on the target workstation or to copy the code images to a code server ("Installation Using a Code Server" on page 65) so that multiple users can use them for installation
- Whether to install interactively, using the installation GUI ("Installing LUM Runtime Interactively" on page 54) or noninteractively ("Installing LUM Runtime Noninteractively" on page 63)

Installation Considerations

Windows Installer is used for installing LUM Runtime. It is installed automatically during Windows 2000 installation.

If Windows Installer is not already installed on your Windows 95, Windows 98, Windows NT, or Windows Terminal Server workstation, it is automatically and silently installed the first time you run the LUM Runtime setup program.

On Windows 95, Windows 98, Windows NT, or Windows Terminal Server, if you are not sure whether Windows Installer is already installed, install LUM Runtime interactively. Do not install it silently or remotely.

When Windows Installer is installed on your workstation, the setup program installs LUM Runtime when you run the setup program.

When you install a new release of LUM Runtime over an earlier release, Windows Installer first silently deletes the earlier release. It then installs all, and only, those features that were previously installed. You cannot change the selection of features to be installed.

If LUM Runtime program files are damaged and need to be replaced, run the setup program again, and select the Modify or Repair option.

Installing LUM Runtime Interactively

To install LUM Runtime interactively, follow these steps:

1. In Windows NT, log on as Administrator.
2. Shut down any currently running programs, including virus scanning programs.
3. Start the installation program:
   - If you are installing LUM Runtime on Windows 95, Windows 98, Windows NT, or Windows 2000, switch to the directory to which you have downloaded the code, and enter:

     setup

Installing Windows Installer

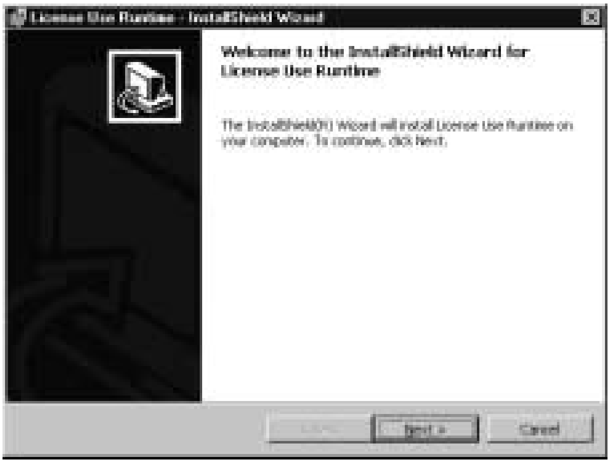
If you are installing LUM Runtime on Windows 95, Windows 98, Windows NT or Windows Terminal Server and Windows Installer is not already installed, it is installed now. You are then requested to restart the system. LUM Runtime has not yet been installed.

On Windows 95 and Windows 98, when the system has restarted, repeat this step 3 to install LUM Runtime.

On Windows NT, when the system has restarted, LUM Runtime installation continues automatically.

- If you are installing LUM Runtime on Windows Terminal Server:
  a. Click **Start** --> **Settings** --> **Control Panel** --> **Install** --> **Next**.
  b. In the **Command line for installation program** field, type the drive identifier, path, and name of the installation program (**setup**).
  c. Click **Finish**.

     The Welcome window is displayed (see Figure 14).



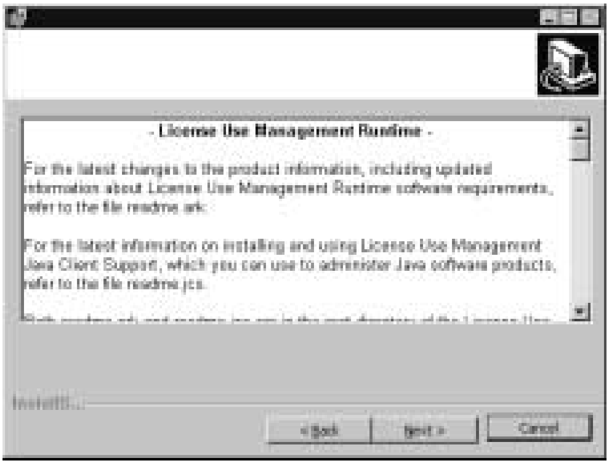
Welcome to the InstallShield Wizard for License Use Runtime

The InstallShield(R) Wizard will install License Use Runtime on your computer. To continue, click Next.

Figure 14. Welcome Window

4. In the Welcome window, select **Next>**.

   The READ.ME file for LUM Runtime is displayed (see Figure 15 on page 56).



- License Use Management Runtime -

For the latest changes to the product information, including updated information about License Use Management Runtime software requirements, refer to the file readme.ark.

For the latest information on installing and using License Use Management Java Client Support, which you can use to administer Java software products, refer to the file readme.jcs.

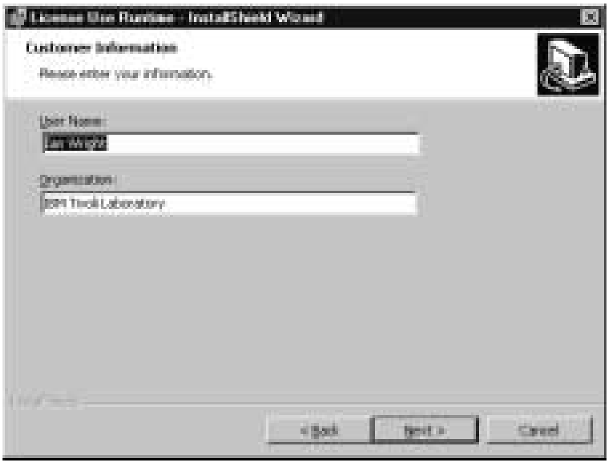
Figure 15. READ.ME Window. The text displayed on your monitor may differ from that shown here.

This file contains an overview of the installation procedure.

Availability of Java Client Support: The second paragraph of the displayed document refers to Java Client Support, which is available only under AIX, Solaris, and Windows NT; it is not available on the Windows Terminal Server, Windows 95, Windows 98, or Windows 2000 platforms.

5. When you have read the document, select **Next>**.

   The Customer Information window is displayed (see Figure 16).



License Use Runtime - InstallShield Wizard

Customer Information

Please enter your information.

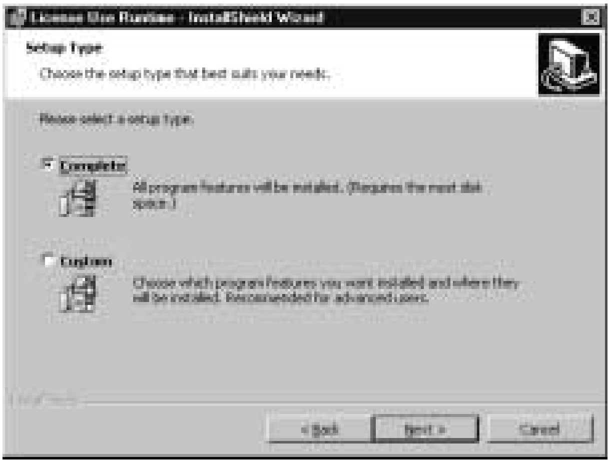
User Name: [illegible]

Organization: IBM Tivoli Laboratory

Figure 16. Customer Information Window

6. Enter your customer details, then click **Next>**.

The Setup Type window is displayed (see Figure 17).



License Use Runtime - InstallShield Wizard

Setup Type

Choose the setup type that best suits your needs.

Please select a setup type.

● Complete

All program features will be installed. (Requires the most disk space.)

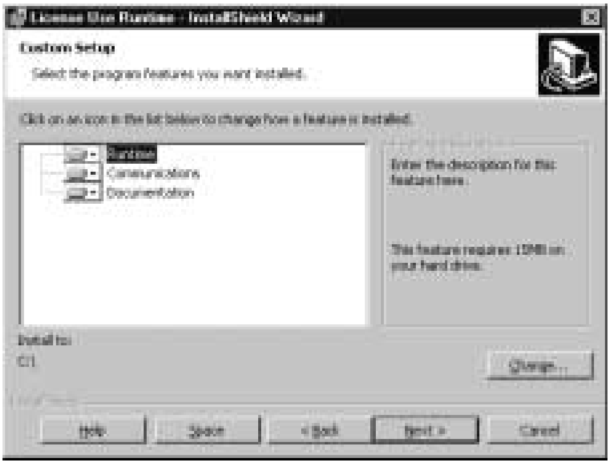
○ Custom

Choose which program features you want installed and where they will be installed. Recommended for advanced users.

Figure 17. Setup Type Window

7. Choose **Complete** or **Custom** setup. If you choose Custom setup, you will be able to specify the disk drive on which to install LUM Runtime and the features to install.
8. Click **Next>**.

   The Custom Setup window is displayed (see Figure 18).



License Use Runtime - InstallShield Wizard

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

Enter the description for this feature here.

This feature requires 15MB on your hard drive.

Install to:
C:\

Figure 18. Custom Setup Window

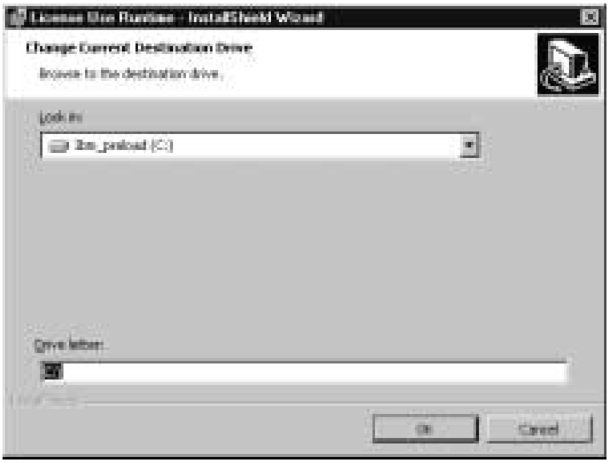
By default, all the features will be installed on the drive on which the operating system is installed. If you want to install all the features on that drive, go to step 14 on page 59.

If you want to install the features on a different drive, go to step 9 .

If you want to install only one or two features, go to step 12.

9. To change the drive on which to install the features, click **Change...**.

   The Change Current Destination Drive window is displayed (see Figure 19).



License Use Runtime - InstallShield Wizard

Change Current Destination Drive

Browse to the destination drive.

Look in:

ibm_preload (C:)

Drive letter:

C:

Figure 19. Change Current Destination Drive Window

10. Select the drive from the **Look in** drop-down list.

    Notes:

    a. Specify the drive only. The product is always installed in the `ifor` directory. Do not specify a folder name in the **Drive letter** field.

    b. If the Application Developer’s Toolkit is installed on the workstation, LUM Runtime is always installed on the same drive.

11. Click **OK**.

    The Custom Setup window is displayed again, showing the changed drive information.

    If you want to install all the features, continue at step 14 on page 59.

12. To not install a feature, click the icon ▭▾ beside its name.

    The feature selection menu is displayed (see Figure 20 on page 59).

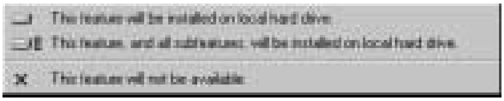

This feature will be installed on local hard drive.
This feature, and all subfeatures, will be installed on local hard drive.
This feature will not be available.

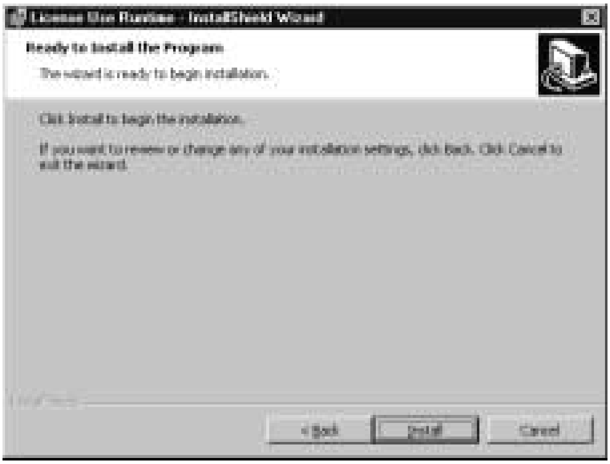
Figure 20. Feature Selection Menu

13. Select the choice **This feature will not be available**.

    The icon changes ✕ ▾ to indicate that the feature will not be installed.

14. Click **Next>**.

    The Ready to Install the Program window is displayed (see Figure 21).


License Use Runtime - InstallShield Wizard

Ready to Install the Program

The wizard is ready to begin installation.

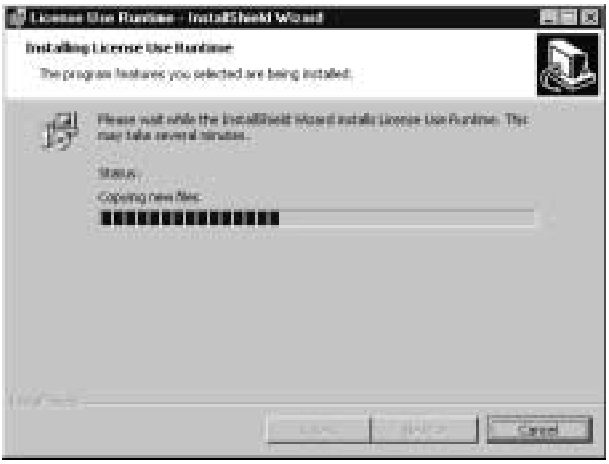
Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Figure 21. Ready to Install Program Window

15. Click **Next>**.

    The Installing License Use Runtime window is displayed (see Figure 22 on page 60), and LUM Runtime starts to copy files.



Installing License Use Runtime

The program features you selected are being installed.

Please wait while the InstallShield Wizard installs License Use Runtime. This may take several minutes.

Status:

Copying new files

Figure 22. Installing License Use Runtime Window

If you get the message:

```
Unable to update PATH system variable.
Add the following path manually:
c:\ifor\WIN\BIN;c:\ifor\WIN\BIN\en_US
```

where:

c Is the installation drive

Windows has not updated the PATH environment variable, because the sum of the length of the current environment variable plus the length of the environment variable would exceed 1024 bytes. See “Verifying and Updating the PATH Environment Variable” on page 64 for information about how to manually add the LUM Runtime path information.

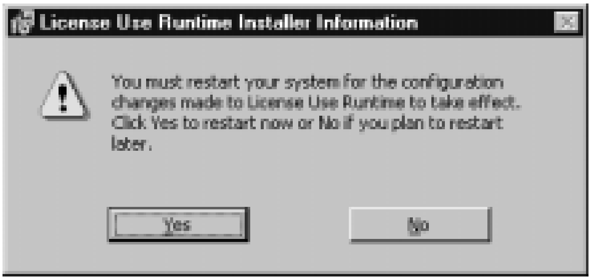
When installation is complete, the InstallShield Wizard Completed window is displayed.

16. Click **Finish**.

    If you installed LUM Runtime on a Windows NT or Windows 2000 workstation, you can start using it immediately.

    If you installed LUM Runtime features on a Windows Terminal Server workstation, go to step 18 on page 61.

    Otherwise, if you installed the features on a Windows 95 or Windows 98 workstation, the InstallShield Wizard Completed window is displayed (see Figure 23 on page 61).



You must restart your system for the configuration changes made to License Use Runtime to take effect. Click Yes to restart now or No if you plan to restart later.

Figure 23. InstallShield Wizard Completed Window or Windows 95 and Windows 98

17. Click **Yes** to restart Windows immediately, or **No** to restart Windows later.

    You must, in any case, restart your workstation before you can use LUM Runtime.

18. If you installed on a Windows Terminal Server workstation, run the **reglum.cmd** script to register the DLLs.

LUM Runtime is installed on your workstation, and a folder called License Use Runtime is added to the Programs menu.

After installation you must configure LUM Runtime as explained in "Chapter 4. Configuring LUM Runtime" on page 69.

Installing after an Uninstallation

If you uninstall 4.*x.y*, your existing license databases are not deleted. They remain in the directory `ifor\ls\conf`. If you then install LUM Runtime Version 4.6, the installation process renames the databases to have an extension of .old. If you want to use the databases with LUM Runtime Version 4.6, follow these steps:

1. Enter the following sequence of commands:

   ```
   i4cfg -stop
   cd ifor\ls\conf
   copy crpdb.dat.old crpdb.dat
   copy crpdb.idx.old crpdb.idx
   copy licdb.dat.old licdb.dat
   copy licdb.idx.old licdb.idx
   copy llmdb.dat.old llmdb.dat
   copy llmdb.idx.old llmdb.idx
   ```

2. Use an ASCII editor to create and save the file:

   ```
   c:\ifor\ls\conf\DBVersion
   ```

   that contains only the following text:

   ```
   451
   ```

   (*c* is the installation drive.)

3. Run the following migration options:
   a. `i4mig -r -lmdLic`
   b. `i4mig -r -c -lmdLic`
   c. `i4mig -r -n -lmdLic`

4. Reopen the file **DBVersion**. If it contains the string **460**, the migration has been successful. Go to step 6. If it contains the string **455**, continue at step 5.

5. Run the following migration options:
   a. `i4mig -r -lmdLic455`
   b. `i4mig -r -c -lmdLic455`
   c. `i4mig -r -n -lmdLic455`

6. Delete the DBVersion file.

Installing LUM Runtime Noninteractively

You can use the **setup** command to install LUM Runtime without being prompted.

You need to be present only to initiate the installation and to restart your workstation, if necessary, when the installation is complete. No questions are asked while the installation is running.

To install LUM Runtime noninteractively, follow these steps:

1. Make sure that the environment variable `I4_INSTALL_DRIVE` is set. To do this, look in **HKEY_LOCAL_MACHINE** --> **SYSTEM** --> **CurrentControlSet** --> **Control** --> **Session Manager** --> **Environment**.
2. To set the `I4_INSTALL_DRIVE` variable, follow these steps:
   a. Click **Start** --> **Settings** --> **Control Panel** --> **Environment**.
   b. In the System Variables box, add the following information:
      - In the **Variable** field, type:

        ```
        I4_INSTALL_DRIVE
        ```

      - In the **Value** field, type the installation drive letter; for example, **c:**.

   c. Click **OK**.
3. Check, in the file **`ifor.bat`**, whether the value of the **`ADDLOCAL`** parameter is suitable. If it is not, change it. This parameter identifies the features to be installed.

   If you are installing LUM Runtime over an older release of the product, all currently installed features are automatically upgraded.
4. In a new command prompt window, run **`ifor.bat`**.

   If you install or reinstall only some LUM Runtime features on a workstation on which more than one feature is already installed, the setup program silently installs or reinstalls those other preinstalled features.
5. When installation is complete, follow the instructions in “Chapter 4. Configuring LUM Runtime” on page 69.

Using the Installation Command

The following syntax shows the parameters accepted by the **setup** installation command:

```
setup /s/v"ADDLOCAL=Runtime,Communications,Documentation LUM_SILENT=1 /qn /l log_file"
```

where:

s Specifies silent installation

v Indicates that what follows is a command line parameter

ADDLOCAL
Specifies the features to be installed

LUM_SILENT=1
Specifies that installation is to run silently

qn Specifies that the user interfaces not to be displayed during installation

l ***log_file***
Specifies the log file in which installation information is to be logged

When the installation completes successfully, the message `Installation operation completed successfully` is written to the log file.

Verifying and Updating the PATH Environment Variable

In Windows NT, check whether the PATH environment variable contains the following information:

```
c:\ifor\WIN\BIN;c:\ifor\WIN\BIN\en_US
```

where:

c Specifies the installation drive

If the LUM Runtime path information has not been added, update the PATH environment variable manually, as follows:

1. Click **Start** --> **Settings** --> **Control Panel** --> **System** --> **Environment**.
2. If necessary, delete unused path information. The LUM Runtime path information requires 38 bytes. Remember to add any necessary delimiting semicolon to the end of the original PATH statement.
3. Add to the end of the PATH statement:

   ```
   ;c:\ifor\WIN\BIN;c:\ifor\WIN\BIN\en_US
   ```
4. Press the **Set** or **OK** push button.

The maximum length of the PATH environment variable in Windows NT is 1024 bytes. Therefore, if the sum of the length of the current path plus the length of the LUM Runtime PATH information would exceed 1024 bytes, Windows NT will not update the current PATH environment variable. You will then not be able to run LUM Runtime, until you manually update the PATH statement.

If LUM Runtime Fails to Start

In certain unusual circumstances, the LUM Runtime environment variables are set during installation, but are not propagated to the system. The product therefore fails to start. If you experience this situation, LUM Runtime select **Start -->** **Settings -->** **Control panel -->** **System -->** **Environment -->** **OK**. LUM Runtime should then start correctly.

Installation Using a Code Server

A *code server* is a workstation that contains the files to be used during the installation.

To install LUM Runtime on a non-Windows Terminal Server workstation, connect to the code images directory of the code server. Then switch to the directory `x:\runtime`, where `x` is the drive letter of the redirected drive and `\runtime` is the LUM Runtime directory name, and type `SETUP`. Type the command without parameters for an interactive installation, or with parameters as indicated in the section “Installing LUM Runtime Noninteractively” on page 63 for a noninteractive installation.

To install LUM Runtime on Windows Terminal Server:

1. Click **Start -->** **Settings -->** **Control Panel -->** **Install -->** **Next**.
2. In the **Command line for installation program** field, type the drive identifier, path, and name of the installation program (**setup**). Type the command without parameters for an interactive installation, or with parameters as indicated in the section “Using the Installation Command” on page 63 for a noninteractive installation.
3. Click **Finish**.

Proceed with the installation as explained in “Installing LUM Runtime Interactively” on page 54 or “Installing LUM Runtime Noninteractively” on page 63. When you have successfully installed LUM Runtime, a message indicates that the installation is complete.

If you have installed the Runtime feature of LUM Runtime under Windows 95 or Windows 98, shut down and restart the workstation.

If you installed LUM Runtime on Windows Terminal Server, run the **reglum** script to register the LUM Runtime dynamic link libraries.

For non-Windows Terminal Server and for Windows Terminal Server installation, follow the instructions in “Chapter 4. Configuring LUM Runtime” on page 69.

Uninstalling LUM Runtime Features

You can remove features of LUM Runtime, or the whole product, from your computer.

Remove LUM Runtime from your workstation, as follows:

1. Click **Start** --> **Settings** --> **Control Panel** --> **Add/Remove Programs**.

   The Add/Remove Programs window is displayed.
2. Select **License Use Runtime**.
3. On Windows 95, Windows 98, Windows NT, or Windows Terminal Server, click **Add/Remove**.

   On Windows 2000, click **Change**.

   The LUM Runtime installer’s Welcome window is displayed.
4. Click **Next**.

   The Program Maintenance window is displayed.
5. Click **Modify**.
6. Click **Next**.

   The Custom Setup window is displayed.
7. Select the features you want to delete. The feature selection menu is displayed.
8. Click **This feature will not be available** for each feature to be deleted.
9. Click **Next**.

   The Ready to Modify the Program window is displayed.
10. Click **Install**.

    When uninstallation is complete, the InstallShield Wizard Completed window is displayed.
11. Click **Finish**.

The selected features have been removed from your computer.

Uninstalling LUM Runtime

Remove LUM Runtime from your workstation, as follows:

1. Click **Start** --> **Settings** --> **Control Panel** --> **Add/Remove Programs**.

   The Add/Remove Programs window is displayed.
2. Select **License Use Runtime**.

3. On Windows 95, Windows 98, Windows NT, or Windows Terminal Server, click **Add/Remove**.

   On Windows 2000, click **Change**.

   The LUM Runtime installer's Welcome window is displayed.

4. Click **Next**.

   The Program Maintenance window is displayed.

5. Click **Remove**.

6. Click **Next**.

   The Remove the Program window is displayed.

7. Click **Next**.

   The Uninstalling LUM Runtime window is displayed.

   When uninstallation is complete, the InstallShield Wizard Completed window is displayed.

8. Click **Finish**.

Note that the uninstallation process does not remove configuration files or other files created after installation. To remove such files, after the uninstallation process has finished, delete the `ifor` directory and all its contents.

Working with Java Applications and Applets

Java Client Support 4.5.8 runs on LUM Runtime 4.6.0.

Java applications and applets can specify the URL of a server on which Java Client Support classes are installed. For an applet, this can be different from the URL of the server from which it was downloaded.

Chapter 4. Configuring LUM Runtime

The scenarios in this chapter describe how to set up your LUM Runtime environment.

LUM Runtime is not required for non-runtime nodelocked workstations (see “License Use Management at a Glance” on page 1).

Setting Up Your Servers and Clients

After you install LUM Runtime, you must configure the program on each workstation.

The configuration tool has a GUI and a command line interface. The information you provide is saved in a configuration file. When you start LUM Runtime, it uses the information in this file to direct the behavior of the local system in the licensing environment. See “Appendix A. LUM Runtime Configuration File” on page 209 for reference information on the configuration file.

Configuring to Handle Nodelocked Licenses

To handle only products with nodelocked licenses, you need only configure the workstation as a nodelocked license server (“Scenario 1: Configuring a Standalone Nodelocked License Server” on page 75) and have the nodelocked license server up and running. Every workstation is automatically configured as a nodelocked license server when LUM Runtime is installed. You do not have to do any configuration unless you want to change the default configuration.

If a workstation configured as a nodelocked license server is in a network, instances of the basic license tool on other workstations can administer licenses on the nodelocked license server. With a little additional configuration (“Scenario 2: Configuring a Nodelocked License Server in a Network” on page 77), you can run the basic license tool on the nodelocked license server workstation and administer licenses stored on remote network license servers, nodelocked license servers, and the central registry.

Configuring to Handle Network Licenses

To handle products with network licenses, you must configure at least one network license server (“Scenario 3: Configuring a Network License Server” on page 80) and configure each client as a network license client (“Scenario 5: Configuring a Network License Client” on page 89). If you use products with customer-managed or reservable licenses, you must also configure one server as the central registry ("Scenario 4: Configuring the Central Registry License Server" on page 86).

Determining the Configuration Required

You can configure a workstation to play more than one role in your licensing environment. For example, if you configure the same workstation as a nodelocked license server, a network license server, and the central registry license server, that workstation can handle all types of licensed products.

If you know the types of licenses used by the license-enabled products in your environment, Table 11 will help you to determine how to configure:

- The license servers for the application
- The workstations that will request licenses for the application

See the application documentation for information that might affect your configuration. The *enrollment certificate file* that you receive from the vendor shows the type of license.

Table 11. Configuration Required to Support All Types of Licenses

License Type	**License Requester**	**License Server**
Simple nodelocked (non-runtime-based enabling) (note 1)	License-enabled application	None
Simple nodelocked (runtime-based enabling) (note 1)	License-enabled application	Nodelocked license server on the same workstation as the application
Use-once nodelocked	License-enabled application	Nodelocked license server on the same workstation as the application
Concurrent nodelocked	License-enabled application	Nodelocked license server on the same workstation as the application
Per-server	License-enabled application	Nodelocked license server on the same workstation as the application
Use-once	Network license client	Network license server (note 2), and central registry license server (note 3)
Concurrent	Network license client	Network license server (note 2) and central registry license server (note 3)
Per-seat	Network license client and nodelocked license server	Central registry license server (note 3)

Table 11. Configuration Required to Support All Types of Licenses (continued)

License Type	License Requester	License Server
Reservable	Network license client and nodelocked license server (note 4)	Network license server (note 2) and central registry license server (note 3)

Notes

1. **If the vendor enabled the product with simple nodelocked licenses and delivered the licenses to you in a compound network password, you must also**:
   - Configure a network license server, where you will install the compound password, and
   - Configure the local workstation (where the nodelocked license is to be installed) as a network license client of that server
2. You can configure one or more network license servers.
3. You can configure only one central registry license server. For customer-managed use products, the central registry is required. It enables you to update the count of product licences, implement the hard stop or soft stop policy, or track the high-water mark. Note that because per-seat licenses are always customer-managed, they require the central registry. For reservable licenses, the central registry is required if you want to be able to reserve licenses for specific users.
4. For reservable licenses, the nodelocked license server is required for the end user to get a reserved license.

Before You Configure

Before you begin the configuration process, for every workstation you are going to configure, you need to decide which roles it will play and how you plan to set up direct binding or namespace binding. You might also decide to override some of the configuration defaults. The actions you can take during configuration are summarized in Table 12. Check the table for all the roles your workstation will play, and make all the indicated decisions before you start configuration.

Table 12. Configuration Options

Configuration Options	Nodelocked License Server	Network License Client	Network License Server	Central Registry License Server
Customize selection of information logged or accept default? ("Customizing Log Information" on page 72)	Yes		Yes	Yes
Customize selection of information logged or accept default? ("Customizing Log Information" on page 72)	Yes		Yes	Yes
Start license servers at system startup (default is no)? ("Automatically Starting License Servers" on page 73)	Yes		Yes	Yes

Table 12. Configuration Options (continued)

Configuration Options	Nodelocked License Server	Network License Client	Network License Server	Central Registry License Server
Disable remote administration of network license servers (default is no)? ("Disabling Remote Administration" on page 73)			Yes	
Disable remote administration of nodelocked license servers (default is no)? ("Disabling Remote Administration" on page 73)	Yes			
Set up direct binding ("Configuring Direct Binding" on page 73) Prepare a list of nodelocked license servers for remote administration Have your direct binding servers list ready Know which workstation is the central registry Change default direct binding ports?	Yes (see note) Yes (see note) Yes (see note) Yes (see note)	Yes Yes Yes Yes	Yes Yes Yes Yes	Yes Yes Yes Yes

Note: Applicable if you plan to run the basic license tool and administer licenses on other workstations from this nodelocked license server

Customizing Log Information

For any license server, you can customize the selection of events that are logged, and the location of the log file. Note that if you want to change the location of the log from the default to a path you choose, the directory you specify must already exist. Otherwise, you will lose the logging function.

The following events can be logged:

All events Includes all the events in the list.

Errors Describes server errors that do not stop the server, but return a status code and a message. This is logged by default.

License timeout
Tells you that the server has canceled the request for a license because the check period expired. This is not logged by default.

License wait Tells you when a license request cannot be satisfied because no licenses are available, and the user is added to a queue. This is not logged by default.

License checkin
Tells you when a licensed product has sent a check-in call to the server to notify that the product is running. This is not logged by default.

License grant/release
Tells you when a license was granted or released. This is not logged by default.

Vendor added/deleted
Tells you when a product of a new vendor was registered or deleted. This is logged by default.

Vendor messages
Provides the log messages the vendor inserted in the enabled product. This is logged by default.

Product added/deleted
Tells you when a new product was registered or deleted. This is logged by default.

Server start/stop
Logs the successful start or stop of the license server. This is not logged by default.

Automatically Starting License Servers

During configuration of any license server, you can specify that license servers should start automatically when you start the workstation. Otherwise, you must remember to start the services manually after configuration and before using the basic license tool or any enabled applications.

Disabling Remote Administration

When you configure a network or nodelocked license server, you can specify that licenses stored on that server cannot be administered from any other license server.

Configuring Direct Binding

When you configure a network license server, network license client, or central registry license server that is to be part of a direct binding environment, you must have your direct binding servers list ready.

When you configure a nodelocked license server, network license server, or central registry license server that is to be part of a direct binding environment, you must also have ready a list of nodelocked license servers whose licenses you want to administer remotely from this workstation.

You will enter the hostnames or network addresses of all the servers in the list, except for the nodelocked license server on the local workstation, which is added to the list automatically. You will also designate which server, if any, is the central registry.

When you configure the servers in the direct binding servers list, and the clients that will use them, be sure you define exactly the same set of servers on each.

You can change the default port numbers for nodelocked license servers, network license servers, and the central registry license server. Do not change the defaults unless they are already in use by other applications.

If you are running Java Client Support on the same workstation and want to change the direct binding list:

1. Stop LUM Runtime and the Web server.
2. Change the direct binding list.
3. Restart LUM Runtime and the Web server.

Using the Configuration Tools

To configure LUM Runtime, you can use one of the following tools:

- Configuration tool graphical user interface
- Configuration tool command line interface

Using the Configuration Tool GUI

If the LUM Runtime graphical user interface is installed on your workstation, you can use it to configure your workstation.

Start the configuration tool by one of the following means:

- Click **Start** --> **Programs** --> **License Use Runtime** --> **Configuration Tool**.
- Enter the `i4cfg` command with no parameters.

In the **License Use Runtime** folder, click the **Configuration Tool** icon. Alternatively, start the configuration tool by entering the `i4cfg` command with no parameters.

A configuration tool notebook appears. Follow the steps of one or more of the configuration scenarios in this section, depending on the role of your workstation in your licensing environment.

Each of the configuration scenarios in this chapter shows how to configure a workstation to play only one role in the licensing environment. If the workstation is to play two or more roles, check all the applicable roles on the **Configure As** page of the configuration tool notebook.

If you configure a specific page of the notebook more than once, the most recent configuration takes effect.

Using the Configuration Tool Command-Line Interface

You can use the i4cfg command to accomplish the same configuration tasks explained in the scenarios in this chapter. At the end of each scenario, a section named “Command-Line Equivalent” shows the i4cfg commands that correspond to the GUI scenario.

For details of the `i4cfg` command, see “i4cfg - Configuration Tool” on page 156.

Scenario 1: Configuring a Standalone Nodelocked License Server

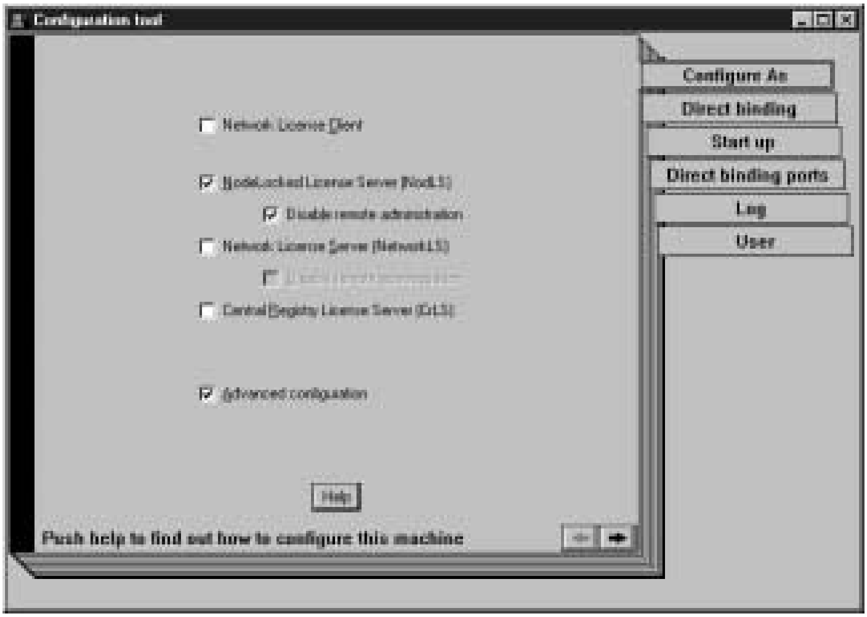
Use this scenario to configure a nodelocked license server without setting up any network connections to other LUM Runtime servers.

Use “Scenario 2: Configuring a Nodelocked License Server in a Network” on page 77 if you want to run the basic license tool and administer licenses on other license servers from this workstation.

A nodelocked license server can use licenses of the types shown in Table 11 on page 70. By default, every workstation is configured as a nodelocked license server. You need to perform this configuration only if you want to change the default configuration.

To configure a workstation as a standalone nodelocked license server after installation, perform the following steps:

1. Double-click the **Configuration Tool** icon in the LUM Runtime folder. The Configuration tool notebook is displayed.
2. On the **Configure As** page, select **Nodelocked License Server** and **Advanced Configuration**. The notebook is shown in Figure 24 on page 76.



☐ Network License Client

☑ NodeLocked License Server (NodLS)

☑ Disable remote administration

☐ Network License Server (NetworkLS)

☐ [illegible]

☐ Central Registry License Server (CrLS)

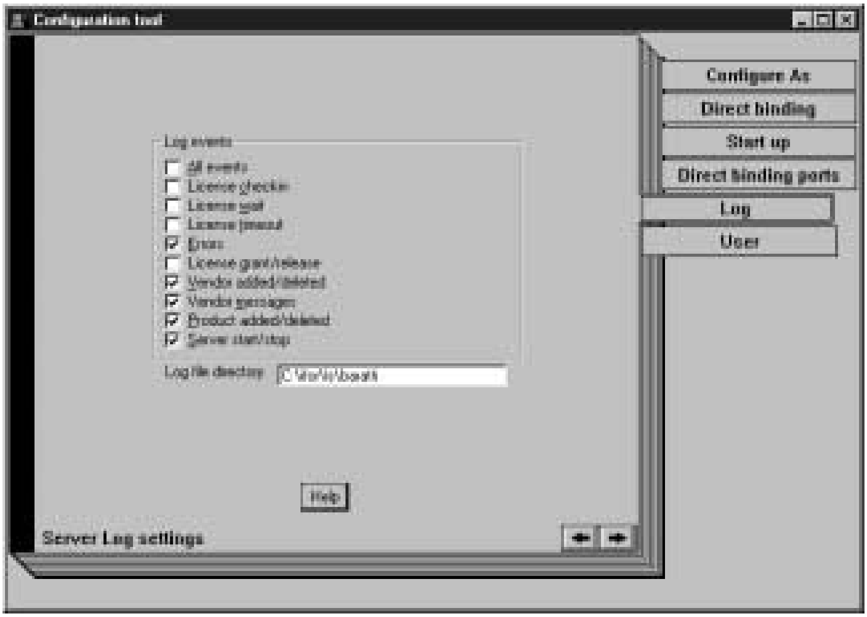
☑ Advanced configuration

Push help to find out how to configure this machine

Figure 24. Configuration Tool Notebook - Standalone Nodelocked License Server

For the type of configuration you are doing, ignore the **Direct binding**, **Direct binding ports**, and **User** pages (if they appear in the notebook) and the **Disable Remote Administration** check box.

3. On the **Start up** page, select **Start services at system startup** to start the nodelocked license server when you power on the workstation. This option is available on Windows NT and Windows 2000 only.
4. On the **Log** page, select the events you want to be logged and specify where you want the log to be kept, as shown in Figure 25 on page 77.



Log events

- □ All events
- □ License checkin
- □ License wait
- □ License timeout
- ☑ Errors
- □ License grant/release
- ☑ Vendor added/deleted
- ☑ Vendor messages
- ☑ Product added/deleted
- ☑ Server start/stop

Log file directory: C:\ifor\ls\baratti

Figure 25. Configuration Tool Notebook - Log Page

If you change the location of the log from the default directory to a directory of your own choice, that directory must already exist. Otherwise, the logging function will be lost. The logged events are stored in the files `llmlg`*nn*`_`, where *nn* assumes values from 00 to 99. When a file is full, according to a maximum number of lines fixed in the configuration file, a new log file is started.

5. Select **Close** from the system menu at the upper left corner of the notebook. A message is displayed to confirm that you are ready to save your choices.

Command-Line Equivalent

To configure the standalone nodelocked license server:

```
i4cfg -a n -S a -e evmps c:\ifor\ls\baratti
```

Scenario 2: Configuring a Nodelocked License Server in a Network

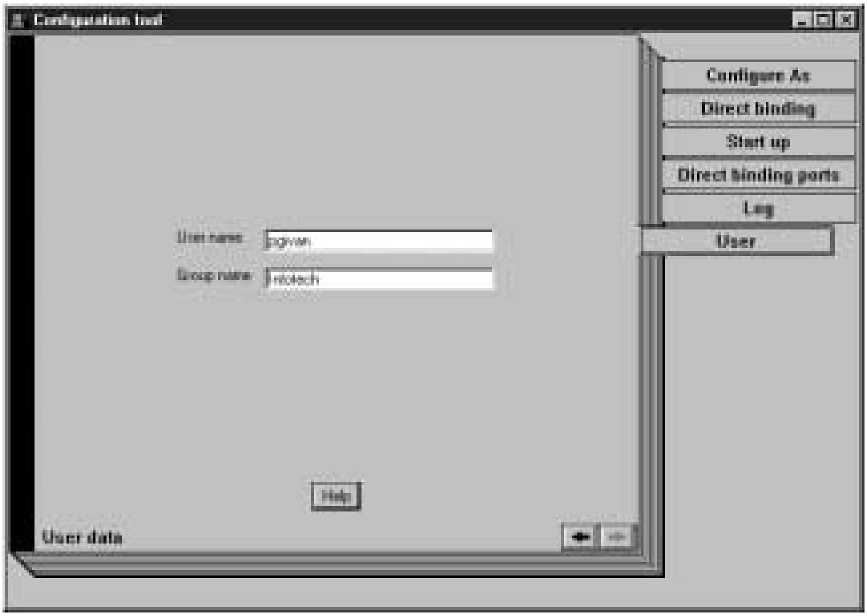
Use this scenario to configure a nodelocked license server, making it part of a direct binding server list. Configuring in this way, you can:

- Run the basic license tool and administer licenses on other license servers from this workstation
- Allow instances of the basic license tool on other workstations to administer nodelocked licenses on this workstation

A nodelocked license server can use licenses of the types shown in Table 11 on page 70 .

To configure a workstation as a nodelocked license server with remote administration, after installation, perform the following steps:

1. Double-click the **Configuration Tool** icon in the License Use Runtime folder. The Configuration Tool notebook is displayed.
2. On the **Configure As** page, select **Nodelocked License Server** and **Advanced Configuration**.
3. Make sure that the **Disable remote administration** box is unchecked, to enable instances of the basic license tool on other workstations to administer nodelocked licenses on this workstation.
4. To change the defaults, complete the **Start up** and **Log** pages, as explained in “Scenario 1: Configuring a Standalone Nodelocked License Server” on page 75.
5. Select the **User** tab. The User section is displayed, as shown in Figure 26.



User name: pgivan

Group name: nlotech

Figure 26. Configuration Tool Notebook - User Section

The user name is the logon user ID in effect when you do the configuration. Enter a group name if you want to change the default.

Note:

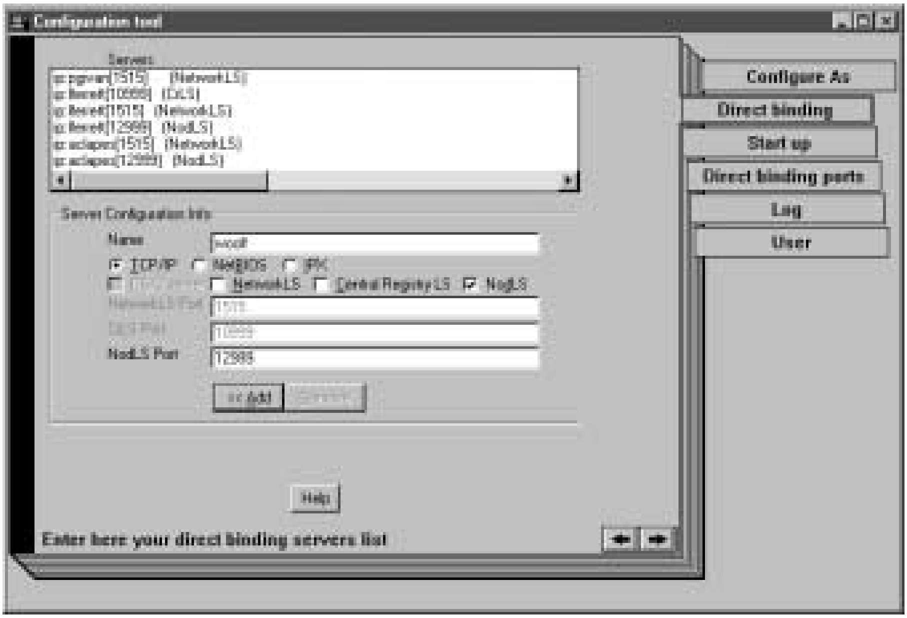
The LUM Runtime group name is not the same as the Windows NT group name.

LUM Runtime treats the user name as the name of the user who requests and uses licenses on this workstation. The user name is displayed as part of various reports about license usage. It is also used to reserve reservable licenses for a specific user. The group name is used to reserve reservable licenses for a specific group.

Use the remaining pages of the notebook to establish remote connections between this server and other LUM Runtime license servers. From this workstation, you will be able to use the basic license tool to administer licenses on those servers.

6. Select the **Direct binding** tab.

   The Direct binding section is displayed, as shown in Figure 27.



ip:pgivan[1515] (NetworkLS)
ip:lbreeit[10999] (CrLS)
ip:lbreeit[1515] (NetworkLS)
ip:lbreeit[12999] (NodLS)
ip:aclapes[1515] (NetworkLS)
ip:aclapes[12999] (NodLS)

Name: woolf

● TCP/IP ○ NetBIOS ○ IPX

□ ☐ NetworkLS ☐ Central Registry LS ☑ NodLS

NetworkLS Port: 1515

CrLS Port: 10999

NodLS Port: 12999

Enter here your direct binding servers list

Figure 27. Configuration Tool Notebook - Direct Binding Section

On this page you specify all the network license servers and nodelocked license servers whose licenses you will administer remotely from this workstation. Do not include this nodelocked license server itself. If the central registry license server, a network license server, and a nodelocked license server run on the same workstation, include all the servers whose licenses you want to administer.

In this example, the administrator performs the following steps for each server:

a. Select the **TCP/IP** radio button
b. In the **Name** field, enter the TCP/IP host name of the workstation you are adding to the Servers list. Note that the server name is case-sensitive.
c. Check **NodLS**, **NetworkLS**, or **Central Registry LS**, or any combination, depending on the roles the workstation plays in the network.
d. Leave the default values in the **NetworkLS Port, CrLS Port**, and **NodLS Port** fields.

   Note: For platforms other than Windows 95 or Windows 98, if NetBIOS is selected, the default values are, respectively, 115, 109, 129. If you change them, use values between 32 and 254.

   If NetBIOS or IPX is selected, the protocol must be configured as explained under "Protocol Configuration Requirements" on page 53.
e. Select the **<<Add** push button to add the server to the **Servers** list.

7. If you want to change the default direct binding port numbers, use the **Direct binding ports** page. If you set the direct binding ports for a particular protocol to **0**, the server does not listen on that protocol.
8. Select **Close** from the system menu in the upper left corner of the notebook. A message is displayed to confirm that you are ready to save your choices.
9. To run the configured processes, see "Starting and Listing Your Processes" on page 93.

The workstation can now be used as a nodelocked license server. The administrator can run the basic license tool on the workstation and can administer licenses on remote LUM Runtime servers.

Command-Line Equivalent

To configure the nodelocked license server in a network:

```
i4cfg -a n -S a,n -e evmps -l c:\ifor\ls\baratti
-b "'network ip:pgivan ip:lferrett ip:aclapes'
'nodelocked ip:lferrett ip:aclapes ip:woolf' 'registry ip:lferrett'"
```

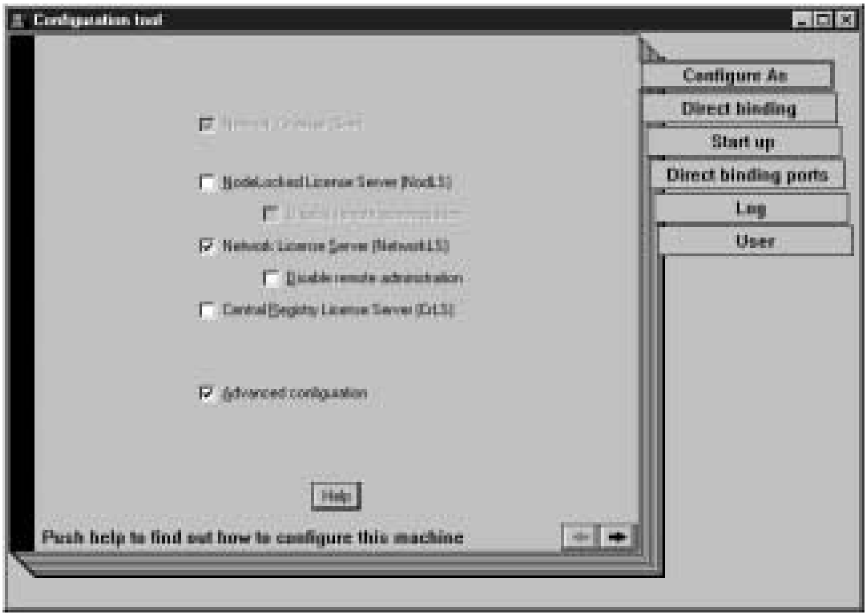
Scenario 3: Configuring a Network License Server

This scenario shows how the administrator configures LUM Runtime as a network license server (in this example, **pgivan**), making it part of a direct binding server list or a namespace binding cell.

When you configure a workstation as a network license server, you can use licenses of the types shown in Table 11 on page 70. You can also use the basic license tool to administer licenses on remote license servers in the network.

To configure the network license server, after installation of LUM Runtime:

1. Double-click the **Configuration Tool** icon in the LUM Runtime folder. The Configuration Tool notebook is displayed.
2. On the **Configure As** page, select **Network License Server**. Note that **Network License Client** is then automatically checked. Leave **Disable Remote Administration** unchecked. Check **Advanced Configuration**, which adds the **Direct binding ports**, **Log** and **User** pages to the notebook. The **Configure As** page is shown in Figure 28.



☑ Network License Client

☐ NodeLocked License Server (NodLS)

☐ [illegible]

☑ Network License Server (NetworkLS)

☐ Disable remote administration

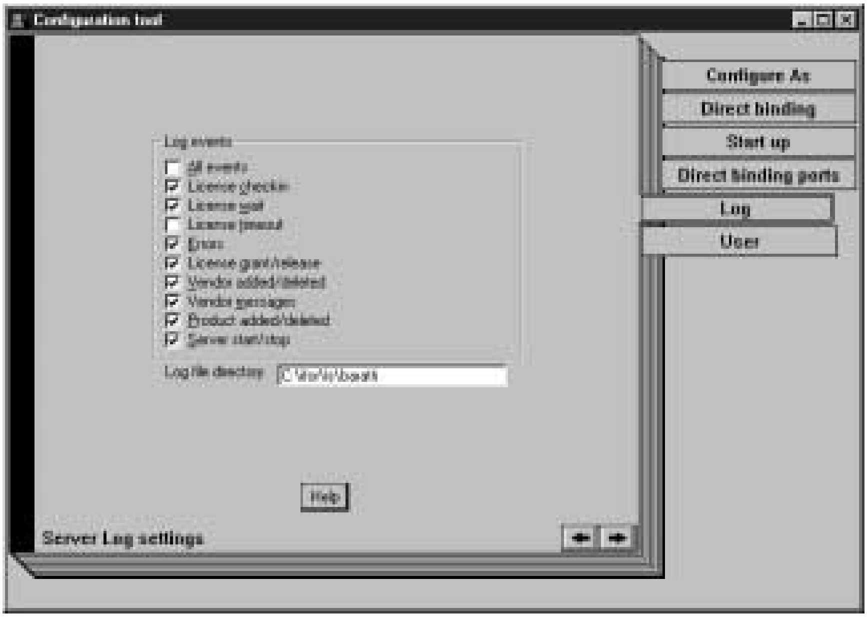
☐ Central Registry License Server (CrLS)

☑ Advanced configuration

Push help to find out how to configure this machine

Figure 28. Configuration Tool Notebook - Network License Server

3. On the **Start up** page, select **Start services at system startup** to start the network license server when you power on the workstation. This option is available on Windows NT and Windows 2000 only.
4. On the **Log** page, select the events you want to be logged and specify where you want the log to be kept, as shown in Figure 29 on page 83.



Log events

- □ All events
- ☑ License checkin
- ☑ License wait
- □ License timeout
- ☑ Errors
- ☑ License grant/release
- ☑ Vendor added/deleted
- ☑ Vendor messages
- ☑ Product added/deleted
- ☑ Server start/stop

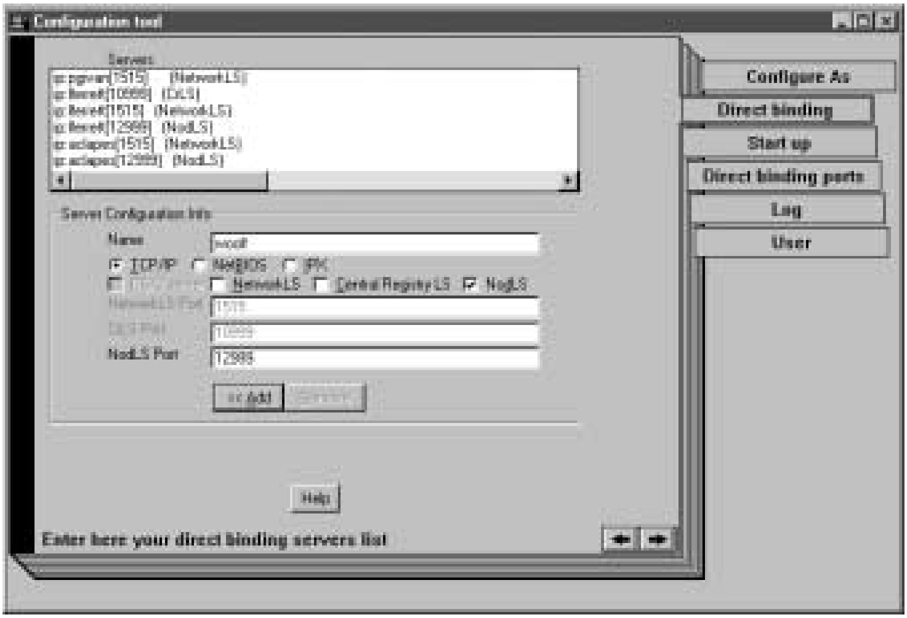
Log file directory: C:\ifor\ls\basrh

Server Log settings

Figure 29. Configuration Tool Notebook - Log Page

The logged events are stored in the files `logdb`*nn*_, where *nn* assumes values from 00 to 99. When a file is full, according to a maximum number of lines fixed in the configuration file, a new one is started.

5. Select the **Direct binding** tab. The **Direct binding** section is displayed, as shown in Figure 30 on page 84.



Configuration tool
Servers
ip:cgrvan[1515] (NetworkLS)
ip:lbreett[10999] (CrLS)
ip:lbreett[1515] (NetworkLS)
ip:lbreett[12999] (NodLS)
ip:aclapes[1515] (NetworkLS)
ip:aclapes[12999] (NodLS)
Server Configuration Info
Name
woolf
● TCP/IP ○ NetBIOS ○ IPX
□ NetworkLS □ Central Registry LS ☑ NodLS
NetworkLS Port
1515
CrLS Port
10999
NodLS Port
12999
<< Add
Remove
Help
Configure As
Direct binding
Start up
Direct binding ports
Log
User
Enter here your direct binding servers list

Figure 30. Configuration Tool Notebook - Direct Binding Section

On this page you specify all the network license servers in the direct binding servers list. You must include this network license server that you are configuring, and the central registry license server, if any. If the central registry license server and a network license server run on the same workstation, make sure you include both servers in the list.

As you configure the servers in the direct binding servers list, define exactly the same set of servers on each.

Specify also any remote nodelocked license servers whose licenses you want to administer from this workstation. Do not include the nodelocked license server on this workstation: you can administer local nodelocked licenses automatically, without specifying direct binding.

In this example, the administrator performs the following steps for each server:

a. Select the **TCP/IP** radio button.

b. In the **Name** field, enter the TCP/IP host name of the workstation you are adding to the **Servers** list. Note that the server name is case-sensitive.

c. If the server being added is a network license server, leave only the **NetworkLS** check box selected.

d. Check **NodLS**, **NetworkLS**, or **Central Registry LS**, or any combination, depending on the roles the workstation plays in the network.

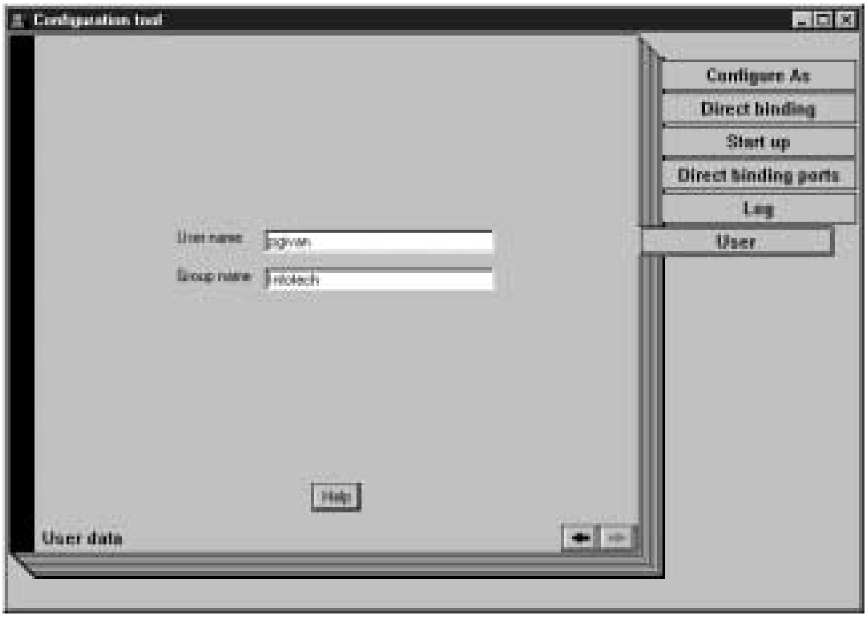
e. Leave the default values in the **NetworkLS Port**, **CrLS Port,** and **NodLS Port** fields.

Note: For platforms other than Windows 95 or Windows 98, if NetBIOS is selected, the default values are, respectively, 115, 109, 129. If you change them, use values between 32 and 254.

If NetBIOS or IPX is selected, the protocol must be configured as explained under "Protocol Configuration Requirements" on page 53.

f. Click **<<Add** to add the server to the **Servers** list.

6. If you want to change the default direct binding port numbers, use the **Direct binding ports** page. If you set the direct binding ports for a particular protocol to **0**, the server does not listen on that protocol.
7. Select the **User** tab. The User section is displayed, as shown in Figure 31.



User name: pgivan

Group name: intotech

Figure 31. Configuration Tool Notebook - User Section

The user name is the logon user ID in effect when you do the configuration. Enter a group name if you want to change the default.

Note:

The LUM Runtime group name is not the same as the Windows NT or Windows 2000 group name.

LUM Runtime treats the user name as the name of the user who requests and uses licenses on this workstation. The user name is displayed as part of various reports about license usage. It is also used to reserve reservable licenses for a specific user. The group name is used to reserve reservable licenses for a specific group.

8. Select **Close** from the system menu in the upper left corner of the notebook. A message is displayed to confirm that you are ready to save your choices.
9. To run the configured processes, see “Starting and Listing Your Processes” on page 93.

Now the workstation can be used as a network license server, to install and grant product licenses and monitor their usage.

Command-Line Equivalent

To configure the network license server:

```
i4cfg -a s -S a,s -e cwegvmps -l c:\ifor\ls\baratti -g Infotech
-b "'network ip:pgivan ip:lferrett ip:aclapes'
'nodelocked ip:lferrett ip:aclapes ip:woolf' 'registry ip:lferrett'"
```

Scenario 4: Configuring the Central Registry License Server

This scenario shows how the administrator configures the central registry license server (in this example, **pgivan**), making it part of a direct binding servers list.

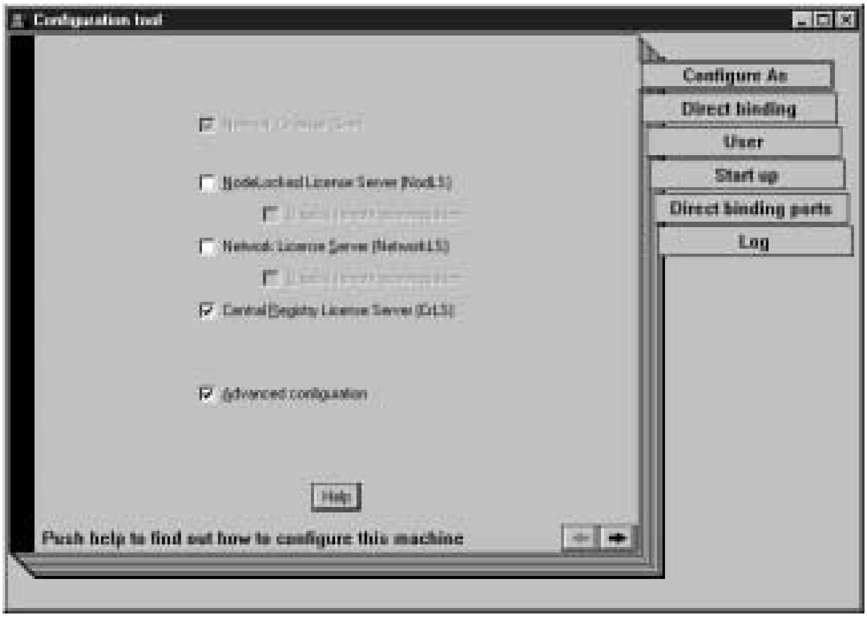
Configure only one central registry license server. Plan carefully where to configure it; once you start it, you cannot move it. (See “Planning the Central Registry” on page 42.)

Configuring a workstation as the central registry license server makes it possible to use licenses of the types shown in Table 11 on page 70. It also makes it possible to use the basic license tool to administer licenses on remote license servers in the network.

To configure the central registry license server after installation of LUM Runtime:

1. Double-click the **Configuration Tool** icon in the LUM Runtime folder. The Configuration Tool notebook is displayed.

2. On the Configure As page, select **Central Registry License Server**. **Network License Client** is then automatically checked. The **Configure As** page is shown in Figure 32.



☑ Network License Client

☐ NodeLocked License Server (NodLS)

☐ [illegible]

☐ Network License Server (NetworkLS)

☐ [illegible]

☑ Central Registry License Server (CrLS)

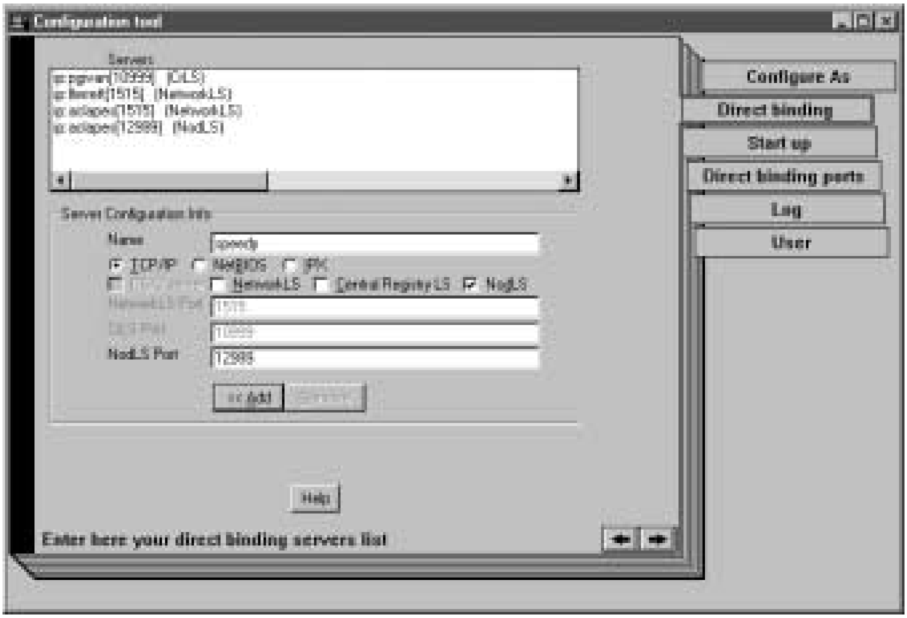
☑ Advanced configuration

Push help to find out how to configure this machine

Figure 32. Configuration Tool Notebook - Central Registry License Server

3. On the **Start up** page, select **Start services at system startup** to start the central registry license server when you power on the workstation. This option is available on Windows NT and Windows 2000 only.
4. Select the **Direct binding** tab.

   The **Direct binding** section is displayed, as shown in Figure 33 on page 88.



Servers

ip:pgvan[10999] (CrLS)
ip:florest[1515] (NetworkLS)
ip:aclapeo[1515] (NetworkLS)
ip:aclapeo[12989] (NodLS)

Server Configuration Info

Name: speedy

● TCP/IP ○ NetBIOS ○ IPX

□ NetworkLS □ Central Registry LS ☑ NodLS

NetworkLS Port: 1515

CrLS Port: 10999

NodLS Port: 12989

Enter here your direct binding servers list

Figure 33. Configuration Tool Notebook - Direct Binding Section

On this page you specify all the license servers in the direct binding servers list. You must include the central registry license server that you are configuring. If the central registry license server and a network license server or nodelocked license server run on the same workstation, include all servers in the list.

As you configure the servers in the direct binding servers list, define exactly the same set of servers on each.

Specify also remote nodelocked license servers whose licenses you want to administer from this workstation. Do not include the nodelocked license server on this workstation: you can administer local nodelocked licenses automatically, without specifying direct binding.

In this example, the administrator performs the following steps for each server with which this server will communicate:

a. Select the **TCP/IP** radio button.

b. Check **NodLS**, **NetworkLS**, or **Central Registry LS**, or any combination, depending on the roles the workstation plays in the network.

c. Leave the default values in the **NetworkLS Port**, **CrLS Port**, and **NodLS Port** fields.

Note: For platforms other than Windows 95 or Windows 98, if NetBIOS is selected, the default values are, respectively, 115, 109, 129. If you change them, use values between 32 and 254.

If NetBIOS or IPX is selected, the protocol must be configured as explained under “Protocol Configuration Requirements” on page 53.

d. Click **<<Add** to add the server to the **Servers** list.

5. Select the **User** tab. The User section is displayed.

   The user name is the logon user ID in effect when you do the configuration. Enter a group name if you want to change the default.

   Note:

   The LUM Runtime group name is not the same as the Windows group name.

   LUM Runtime treats the user name as the name of the user who requests and uses licenses on this workstation. The user name is displayed as part of various reports about license usage. It is also used to reserve reservable licenses for a specific user. The group name is used to reserve reservable licenses for a specific group.

6. Select **Close** from the system menu in the upper left corner of the notebook. A message is displayed to confirm that you are ready to save your choices.
7. To run the configured processes, see “Starting and Listing Your Processes” on page 93.

The workstation can now be used as the central registry license server.

Command-Line Equivalent

To configure the central registry license server:

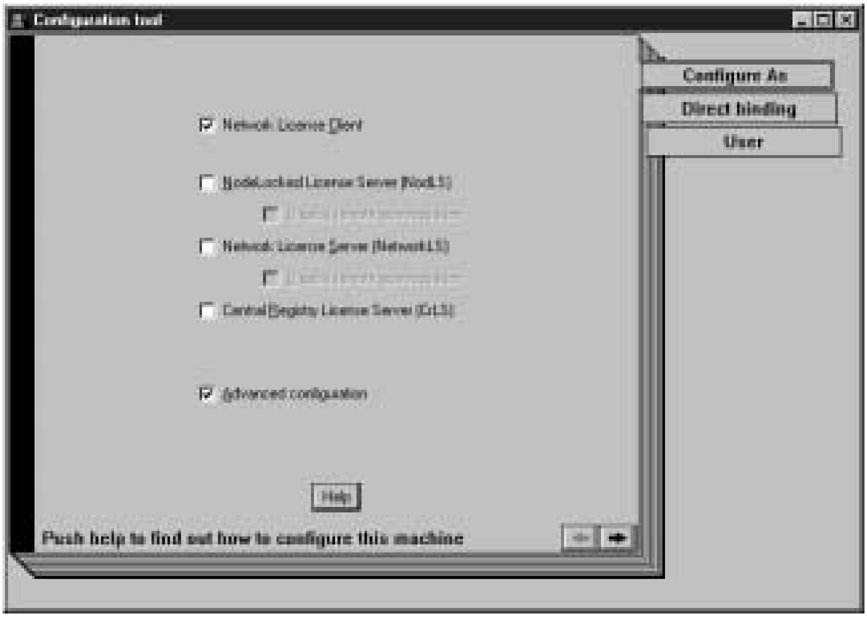
```
i4cfg -a r -S a -g Infotech -b "'network ip:lferrett ip:aclapes'
'nodelocked ip:speedy ip:aclapes' 'registry ip:pgivan'"
```

Scenario 5: Configuring a Network License Client

This scenario shows how the administrator or the end user configures LUM Runtime as a client of the network license servers configured in “Scenario 3: Configuring a Network License Server” on page 80, and of the central registry license server configured in “Scenario 4: Configuring the Central Registry License Server” on page 86. Configuring a workstation as a network license client makes it possible to use licenses of the types shown in Table 11 on page 70.

This scenario shows the steps the end user follows to configure the network license client after installation of LUM Runtime.

1. Double-click the **Configuration Tool** icon in the LUM Runtime folder. The Configuration Tool notebook is displayed, as shown in Figure 34.



☑ Network License Client

☐ NodeLocked License Server (NodLS)

☐ Network License Server (NetworkLS)

☐ Central Registry License Server (CrLS)

☑ Advanced configuration

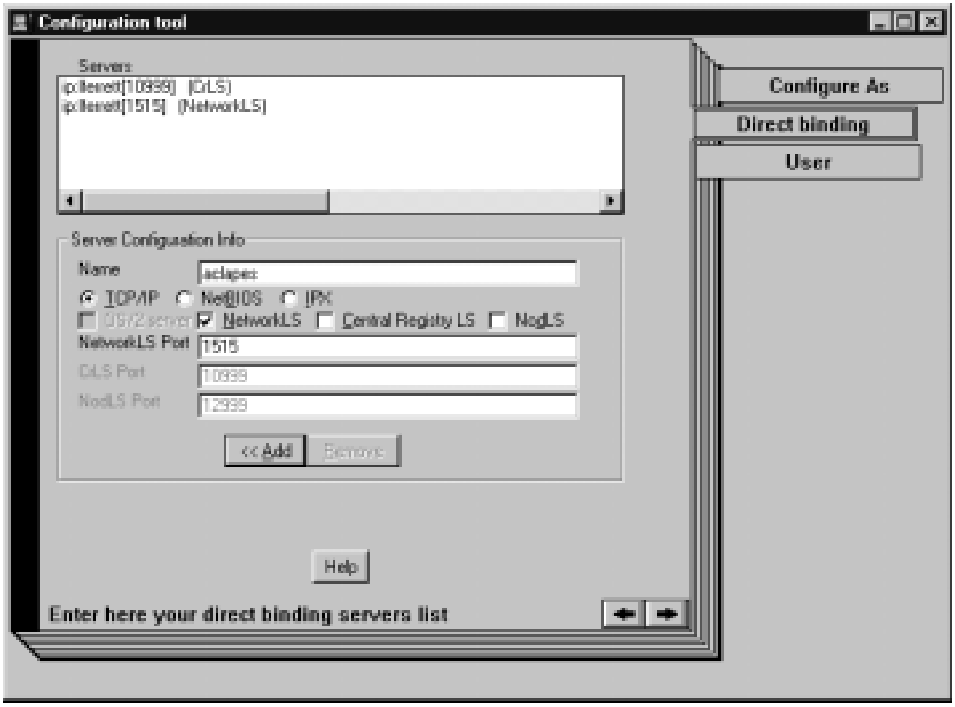
Push help to find out how to configure this machine

Figure 34. Configuration Tool Notebook - Network License Client

Check **Advanced Configuration** to add the User tab to the notebook.

2. Select the **Direct binding** tab.

   The Direct binding section is displayed, as shown in Figure 35 on page 91.



Servers:

ip:lenett[10999] (CrLS)
ip:lenett[1515] (NetworkLS)

Server Configuation Info

Name: aclapes

● TCP/IP ○ NetBIOS ○ IPX

□ OS/2 server ☑ NetworkLS □ Central Registry LS □ NodLS

NetworkLS Port: 1515

CrLS Port: 10999

NodLS Port: 12999

Figure 35. Configuration Tool Notebook - Direct Binding Section

On this page you specify all the license servers with which this client will communicate. Before you begin specifying the servers, be sure you have read the performance notes under “Planning Direct Binding” on page 39.

In this example, for each license server, the administrator does the following:

a. Select the **TCP/IP** radio button.

b. Enter the TCP/IP host name of the license server in the **Name** field. (Note that the server name is case-sensitive.)

c. Check **NodLS**, **NetworkLS**, **Central Registry LS**, or any combination, depending on the roles the server plays in the network.

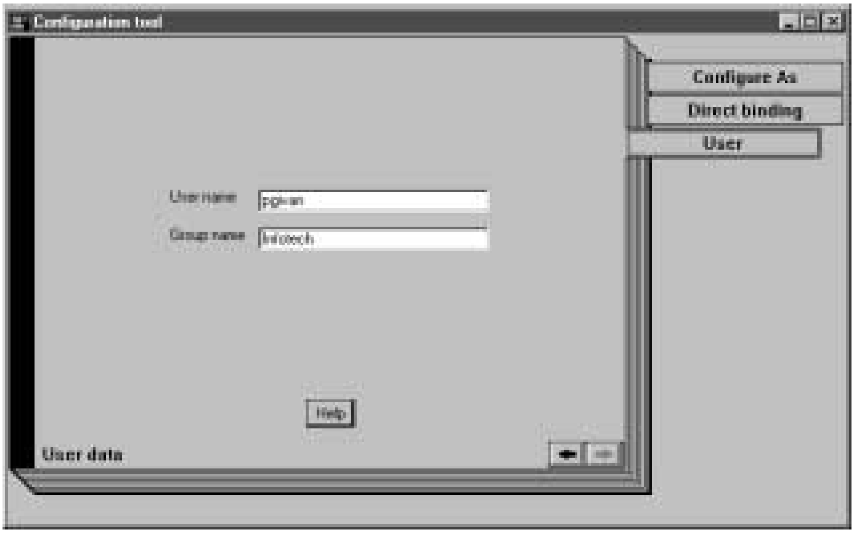
d. Leave the default values in the **NetworkLS Port**, **CrLS Port,** and **NodLS** fields.

Note: For platforms other than Windows 95 or Windows 98, if NetBIOS is selected, the default values are, respectively, 115, 109, 129. If you change them, use values between 32 and 254.

If NetBIOS or IPX is selected, the protocol must be configured as explained under REF.

e. Click **<<Add** to add the server to the Servers list.

3. Select the **User** tab. The User section is displayed, as shown in Figure 36.



User name: pgkan

Group name: infotech

Figure 36. Configuration Tool Notebook - User Section

The user name is the logon user ID in effect when you do the configuration. Enter a group name if you want to change the default.

Note:

> The LUM Runtime group name is not the same as the Windows NT group name.

LUM Runtime treats the user name as the name of the user who requests and uses licenses on this workstation. The user name is displayed as part of various reports about license usage. It is also used to reserve reservable licenses for a specific user. The group name is used to reserve reservable licenses for a specific group.

4. Select **Close** from the system menu in the upper left corner of the notebook. A message is displayed to confirm that you are ready to save your choices.

The end user can now use products that have network licenses.

Command-Line Equivalent

To configure the network license client:

```
i4cfg -a c -g Infotech
-b "'network ip:lferrett ip:aclapes' 'registry ip:lferrett'"
```

Starting and Listing Your Processes

In Windows NT and Windows 2000, you can start and list your processes in two ways: treating them as Windows services (“LUM Runtime Processes Defined as Windows Services”) or using the service manager tool (“Managing LUM Runtime with Service Manager Tool”).

LUM Runtime Processes Defined as Windows Services

During installation, your LUM Runtime processes are defined as Windows NT services. You can view them in the Settings --> Control Panel --> Services window. If you are logged on with administrator authority, you can also start and stop the services in the same window.

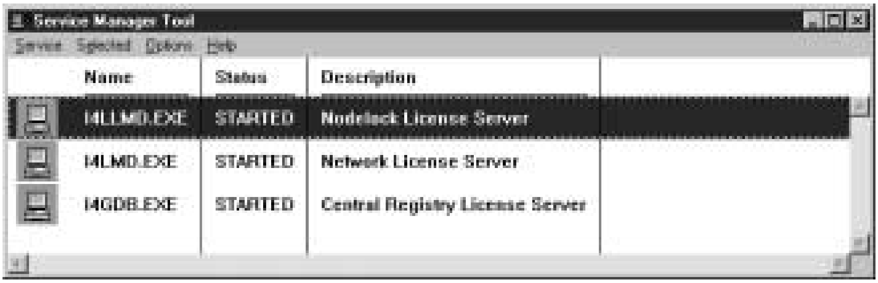
During configuration, the processes you configure are defined as manual services, which means they are not started when you start the workstation. You can change them to automatic services in the Services window, or by checking the **Start services at system startup** box on the **Start up** page of the configuration tool. Services you do not configure are defined as disabled services.

If you uninstall LUM Runtime or one or more of its features, go to the same window and disable any services you have designated as automatic. In this way you will avoid errors at system startup.

Note that errors during start or stop of the LUM Runtime processes are logged in the Windows NT log. To view them, go to the Programs --> Administrative Tools (Common) --> Event Viewer window, and select Log --> Application.

Managing LUM Runtime with Service Manager Tool

When you finish your configuration, double-click the **Service Manager Tool** icon in the License Use Runtime folder to start the processes. The Service Manager Tool window is displayed, as shown in Figure 37 on page 94.



Name	Status	Description
I4LLMD.EXE	STARTED	Nodelock License Server
I4LMD.EXE	STARTED	Network License Server
I4GDB.EXE	STARTED	Central Registry License Server

Figure 37. Service Manager Tool Window

In the Service Manager Tool window select **Service** from the menu bar and **Start** from the pull-down menu.

If you prefer to use the command-line interface, to start the processes you have configured on your workstation, issue the command:

```
i4cfg -start
```

To verify that they are up and running, issue the command:

```
i4cfg -list
```

Verifying Connections to Servers

To verify that license servers are running properly, use the `i4tv` (test verification) tool, or use the `i4blt -ln` command to get a list of active servers (network license servers, nodelocked license servers, and the central registry license server). For more information about these commands, see “Chapter 6. Command Line Interface” on page 125.

Chapter 5. Daily Tasks

This chapter contains scenarios that illustrate the daily tasks needed to manage license-enabled products.

The tasks can be done using either the basic license tool GUI (BLT GUI) or the command-line interface (CLI). The scenarios use the GUI. The equivalent CLI commands are shown at the end of each scenario.

Attention: If you want to run the scenarios, substitute your own values for the sample or fictitious names: **server name**, **product name**, **vendor name**, **enrollment certificate name**, **product version**, and **user name**.

The scenarios assume that the server **PGIVAN** is running the nodelocked license server, a network license server, and the central registry license server. The basic license tool is run from **PGIVAN**.

The scenarios use three sample products from three fictitious IBM vendors:

- SMARTJava Version 2.3, a product of the vendor IBM Software Group. SMARTJava has concurrent licenses, which the vendor delivers via a compound password. It is a customer-managed use product, and the vendor enabled it to allow the customer to exercise the hard stop/soft stop policy. Its enrollment certificate is named `smrtjava.lic`.
- DataMaster Version 2.1a, a product of the vendor IBM Corporation. DataMaster is a vendor-managed use product with reservable licenses. Its enrollment certificate is named `datamst.lic`.
- e-MailVision Version 1.2, a product of the vendor IBM Software Solutions. e-MailVision has per-server/per-seat licenses. The enrollment certificate for the per-server license is `emailvps.lic`, and for per-seat it is `emailvpt.lic`.

The administrator has placed the enrollment certificate files for these three products in the directory `c:\certifs`.

Using the Basic License Tool GUI

This section gives some general guidance for using the basic license tool GUI (BLT GUI).

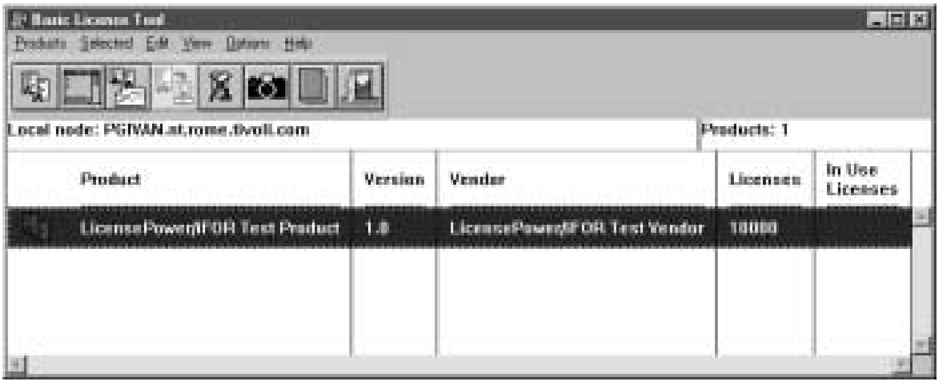
Starting the Basic License Tool GUI

To start the BLT GUI, double-click the Basic License Tool icon in the License Use Runtime folder.

Alternatively, start the BLT GUI by issuing the command:

```
i4blt
```

The main window appears (Figure 38).



Local node: PGIVAN.at.rome.tivoli.com

Products: 1

Product	Version	Vendor	Licenses	In Use Licenses
LicensePower/iFOR Test Product	1.0	LicensePower/iFOR Test Vendor	10000	

Figure 38. Basic License Tool GUI - Main Window

The products shown belong to all the active network license servers, nodelocked license servers, and the central registry.

In the Basic License Tool window the heading **Licenses** indicates licenses that are available to end users (including those that are in use) unless the current date is before the start date of the licenses or the licenses have expired.

Refreshing License Information

Before viewing information about concurrent, reservable, per-server, and concurrent nodelocked licenses in use for a product, it is a good idea to clean up stale licenses. In the main window:

1. Highlight the product.
2. Select **Clean up stale licenses** from the Selected menu.

If you turned the automatic cleanup feature on (see “Appendix A. LUM Runtime Configuration File” on page 209) all stale licenses are cleaned up automatically at each BLT refresh.

Selecting Servers, Vendors, and Licenses

To reduce the number of servers, vendors, or licenses shown in the window:

- Use the **Include...** option of the View menu. You can specify which servers and vendors are shown as of the next refresh.
- Use the **Settings...** option of the Options menu. You can specify the types of licenses (nodelocked, network, or both) to be shown as of the next refresh.

Note: If a selected server shuts down it is no longer selected, even if it is restarted.

If a selected server shuts down, and you subsequently request a report or a display of information gathered from all selected servers, the request fails and error messages notify you that it was not possible to communicate with the server. In this case, use the **Refresh Now** option of the View menu, or deselect or restart the failing server, and try again.

Performing Basic Administration

The scenarios in this section demonstrate how to:

- Enroll a licensed product ("Enrolling the Product").
- Distribute licenses from a compound password ("Distributing the Licenses" on page 100).
- Get a report on the use of licensed products ("Generating Reports" on page 101).
- Check the current users of licensed products ("Checking the Number of Product Users" on page 102).
- Reserve reservable licenses for specific users and monitor the use of reservable licenses ("Scenario 7: Managing Reservable Licenses" on page 104).

Scenario 6: Managing a Licensed Product

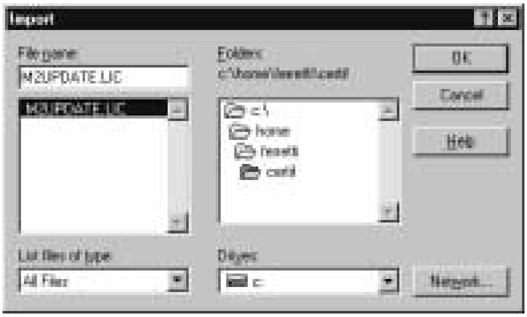
In this scenario, the administrator enrolls and manages the SMARTJava product. This scenario shows you how to:

- Enroll the SMARTJava product
- Enroll 20 licenses for SMARTJava
- Distribute five of the SMARTJava licenses to a network license server
- Request a report on usage of SMARTJava licenses during a one-month period
- Check the number of concurrent users of SMARTJava

Enrolling the Product

To enroll the SMARTJava product:

1. In the main window, select **Enroll** from the Products menu, then **Single product**.... The Enroll Product window is displayed (Figure 40 on page 98).
2. Select the **Import** button. The Import window is displayed (Figure 39 on page 98).



Import

File name: M2UPDATE.LIC

M2UPDATE.LIC

Folders: c:\home\fessett\certif

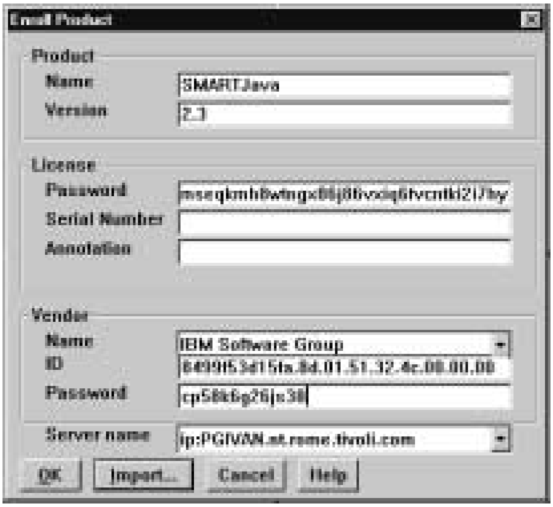
c:\
home
fessett
certif

List files of type: All Files

Drives: c:

Figure 39. Import Window for Enrolling a Product

3. On the Import window:
   a. Select the correct directory.
   b. Select the enrollment certificate. For the SMARTJava product, the file is **smrtjava.lic**.
   c. Click **OK**. The Enroll Product window is redisplayed, filled in with the information from the enrollment certificate file. See Figure 40.



Enroll Product

Product
Name: SMARTJava
Version: 2.1

License
Password: mseqkmh8wtngx86j86vxiq6fvcntk2i7hy
Serial Number:
Annotation:

Vendor
Name: IBM Software Group
ID: 849915 3d15fa.8d.01.51.32.4c.00.00.00
Password: cp58k6g26jx38

Server name: ip:PGIVAN.st.rome.tivoli.com

Figure 40. Enroll Product Window

4. In the **Server name** field, select the server on which the licenses are to be installed. In this example, because the product is customer-managed and has network licenses, the licenses are enrolled on the central registry license server **PGIVAN**.

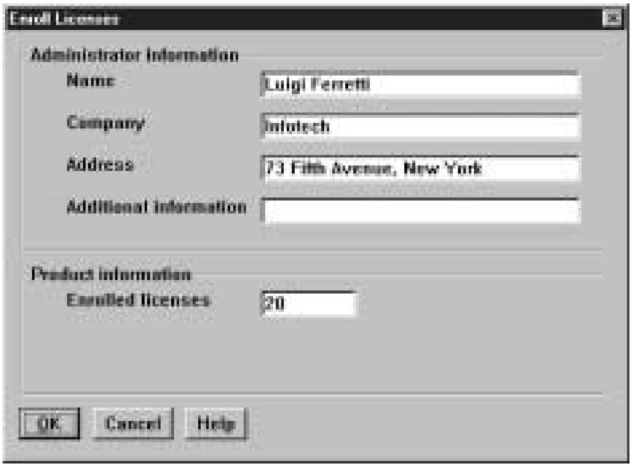
Notes:

a. If the product is vendor-managed, and has a specific target ID set in the enrollment certificate, the licenses must be enrolled on the network license server or nodelocked license server of that target workstation.
b. If the product is vendor-managed and the target ID in the enrollment certificate file is set to **ANY**, select a network license server or a nodelocked license server, depending on license type.
c. If the product is customer-managed and has nodelocked licenses, select a nodelocked license server.
d. If the password is bound to a cluster rather than an individual server, select any network license server that is an activated member of the cluster.

5. Click **OK**.

   Because the product is customer-managed, the Enroll Licenses window is displayed.

6. Fill in the Enroll Licenses window with your user information and the number of licenses you want to enroll (in this example, 20) as shown in Figure 41.



Enroll Licenses

Administrator information

Name: Luigi Ferretti

Company: Infotech

Address: 73 Fifth Avenue, New York

Additional information:

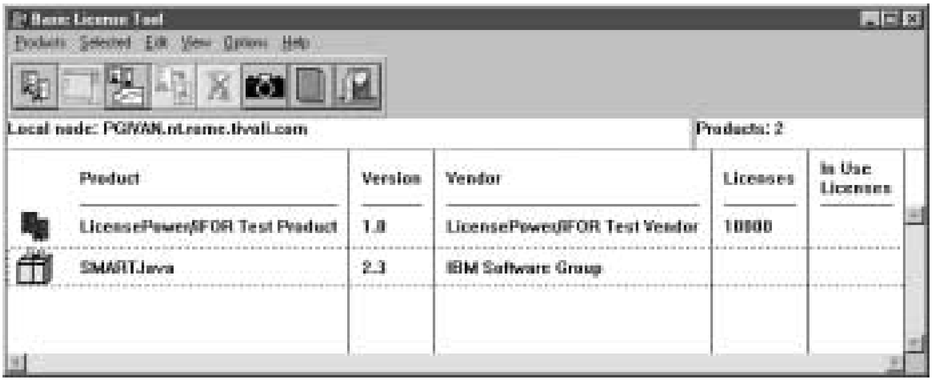
Product information

Enrolled licenses: 20

Figure 41. Enroll Licenses Window

7. Click **OK**. A message is displayed indicating that the product has been enrolled.

   The main window is displayed again, as shown in Figure 42 on page 100. There is a new line corresponding to SMARTJava, and the icon for the product (a wrapped box) indicates that the product has a compound password.



Local node: PGIVAN.nl.rome.tivoli.com

Products: 2

Product	Version	Vendor	Licenses	In Use Licenses
LicensePower/iFOR Test Product	1.0	LicensePower/iFOR Test Vendor	10000	
SMARTJava	2.3	IBM Software Group		

Figure 42. Main Window with SMARTJava Enrolled

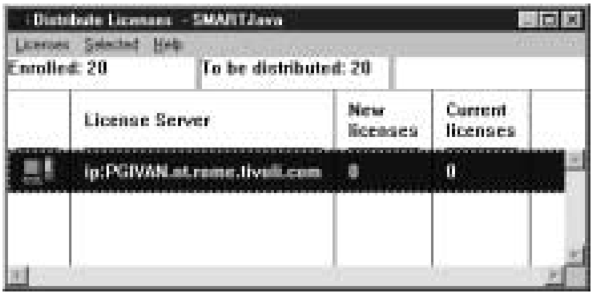
Distributing the Licenses

Before the network license clients can be granted licenses to use the product, the administrator must distribute the licenses to a network license server.

Network licenses delivered with a compound password must be distributed. Simple passwords (for example, per-server, per-seat, and concurrent nodelocked licenses) must not be distributed.

To distribute five licenses to the network license server **PGIVAN**:

1. In the main window, select the line containing the product.
2. Select the **Distribute licenses...** option from the Selected menu. The Distribute Licenses window is displayed, as shown in Figure 43.



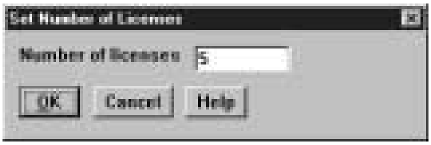
Enrolled: 20

To be distributed: 20

License Server	New licenses	Current licenses
ip:PGIVAN.nl.rome.tivoli.com	0	0

Figure 43. Distribute Licenses Window

3. In the Distribute Licenses window, select the network license server. Right click the selected server, and select the **Set number of licenses....** option from the pop-up menu. The Set number of licenses window is displayed.
4. Enter **5** in the **Number of licenses** field, as shown in Figure 44 on page 101, and click **OK**.

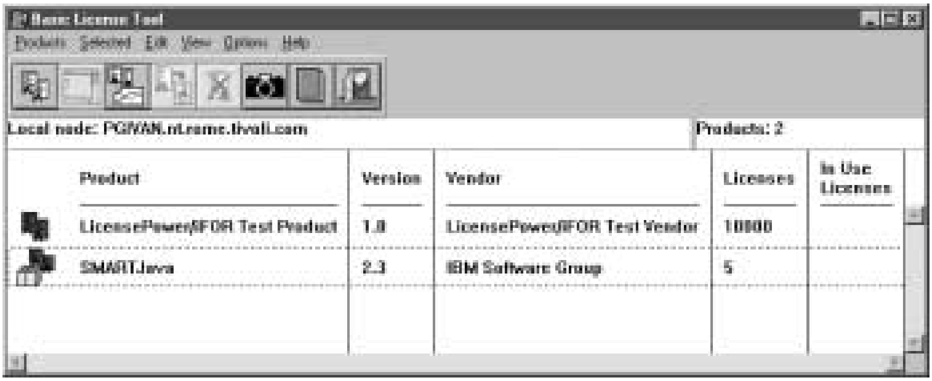


Number of licenses 5

Figure 44. Set Number of Licenses Window

The Distribute Licenses window is redisplayed. The window now shows 15 as the number of licenses available to be distributed. Select the **Distribute** option from the Licenses menu to confirm data and distribute the licenses.

5. In the main window, the number of available licenses changes to 5 and the icon for SMARTJava changes to show a compound password with distributed concurrent licenses.



Local node: PGIVAN.nl.rome.tivoli.com

Products: 2

Product	Version	Vendor	Licenses	In Use Licenses
LicensePower/iFOR Test Product	1.0	LicensePower/iFOR Test Vendor	10000	
SMARTJava	2.3	IBM Software Group	5	

Figure 45. Basic License Tool Window with Distributed Licenses

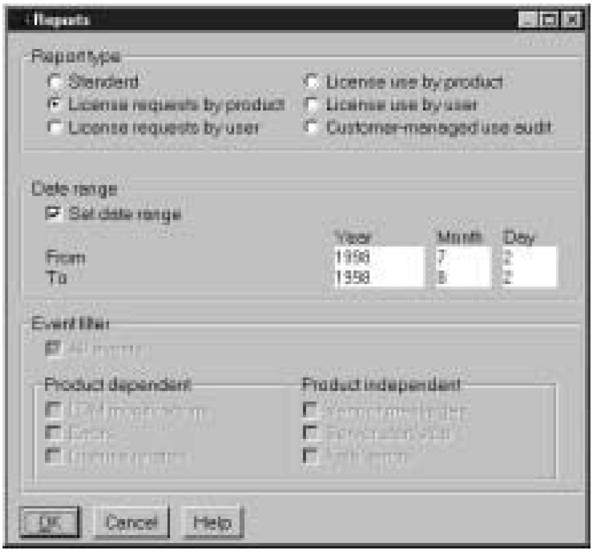
End users can now use the SMARTJava product.

Generating Reports

This section shows how to get a report of SMARTJava usage for one month: from July 2 to August 2, 1998.

1. Select SMARTJava in the main window (Figure 45).
2. Select the **Reports...** option from the Selected menu.

   The Reports window is displayed.
3. Fill in the Reports window as shown in Figure 46 on page 102.



Reports

Report type

○ Standard ○ License use by product

● License requests by product ○ License use by user

○ License requests by user ○ Customer-managed use audit

Date range

☑ Set date range

	Year	Month	Day
From	1998	7	2
To	1998	8	2

Event filter

Product dependent

Product independent

Figure 46. Reports Window

a. Select the report type **License requests by product**.

b. Check the **Set date range** check box and set the date range for the report in the **From** and **To** fields (in this example, from July 2 to August 2, 1998).

c. Click OK.

4. View the Report window, which contains the following report:

```
========================================================
L i c e n s e   R e q u e s t s   B y   P r o d u c t
=====================================================

                                Licenses    Licenses   Percent
Vendor/Product             Vrsn   Requested   Granted    Rejections
-------------------------  ----  ---------    --------   -----------
IBM Software Group         2.3      120         120         0
SMARTJava

     ==========================================
     *** End of License Requests By Product ***
     ==========================================
```

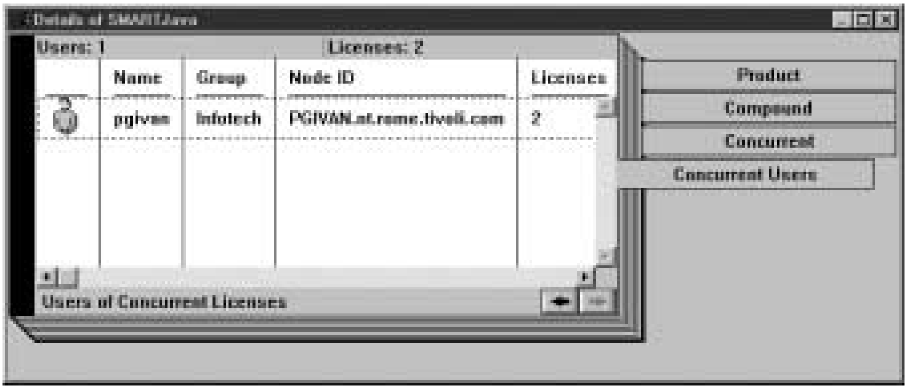
The report shows that 120 requests were made to use SMARTJava Version 2.3, and all of them were granted.

5. Click **Cancel** in the Reports window.

Checking the Number of Product Users

To check the number of concurrent users of SMARTJava:

1. Select the line corresponding to the SMARTJava product on the main window.
2. Select the option **Open as Details** from the Selected menu. The Details notebook is displayed.
3. Select the **Concurrent Users** tab. The Concurrent Users page is displayed, as shown in Figure 47.



Details of SMARTJava

Users: 1 Licenses: 2

Name	Group	Node ID	Licenses
pgivan	Infotech	PGIVAN.nt.rome.tivoli.com	2

Users of Concurrent Licenses

Figure 47. Concurrent Users Page

It shows the number of licenses in use, with the name of each user. In this example, the user **pgivan** is using two SMARTJava licenses.

4. Close the notebook.

Command-Line Equivalent

This scenario showed how to enroll SMARTJava and its 20 licenses on the central registry license server, distribute five of the licenses to network license server **PGIVAN**, get a report on one month’s usage, and check the number of concurrent users.

The equivalent CLI commands are as follows.

- To enroll the product on the central registry license server:

```
i4blt -a -n PGIVAN -f c:\certifs\smrtjava.lic -T 20
-R "'Luigi Ferretti' Infotech '73 Fifth Avenue New York'"
-I "'First installed by Luigi'"
```

  If you use the CLI to enroll a product, check the top of the enrollment certificate file. The vendor may have generated the command to be used.

- To distribute five licenses to network license server **PGIVAN**:

```
i4blt -E -n PGIVAN -v "'IBM Software Group'" -p "SMARTJava 2.3"
-A 5 -w PGIVAN -I "'Luigi'"
```

- To generate a report of requests for SMARTJava from July 2 to August 2, 1998:

```
i4blt -r2 -p "SMARTJava" -b 07/02/1998 -g 08/02/1998
```

- To display information about concurrent users of SMARTJava:

```
i4blt -s -lc -p "SMARTJava"
```

For more information about the CLI, see “Chapter 6. Command Line Interface” on page 125.

Scenario 7: Managing Reservable Licenses

In this scenario, the administrator manages licenses of the DataMaster product. This scenario shows you how to:

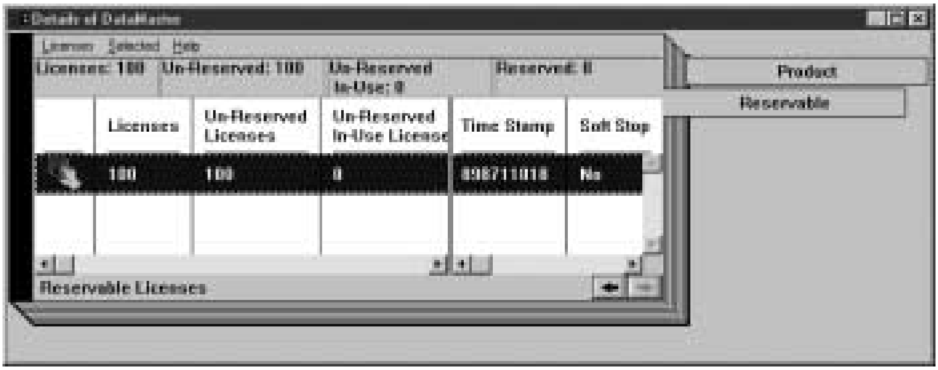
- Reserve some reservable licenses for the exclusive use of a specified user
- Monitor usage of reserved licenses by the users for which they were reserved
- Monitor use of unreserved reservable licenses by other users

In this scenario, 100 reservable licenses for DataMaster have already been enrolled. The enrollment process is the same as for concurrent licenses of a customer-managed use product, as shown in “Enrolling the Product” on page 97, except that the Enroll Licenses window is not used. After enrollment, the product appears in the main window. Note that the icon (a hand holding some licenses) indicates that the licenses are reservable.

The enrollment certificate file for DataMaster is shown as an example in “Checking License Details” on page 191.

To reserve some of the licenses for a specific user and monitor use of the reservable licenses:

1. Double-click on the product in the main window and go to the **Reservable tab** of the Details notebook, as shown in Figure 48.



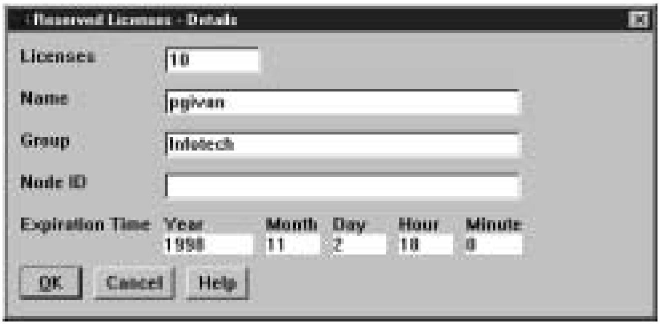
Licenses: 100 | Un-Reserved: 100 | Un-Reserved In-Use: 0 | Reserved: 0

	Licenses	Un-Reserved Licenses	Un-Reserved In-Use License	Time Stamp	Soft Stop
	100	100	0	898711018	No

Figure 48. Details Notebook with Reservable Tab. Custom configuration serial number not shown.

Note that there are 100 total licenses, with none in use and none reserved.

2. Right click on the product, and select **Reserve...** from the pop-up menu. The Reserved Licenses - Details window is displayed.
3. Fill in the number of licenses you want to reserve; the user, group, and node ID (alone or in any combination) for which you want to reserve licenses; and the date and time that the reservation is to expire, as shown in Figure 49. The latest allowed expiration date of a reservation is 12/31/2037.


Reserved Licenses - Details

Licenses: 10

Name: pgivan

Group: Infotech

Node ID:

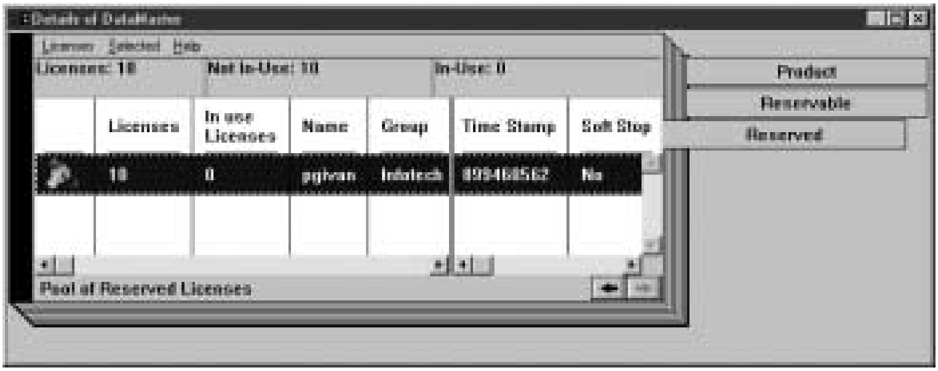
Expiration Time	Year	Month	Day	Hour	Minute
	1998	11	2	18	0

Figure 49. Reserving Reservable Licenses

In the figure, the administrator is reserving ten licenses for user **pgivan**, a member of the group **Infotech**, for use on any node. The reservation is to expire on November 2, 1998 at 18:00. Note that normally you reserve one license for a specific user and more than one license for a group or node.

4. Click **OK**.

The **Reserved** page is added to the Details notebook, as shown in Figure 50.


Details of DataMarke

Licenses: 10 Not In-Use: 10 In-Use: 0

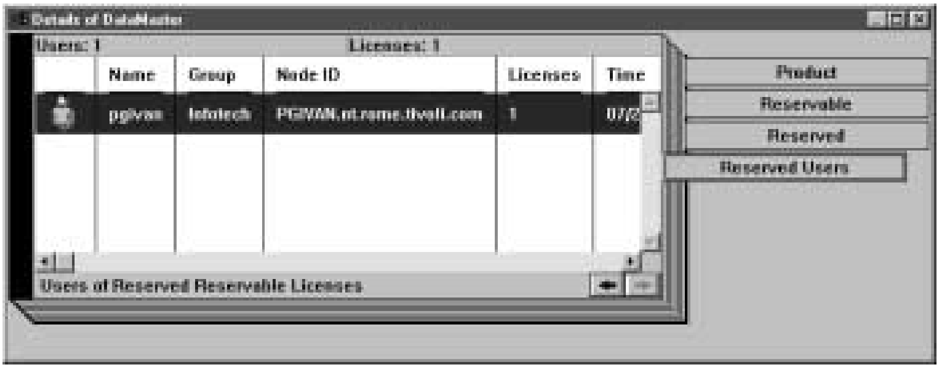
	Licenses	In use Licenses	Name	Group	Time Stamp	Soft Stop
	10	0	pgivan	Infotech	899460562	No

Pool of Reserved Licenses

Figure 50. Details Notebook with Reserved Licenses

The example page shows 10 licenses reserved for the user **pgivan** in the group **Infotech**.

Now assume the user **pgivan** requests a license. The **Reserved Users** page is added to the notebook details, as shown in Figure 51. To see the change, press **F5** to refresh the window.



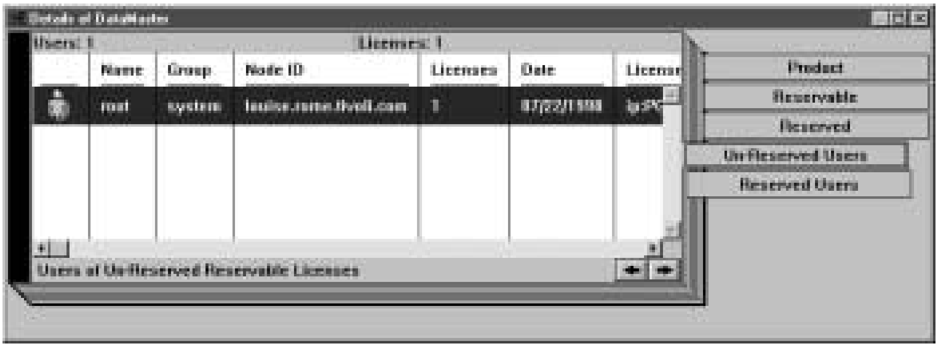
Users: 1 Licenses: 1

	Name	Group	Node ID	Licenses	Time
	pgivan	Infotech	PGIVAN.nt.rome.tivoli.com	1	07/2

Figure 51. Reserved Users Page with Reservable Licenses in Use

In this example, the **Reserved Users** page shows that the user **pgivan** is using one reserved license.

Now assume that the user **root** requests a license to DataMaster. No licenses have been reserved for **root**, but there are 90 unreserved licenses. Such licenses are available to all users until the administrator reserves them. One of those licenses is granted to **root**. It is managed exactly like a concurrent license.



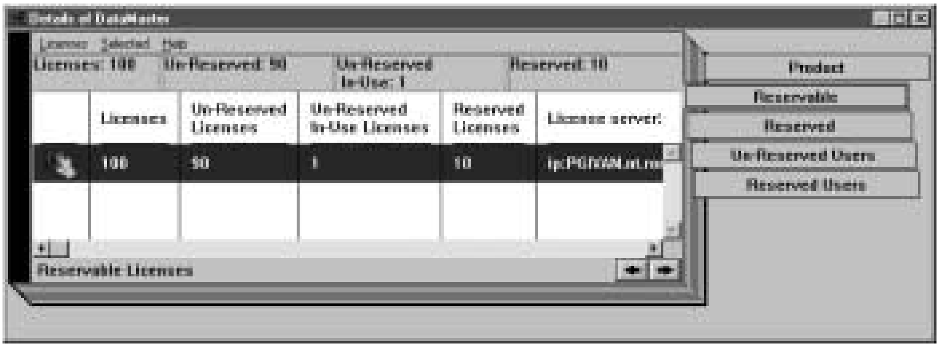
Users: 1 Licenses: 1

	Name	Group	Node ID	Licenses	Date	License
	root	system	louise.rome.tivoli.com	1	07/22/1998	lg.S

Figure 52. Un-Reserved Users Page

The **Un-Reserved Users** page is added to the Details notebook for DataMaster, as shown in Figure 52.

The Reservable page now shows 10 reserved licenses, 1 unreserved license in use, and 90 unreserved licenses, as shown in Figure 53.



Licenses: 100 Un-Reserved: 90 Un-Reserved In-Use: 1 Reserved: 10

	Licenses	Un-Reserved Licenses	Un-Reserved In-Use Licenses	Reserved Licenses	License server:
	100	90	1	10	ip:PGIVAN.at.m

Figure 53. Reservable Page with Unreserved Licenses in Use

Command-Line Equivalent

This scenario showed how to reserve ten DataMaster licenses for the user **pgivan**, display information about that user’s use of the reserved licenses, display information about the use of the unreserved licenses by other users, and display detailed information about the product.

The equivalent CLI commands are as follows.

- To get the timestamp of the licenses to be reserved:

```
i4blt -lp -i -v "'IBM Corporation'" -p "DataMaster"
```

- To reserve ten DataMaster licenses for the user **pgivan** in group **Infotech** using the license password identified by timestamp 899460562:

```
i4blt -R r -v "'IBM Corporation'" -p "DataMaster 2.1a"
-t 899460562 -A 10 -g 11/02/1998 -H 18:00 -u "pgivan Infotech *"
```

- To display information about the users of reserved licenses:

```
i4blt -s -lrr -v "'IBM Corporation'" -p "DataMaster"
```

- To display information about the users of unreserved licenses:

```
i4blt -s -lru -v "'IBM Corporation'" -p "DataMaster"
```

- To display detailed information about the product, including the number of reserved and unreserved licenses:

```
i4blt -lp -i -v "'IBM Corporation'" -p "DataMaster"
```

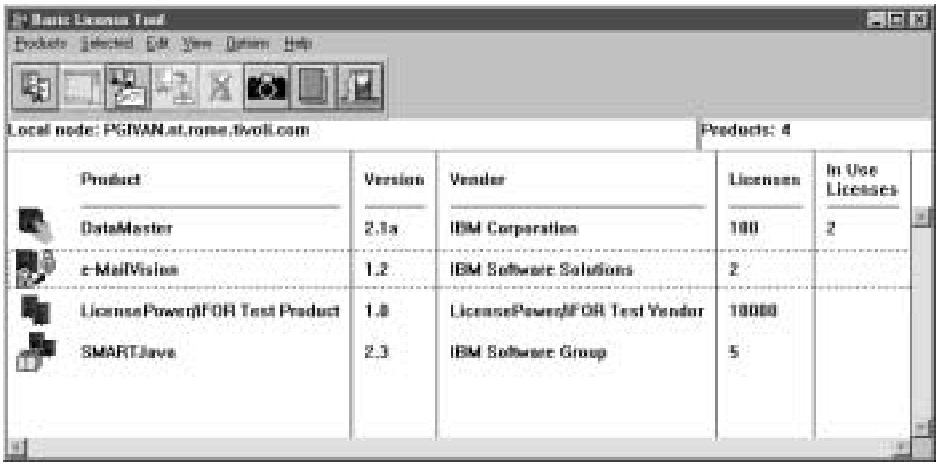
Exercising Customer-Controlled Policies

The scenarios in this section explain how to exercise the customer-controlled policies outlined in “Customer-Managed Use Products” on page 6. The scenarios show how to:

- Switch from per-server to per-seat licenses (“Scenario 8: Switching from Per-Server to Per-Seat Licenses”)
- Use the hard stop / soft stop policy (“Scenario 9: Using the Hard Stop / Soft Stop Policy” on page 109)
- Update the number of licenses of a customer-managed use product (“Scenario 9: Using the Hard Stop / Soft Stop Policy” on page 109)
- Control the set of users who are permitted to use a specific application (“Scenario 10: Restricting User Access” on page 112)

Scenario 8: Switching from Per-Server to Per-Seat Licenses

In this scenario, the administrator switches the policy for the product e-MailVision from per-server to per-seat. The per-server license has already been enrolled, as shown in Figure 54. The enrollment process is the same as for customer-managed concurrent licenses, as described in “Enrolling the Product” on page 97.



Local node: PGIVAN.at.rome.tivoli.com

Products: 4

Product	Version	Vendor	Licenses	In Use Licenses
DataMaster	2.1a	IBM Corporation	100	2
e-MailVision	1.2	IBM Software Solutions	2	
LicensePower/iFOR Test Product	1.0	LicensePower/iFOR Test Vendor	10000	
SMARTJava	2.3	IBM Software Group	5	

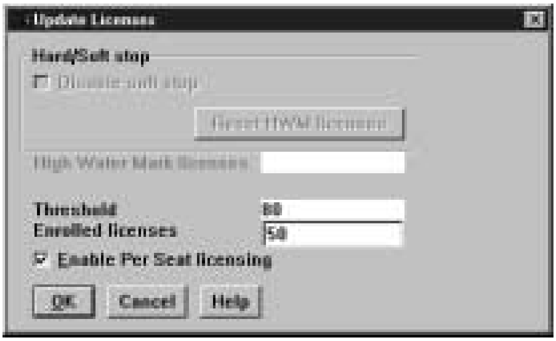
Figure 54. Basic License Tool Window with Per-Server Licenses Enrolled

The icon for e-MailVision shows per-server licenses.

To make the switch:

1. Install the per-seat license, importing the per-seat enrollment certificate **emailvpt.lic** and enrolling 50 licenses. The enrollment process is the same as for customer-managed concurrent licenses, as described in “Enrolling the Product” on page 97.
2. Select **e-MailVision** in the main window (Figure 54). Select the **Update licenses...** option from the **Selected** menu. The Details notebook is displayed.

3. Click the **Per-Seat** tab. Now right-click the product, and select **Update licenses...** from the pop-up menu. The Update Licenses window is displayed. Check the box **Enable per-seat licensing**, as shown in Figure 55.



□ Disable soft stop

High Water Mark licenses

Threshold 80

Enrolled licenses 50

☑ Enable Per Seat licensing

Figure 55. Enabling Per-Seat Licensing

4. Click **OK**.

   In the main window the icon changes to show per-seat licenses. The license server will remove the old per-server licenses.

Command-Line Equivalent

This scenario showed how to switch the e-MailVision product from per-server licensing to per-seat licensing.

The equivalent CLI commands are as follows.

- To enroll the per-seat licenses for e-MailVision:

  ```
  i4blt -a -f c:\certifs\emailvpt.lic -T 2 -R "'Luigi Ferretti'"
  ```

- To switch e-MailVision from per-server to per-seat licensing:

  ```
  i4blt -U -v "'IBM Software Solutions'" -p "e-MailVision 1.2" -S yes
  ```

Scenario 9: Using the Hard Stop/Soft Stop Policy

In “Scenario 6: Managing a Licensed Product” on page 97, the administrator enrolled the SMARTJava product and distributed five licenses from a network compound password. Now the administrator has distributed the remaining 15 enrolled licenses. Because the vendor enabled this product to use the hard stop/soft stop policy, and the administrator is running it with soft stop set, it is possible that more than 20 licenses are being used at any given time.

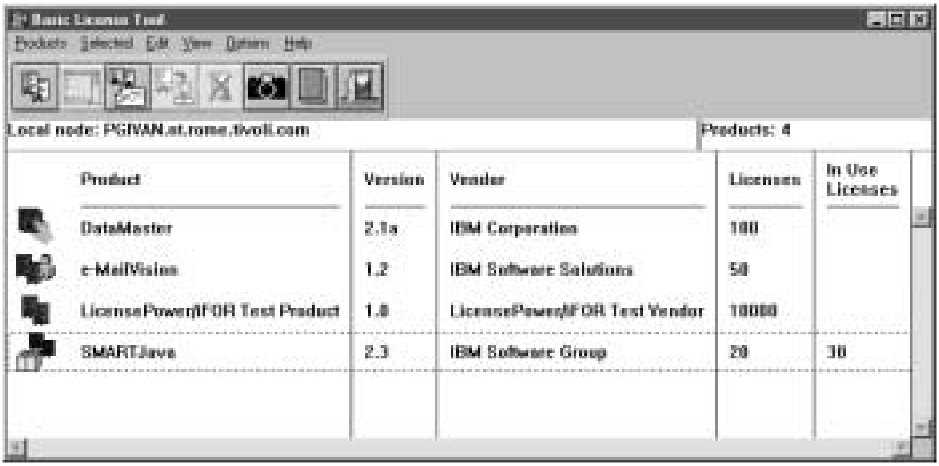
In this scenario, the administrator:

1. Checks the current number of licenses in use and the maximum number of licenses that have been granted beyond the 20 enrolled (the high-water mark, which in this example is 10)

2. Decides to acquire ten more licenses
3. Updates the number of enrolled licenses to 30
4. Resets the high-water mark to 0
5. Distributes ten more licenses from the compound password

Follow these steps:

1. Check the current license usage in the main window (Figure 56).



Local node: PGIVAN.at.rome.tivoli.com

Products: 4

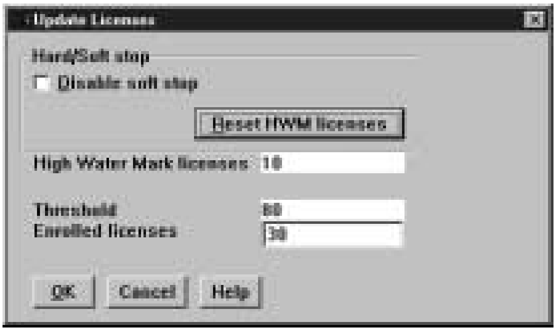
Product	Version	Vendor	Licenses	In Use Licenses
DataMaster	2.1a	IBM Corporation	100	
e-MailVision	1.2	IBM Software Solutions	50	
LicensePower/iFOR Test Product	1.0	LicensePower/iFOR Test Vendor	10000	
SMARTJava	2.3	IBM Software Group	20	30

Figure 56. Main window with Soft-Stop Licenses in Use

   The window shows that ten licenses beyond the 20 enrolled are in use.

2. Check the longer-term license usage:
   a. Select **SMARTJava** on the main window.
   b. Select **Update licenses...** from the **Selected** menu.

      The Details notebook is displayed.
3. Go to the Concurrent page, click with the right mouse button on an entry for the product, and select **Update licenses...** from the pop-up menu. The Update Licenses window is displayed (Figure 57 on page 111).



Hard/Soft stop

□ Disable soft stop

High Water Mark licenses: 10

Threshold: 80

Enrolled licenses: 30

Figure 57. Update Licenses Window with High-Water Mark

The **High Water Mark licenses** field shows that a maximum of 10 soft-stop licenses were in use at any time since the high-water mark was last reset.

4. Decide to acquire ten more licenses, and pay the vendor for them.
5. Returning to the Update Licenses window, enter **30** in the **Enrolled licenses** field. At the same time, reset the high-water mark to **0** by clicking on **Reset HWM Licenses**.
6. Click **OK**. A confirmation message is displayed.
7. Distribute the ten newly enrolled licenses (see “Distributing the Licenses” on page 100). In the main window (Figure 38 on page 96), the Licenses column for SMARTJava is updated to 30.

Command-Line Equivalent

This scenario showed how to check usage of soft stop licenses for SMARTJava, enrolled and distributed an additional ten licenses, and reset the high-water mark to **0**.

The equivalent CLI commands are as follows.

- To display information about usage of soft stop licenses of SMARTJava:

  ```
  i4blt -lp -p "SMARTJava" -i
  ```

- To update the enrollment, enrolling ten more licences on the central registry license server:

  ```
  i4blt -U -v "'IBM Software Group'" -p "SMARTJava 2.3" -T 30 -I "'Luigi Ferretti'"
  ```

- To reset the high-water mark to 0:

  ```
  i4blt -U -v "'IBM Software Group'" -p "SMARTJava 2.3" -M
  ```

- To distribute the ten licenses to network license server **PGIVAN**:

  ```
  i4blt -E -n PGIVAN -v "'IBM Software Group'" -p "SMARTJava 2.3"
  -A 10 -w PGIVAN -I "'Luigi Ferretti'"
  ```

Scenario 10: Restricting User Access

This scenario explains how to create a user file to designate that certain users may or may not use certain products. You could also use a previously created user file as a base. To create a user file, follow these steps:

1. Using a text editor, open a file named `userfile`.
2. Within the file, to restrict access to a product, use the **vendor** keyword, followed by the name of the vendor, followed by either **all** (meaning all products of this vendor) or the name of a product. Enclose vendor names and product names in quotation marks if they contain embedded blanks. For example:

   ```
   vendor "IBM Software Group" SMARTJava
   vendor Grafix,Inc. all
   ```

   You need one `vendor` statement for each product of the same vendor, unless `all` is sufficient for your purposes.
3. After each vendor statement, code either an `allow` or a `disallow` statement. The `allow` and `disallow` statements are mutually exclusive.

   allow Specifies that the user names that follow this keyword are allowed to use the product. If no user names follow this keyword, no users can use the product. The user name is the name specified in the configuration file with the tag UserName. For example:

   ```
   vendor "IBM Software Group" SMARTJava
   allow fritz harry monique penny
   ```

   This specifies that only four users can use the SMARTJava product: Fritz, Harry, Monique, and Penny.

   disallow Specifies that the user names that follow this keyword are not allowed to use the product. If no user names follow this keyword, all users can use the product. The user name is the name specified in the configuration file with the tag UserName. For example:

   ```
   vendor Grafix,Inc. all
   disallow heather jason
   ```

   This specifies that all users except Heather and Jason can use all Grafix, Inc. software products.
4. Store the file in the `ifor\ls\conf` directory of the workstation where the licenses to be restricted are installed. In this example, the complete user file is:

   ```
   % This line is a comment
   % ******
   vendor "IBM Software Group" SMARTJava
   ```

```
allow fritz harry monique penny
% ******
vendor Grafix,Inc. all
disallow heather jason
```

5. For a consistent user authorization policy, store the same use file on all network license servers and nodelocked license servers in your environment, including the central registry license server.
6. When adding a new product, remember to update user files at all the license servers accordingly.

Managing High-Availability Licensing (HAL)

The scenario in this section shows how to set up and manage a cluster of network license servers to ensure high availability of concurrent licenses.

When you create a cluster, LUM Runtime generates the cluster ID. For a software vendor to create passwords that are bound to a cluster rather than to a single server, you must provide the cluster ID to the vendor. Therefore, you must create the cluster before you can request licenses bound to the cluster from a software vendor.

Scenario 11: Creating and Managing a Cluster

In this scenario, the administrator:

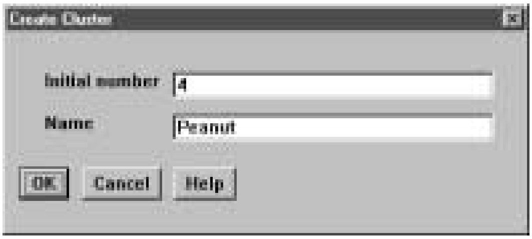
- Creates a cluster consisting of four network license servers (members)
- Activates all the servers in the cluster
- Adds a fifth server to the cluster
- Deactivates a cluster member

Creating a Cluster

To create a cluster:

1. Before you begin, make sure all the servers you intend to put in the cluster are configured to communicate with one another through direct binding.
2. In the main window (Figure 38 on page 96), select **View** from the menu bar and the **Clusters...** option from the menu. The Clusters window is displayed, as shown in Figure 60 on page 115.
3. Select **Clusters** from the menu bar and **Create...** from the menu. The Create Cluster window is displayed.
4. Use the **Initial Number** box to specify how many servers you intend to put in the cluster. Make sure that this number is what you want, because it determines the maximum number of members the cluster can ever have and the minimum number of activated members necessary to run the cluster.

In this example, as shown in Figure 58, the administrator specifies that the initial number of members is 4.



Initial number: 4

Name: Peanut

Figure 58. Create Cluster Window

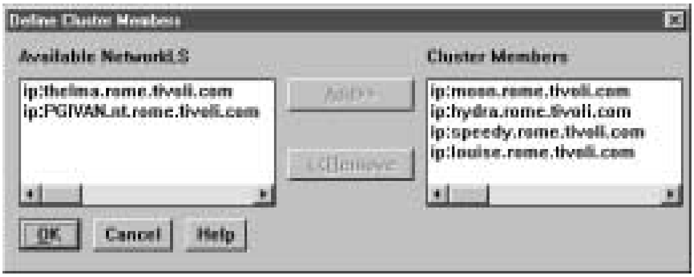
Use the **Name** field to give the cluster a unique name. You will use the name to specify the cluster in GUI and command-line operations. In this example, the administrator names the cluster **Peanut**.

5. Click **OK**. A pop-up message gives you an opportunity to change the initial number of members.
6. After you confirm your choices, the Define Cluster Members window is displayed.

   Potential members of the cluster are shown in the **Available NetworkLS** box. These are network license servers that are up and running, and are not activated in any other cluster.

   Although Windows 95 and Windows 98 servers may be listed, do not select them. They can not be members of a cluster.

   Select a server from the **Available NetworkLS** list and use the **Add>>** push button to move it to the **Cluster Members** list. Continue until the **Cluster Members** list has exactly the number of servers you specified in **Initial Number**. You can move servers between the **Available NetworkLS** and **Cluster Members** boxes until your selections are final. In the example shown in Figure 59, the administrator selects servers **moon**, **hydra**, **speedy**, and **louise**.



Available NetworkLS

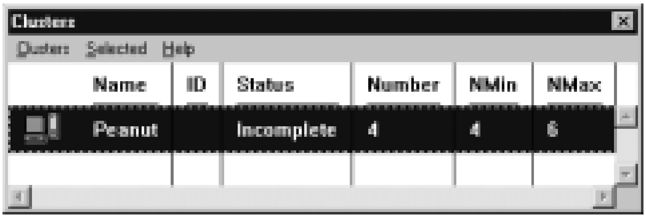
ip:thelma.rome.tivoli.com
ip:PGIVAN.nt.rome.tivoli.com

Cluster Members

ip:moon.rome.tivoli.com
ip:hydra.rome.tivoli.com
ip:speedy.rome.tivoli.com
ip:louise.rome.tivoli.com

Figure 59. Define Cluster Members Window

7. Click **OK**. A pop-up window is displayed to give you an opportunity to change your selections. The Clusters window is redisplayed with an entry for the newly-defined cluster, as shown in Figure 60.



Name	ID	Status	Number	NMin	NMax
Peanut		Incomplete	4	4	6

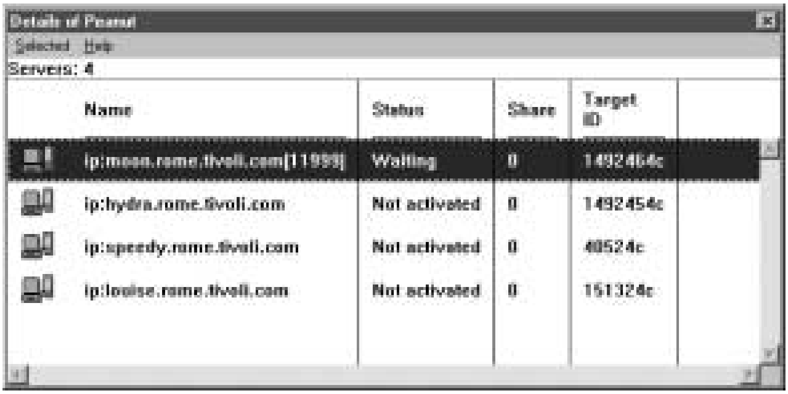
Figure 60. Clusters Window with New Cluster Added

Note that the status of the cluster is **Incomplete**, because not enough servers have yet been activated.

Activating Cluster Members

For a cluster to be able to serve licenses, a minimum number of servers must be activated, and more than half of the activated servers in the cluster must be available. When you create a cluster, only the first server (in this example, **moon**) is activated. To activate the other servers, follow these steps:

1. In the Clusters window, highlight the cluster **Peanut**. Select **Selected** from the menu bar and **Open As Details...** from the menu. The Details of Peanut window is displayed, as shown in Figure 61.



Servers: 4

Name	Status	Share	Target ID
ip:moon.rome.tivoli.com[11999]	Waiting	0	1492464c
ip:hydra.rome.tivoli.com	Not activated	0	1492454c
ip:speedy.rome.tivoli.com	Not activated	0	40524c
ip:louise.rome.tivoli.com	Not activated	0	151324c

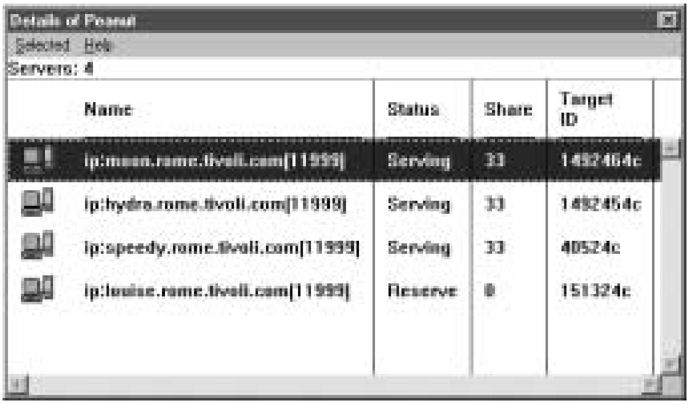
Figure 61. Details of New Cluster

The status of server **moon** is **Waiting**, because it has been activated but the cluster does not yet have enough activated members to start serving licenses. Next to each activated server is displayed the number of the port on which the server performs its high-availability licensing activities.

2. Select server **hydra**. Click on it with the right mouse button and select **Activate** to activate the server. Similarly activate servers **speedy** and **louise**.

   While each activation is being processed and all the servers in the cluster are being updated, the cluster goes into **Change Pending** status for a time that depends on the number of activated servers in the cluster. While the cluster is in this status, you cannot perform any administration on it. Return to the main window and use F5 to refresh the display, and then view the Clusters window again until the cluster has exited from **Change Pending** status.

3. Return to the Details of Peanut window, where:
   - The status of servers **moon**, **hydra**, and **speedy** has changed to **Serving**.
   - Each server is serving 33% of the licenses, as shown in Figure 62.
   - Server **louise** is in **Reserve** status, ready to take over if **moon**, **hydra**, or **speedy** goes down.



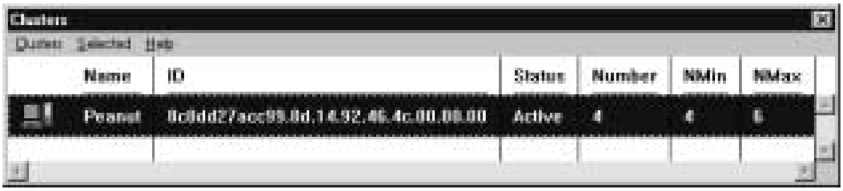
Servers: 4

Name	Status	Share	Target ID
ip:moon.rome.tivoli.com[11999]	Serving	33	1492464c
ip:hydra.rome.tivoli.com[11999]	Serving	33	1492454c
ip:speedy.rome.tivoli.com[11999]	Serving	33	40524c
ip:louise.rome.tivoli.com[11999]	Reserve	0	151324c

Figure 62. Details of Cluster after Three Activations

4. Return to the Clusters window (Figure 63 on page 117). The status of the cluster has changed to **Active**. The cluster ID has been generated and is displayed.

   The administrator can now give the cluster ID to software vendors who will create passwords bound to the cluster.



	Name	ID	Status	Number	NMin	NMax
	Peanut	8c8dd27acc99.0d.14.92.46.4c.00.00.00	Active	4	4	5

Figure 63. Clusters Window with Cluster ID

5. Return to the main window, where the HAL test product has been enrolled on the cluster. You can use it to test operation of the cluster.

Adding a Cluster Member

If a cluster has fewer than its maximum number of members, you can add members, one at a time, up to the maximum number. In this example, you can add one additional member to **Peanut**, as follows:

1. In the Clusters window, highlight the **Peanut** cluster. From the **Selected** menu, select **Add cluster member...**. The Add Cluster Members window is displayed. The window is similar to the Define Clusters window shown in Figure 59 on page 114.

   Potential members of the cluster are shown in the **Available NetworkLS** box. These are network license servers that are not activated in any other cluster. Although Windows 95 and Windows 98 servers may be listed, do not select them. They can not be members of a cluster.

   Select a server (in this example, **thelma**) from the **Available NetworkLS** list and click **Add>>** to move it to the **Cluster Members** list. You can move servers between the **Available NetworkLS** and **Cluster Members** boxes until your selection is final.

2. Click **OK**. A pop-up window is displayed to give you an opportunity to change your selection. The Clusters window is redisplayed. The number of servers in **Peanut** is now **5**, and the status of the cluster is **Active**. The newly added server is automatically activated.

3. Open the **Details of Peanut** window (Figure 61 on page 115). The status of servers **moon**, **hydra**, and **speedy** is **Serving**, and each is serving one-third of the licenses. The status of **louise** and **thelma** is **Reserve**.

Deactivating a Server

If you want a server to stop functioning as part of a cluster, you can deactivate it. You can then activate the server in another cluster. You cannot, however, substitute another server for the deactivated server in the cluster where it is deactivated.

To deactivate a server in a cluster:

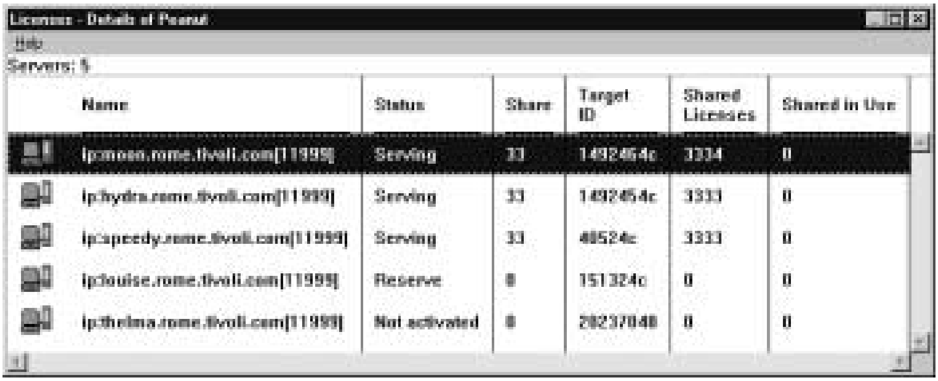
1. In the Details of Peanut window, select server **thelma**. Click on it with the right mouse button and select **Deactivate** to deactivate the server. While

the deactivation is being processed and all the servers in the cluster are being updated, the cluster goes into **Change Pending** status for a time that depends on the number of activated servers in the cluster. While the cluster is in this status, you cannot perform any administration on it. Return to the main window and use F5 to refresh the display, and then view the Clusters window again until the cluster has exited from **Change Pending** status.

2. Return to the Details of Peanut window. The status of server **thelma** has changed to **Not Activated**.

Viewing Licenses Being Served

For a view of a cluster by product, go to a high-availability product (in this case, the HAL Test Product) in the main window. Open the product’s Details notebook. On the **Concurrent** page, right-click the product and select the **Show Servers** option. The Details of Cluster window is displayed with additional information about the number of licenses being served by each server, as shown in Figure 64.



Servers: 5

Name	Status	Share	Target ID	Shared Licenses	Shared in Use
ip:moon.rome.tivoli.com[11999]	Serving	33	1492464c	3334	0
ip:hydra.rome.tivoli.com[11999]	Serving	33	1492454c	3333	0
ip:speedy.rome.tivoli.com[11999]	Serving	33	48524c	3333	0
ip:louise.rome.tivoli.com[11999]	Reserve	0	151324c	0	0
ip:thelma.rome.tivoli.com[11999]	Not activated	0	28237848	0	0

Figure 64. Details of Servers Serving HAL Test Product

Enrolling and Removing Licenses on a Cluster

In the Clusters window, you can see the cluster ID of any cluster. Give the cluster ID to the software vendor, who can use it to create passwords that are bound to the cluster, rather than to an individual server. Enroll the licenses on any server of the cluster, as shown in “Enrolling the Product” on page 97.

If you later decide to remove licenses enrolled on a cluster, all activated servers in the cluster must be up and running at the time you do the removal. You must issue the command:

```
i4blt -d -n server_name
```

where *server_name* identifies one of the servers in the cluster on which the license is enrolled.

Command-Line Equivalent

- To create a cluster named **Peanut** that has members **moon**, **hydra**, **speedy**, and **louise**:

  ```
  i4blt -H c -N Peanut -T 4 -n "moon hydra speedy louise"
  ```

- To activate the servers **hydra**, **speedy**, and **louise**:

  ```
  i4blt -H a -N Peanut -n hydra
  i4blt -H a -N Peanut -n speedy
  i4blt -H a -N Peanut -n louise
  ```

  The server **moon**, which is the first in the list, is automatically activated.

- To add the server **thelma** to the cluster:

  ```
  i4blt -H a -N Peanut -n thelma
  ```

- To deactivate the server **thelma**:

  ```
  i4blt -H d -N Peanut -n thelma
  ```

- To get an overall report of cluster status:

  ```
  i4blt -H s -N Peanut
  ```

- To get a report of cluster status from the perspective of one of the activated servers in the cluster:

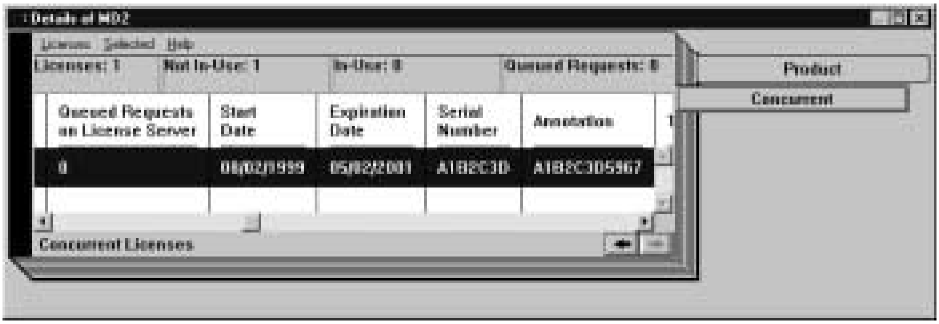
  ```
  i4blt -H s -n hydra
  ```

Scenario 12: Upgrading a Custom Configuration

The scenario in this section shows you how to upgrade a custom configuration by adding a product to a current custom configuration.

To ensure that the products used are up to date, functionally suitable, and competitive, you will occasionally need to add new product features, increase the number of licenses, or extend the license period. To do this, you request from the vendor a new custom configuration password and supply the serial number of the current license. This serial number identifies your current custom configuration. You then pay for only the difference between the cost of the current license and that of the new license. Next, you install the upgraded license as shown in the following procedure.

Before you start the procedure, look at the current entries on the **Concurrent** page of the Details notebook. Double-click the product name in the Basic License Tool window, then select the **Concurrent** tab of the Details notebook. The Details page looks similar to that shown in Figure 65 on page 120.



Licenses: 1 | Not In-Use: 1 | In-Use: 0 | Queued Requests: 0

Queued Requests on License Server	Start Date	Expiration Date	Serial Number	Annotation
0	06/02/1999	05/02/2001	A1B2C3D	A1B2C3D5967

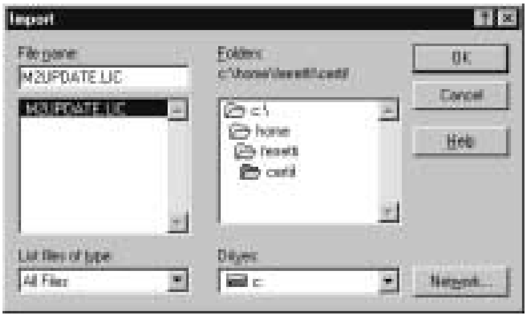
Figure 65. Initial State of the Concurrent Page for a Custom Configuration License

To upgrade your current license:

1. In the main window, select **Enroll** from the Products menu, and then **Single product...**. The Enroll Product window is displayed.
2. Select **Import**.

   The Import window is displayed.

   Select the correct directory, and then select the custom configuration enrollment certificate (in this example, **m2update.lic**), as shown in Figure 66.



File name: M2UPDATE.LIC

Folders: c:\home\hewitt\certif

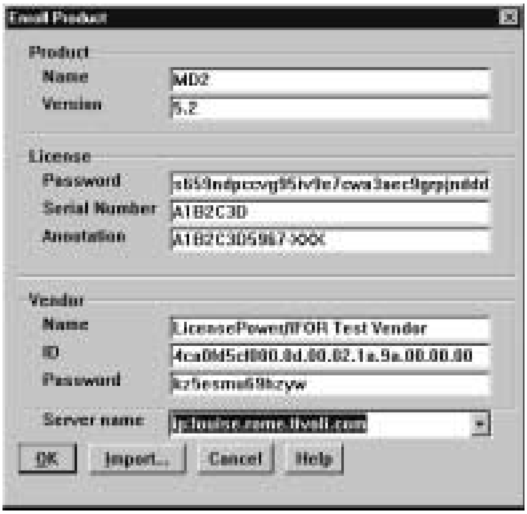
List files of type: All Files

Drives: c:

Figure 66. Import Window for Custom Configuration

3. Click **OK**.

   The Enroll Product window is redisplayed, filled in with information from the enrollment certificate file, as shown in Figure 67 on page 121.



Product
Name: MD2
Version: 5.2

License
Password: s659ndpccvg95iv9e7cwa3aec9grpjnddd
Serial Number: A1B2C3D
Annotation: A1B2C3D5967-XXX

Vendor
Name: LicensePowerIFOR Test Vendor
ID: 4ca0bd5cd000.0d.00.02.1a.9a.00.00.00
Password: kz5esmu69kzyw

Server name: ip:louise.rome.tivoli.com

Figure 67. Enroll Product Window for Custom Configuration

4. In the **Server name** field, select the server on which the initial key is installed. In this example, because the product is vendor-managed and has network licenses, the custom configuration license is enrolled on the network license server that has the target ID for which the license was created.

   Notes:

   a. If a specific target ID is set in the enrollment certificate, the licenses must be enrolled on the network license server of that target workstation.
   b. If the target ID in the enrollment certificate file is set to **ANY**, select a network license server.

   The license serial number is the same as it was for the previously enrolled license.

5. Click **OK**.

   The main window is displayed. The **Concurrent** page of the Details notebook now contains the updated license information. To display this page, double-click on the product name, then select the **Concurrent** tab of the Details notebook. Scroll to the right to see the serial number and the annotation.

Command-Line Equivalent

To upgrade a custom configuration license, using the license certificate file **m2update.lic**, on server **louise**:

```
i4blt -a -f m2update.lic -n louise
```

where:

m2update.lic The name of the file that contains the upgraded license certificate

louise The name of the server on which the initial key is installed

Scenario 13: Managing Bundles

In this scenario, the administrator manages licenses belonging to a bundle. This scenario shows you how to perform the following tasks:

- Enroll multiple certificates belonging to a bundle.
- Monitor bundles and licenses within them.
- Manage groups of licenses belonging to a bundle.
- Delete a bundle.

Enrolling Multiple Certificates

To enroll multiple certificates, the administrator performs the following steps:

1. Select **Enroll** from the menu, and then **Multiple certificates...** The Import window is displayed; see Figure 68.

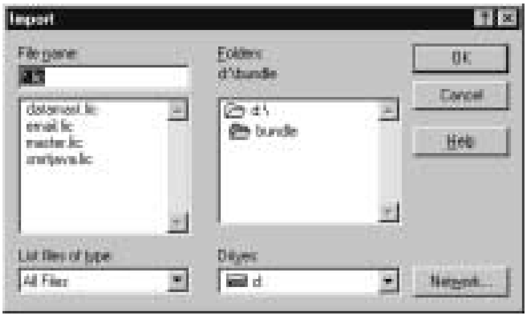
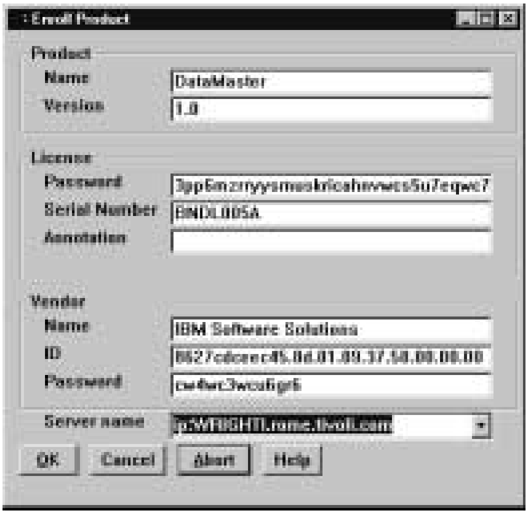


Figure 68. Import Window for Enrolling a Bundle

2. From the **File Name** list, select the enrollment certificates `smrtjava.lic`, `email.lic`, and `datamst.lic`, and the master license `master.lic`.
3. Click **OK**. The Enroll Product window is displayed for each certificate, filled in with information from the enrollment certificate file; see Figure 69 on page 123.



Product
Name: DataMaster
Version: 1.0

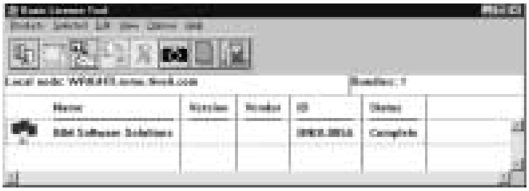
License
Password: 3pp6mznyysmuskricahwwcs5u7eqwc7
Serial Number: BNDL005A
Annotation:

Vendor
Name: IBM Software Solutions
ID: 8627cdcee c45.0d.01.09.37.58.00.00.00
Password: cw4wc3wcu6gr6

Server name: WRIGHT1.rome.tivoli.com

Figure 69. Enroll Product Window

4. For each certificate, the administrator performs the following steps:
   a. In the **Server name** field, select the server on which licenses are to be installed.
   b. Click **OK**. The Enroll Product window is redisplayed, filled in with information about the next selected certificate.
5. Repeat step 4 for all certificates. When you have enrolled all the certificates, the Basic License Tool window is displayed; see Figure 70.



Local node: WRIGHT1.rome.tivoli.com
Bundles: 1

Name	Version	Vendor	ID	Status
IBM Software Solutions			BNDL005A	Complete

Figure 70. Bundle Enrolled

Monitoring Bundles and their Licenses

When all certificates are enrolled, the administrator shifts the Basic License Tool window to bundles view by selecting **Bundles** from the **View** pull-down menu. The IBM Software Solutions bundle, whose bundle ID is **BNDL005A**, is displayed.

To display the licenses within a bundle, double-click the bundle or select **Expand bundle** from the **Selected** pull-down menu.

To display license details, double-click the license, or select **Open as details** from the **Selected** pull-down menu.

To hide licenses within a bundle, double-click the expanded bundle, or select **Collapse bundle** from the **Selected** pull-down menu.

Deleting a Bundle

To delete the IBM Software Solutions bundle, the administrator performs the following steps:

1. Select the bundle.
2. Select **Remove bundle** from the **Selected** pull-down menu.

Command-Line Equivalent

To accomplish the same results (bundle enrollment, managing, and deletion), use the following commands.

To enroll multiple certificates:

```
i4blt –a –f  "smrtjava.lic master.lic email.lic datamst.lic"
```

To view bundles, with all licenses within them:

```
i4blt –lb
```

To remove the bundle:

```
i4blt –d –b BNDL005A
```

Chapter 6. Command Line Interface

This chapter describes how to use the command line interface (CLI).

In the descriptions of command syntax, the following conventions are used:

Bold type Code this item exactly as shown.

Italic type Replace this item with your own value.

Brackets [] The item or items in brackets are optional.

Braces {} Choose one item from the list of alternative items inside the brackets.

Vertical line | This OR sign separates the alternative items.

Three dots An ellipsis following an item means you can include more than one of the item. For example, [*filename*]... specifies an optional list of files.

The following commands are available:

i4blt	Basic license tool (page 126)
i4cfg	Configuration tool (page 156)
i4tv	Test verification tool (page 161)
i4target	Target view tool (page 162)
i4ls	Network license server process (page 164)
i4ls -b	Nodelocked license server process (page 166)
i4ls -a	Central registry process (page 168)
i4nbssvr	NetBIOS socket server process (page 168)
i4lct	License creation tool (page 171)

i4blt - Basic License Tool

Primary Command Options

The i4blt command has the following primary command options:

(No option) Start the basic license tool GUI.

-a (Enroll a Product) Add products to a license database.

-U (Update a Product) Update the number of licenses you enrolled, update the hard stop / soft stop policy and high-water mark when enabled on the product, switch from per-server to per-seat licenses, and set the threshold value of a customer-managed product.

-E (Extract and Distribute Licenses)
Extract and distribute licenses from a network compound password of a given product to the servers.

-d (Delete a Bundle, Product License or an Application Client Identifier)
Delete bundles, delete products from a license database, or delete application client identifiers from the central registry of application clients.

-R (Reserve Licenses; Delete or Update Reserved Licenses)
Reserve reservable licenses for use by a specific user, group, or node; deletes reserved licenses; updates reservation status.

-C (Clean Up Stale Licenses) Update the number of concurrent, reservable, per-server, and concurrent nodelocked licenses in use.

-l (Display a List) List license database information about servers, vendors, bundles, products, and licenses.

-s (Display Product License Status)
Gather status information about license usage.

-r (Generate a Report) Report on information recorded in the log file of a license server.

-x (Delete Log Entries) Delete license server and central registry log file entries.

-m (Monitor and Log Threshold Events)
Monitor and log the threshold messages.

-H (Administer High-Availability Licensing)

Create a cluster of network license servers; add servers to an existing cluster; display cluster status; activate and deactivate servers in a cluster.

-h (Display Help)

Display command syntax and usage information about the basic license tool CLI.

General Rules for the i4blt command

1. A user with administrator authority can run i4blt command with all its options. Other users can run the -l (Display a List), -s (Display a Product License Status), -r (Generate a Report), and -h (Display Help) options and the equivalent GUI actions, allowing them to display the content of the license and license users databases.
2. The parameters within any of the following name specifications are positional:
   - vendor_information (*vendor_name vendor_id vendor_password*)
   - product_information (*product_name product_version license_password license_annotation*)
   - administrator_information (*administrator_name company_name address additional_info*)
   - user_information (*user_id user_group user_node*)
3. All the following name specifications must be enclosed within double quotation marks (for example: "*vendor_name vendor_id vendor_password*").
   - vendor_information (*vendor_name vendor_id vendor_password*)
   - product_information (*product_name product_version license_password license_annotation*)
   - administrator_information (*administrator_name company_name address additional_info*)
   - user_information (*user_id user_group user_node*)
   - signature_information
4. When a list of values (such as server names, vendor names, product names, or user names) is entered as a parameter, the list must be enclosed in double quotation marks. For example:

   ```
   i4blt -r3 -u "katie dustin emily adam"
   ```
5. A name that contains character spaces must additionally be enclosed within single quotation marks. If multiple blanks within the name must be preserved, each must be preceded by a backslash. For example:

   ```
   -v "'IBM Corporation'"
   ```

   ```
   -p "'Corel\ \ - System' 1.1"
   ```

6. The parameters you specify in any of the command options (for example, server names, vendor names, and product names) are case-sensitive.
7. The following command parameters have a limited length.

   group_name 31 characters

   server_name 31 characters

   Attention: The first three character positions are reserved for the string `ip:` Any server name that is longer than 28 characters is truncated on the right.

   user_name 31 characters.
8. You can display help on i4blt command options as follows:
   - To get help on just the -a, -U, -E, -d, or -m option:

     ```
     i4blt -option
     ```
   - To get help on just the -R, -l, -r, x, or -H option:

     ```
     i4blt -optionh
     ```

   Examples

   - Display the i4blt -E syntax:

     ```
     i4blt -E
     ```
   - Display the i4blt -r syntax:

     ```
     i4blt -rh
     ```

-a Enroll a Product

This option adds a product to the license database on the license server that you specify. Use the `i4blt -a` command to add a new product and its initial licenses to a license server database. You can also use this command to add licenses for existing vendor-managed products.

You can add product license information to a server in two ways:

- If you got the product license information in the form of an enrollment certificate file, you can install the product importing the enrollment certificate. You can enroll more than one product at the same time.
- If you got the product license information in a format other than an enrollment certificate file, you must enter the product information manually.

Syntax: If you have the enrollment certificate file:

i4blt -a
[**-n** *server_name*]**-f** *filename* [*filename*]...
[**-R** a*dministrator_name* [*company_name address additional_info*]]
[**-T** *enrolled_licenses*]
[**-I** *signature_information*]

If you do not have the enrollment certificate file:

i4blt -a
[**-n** *server_name*]
-v *vendor_name vendor_id vendor_password*
-p *product_name product_version license_password* [*license_annotation*]
[**-R** *administrator_name* [*company_name address additional_info*]]
[**-T** *enrolled_licenses*]
[**-I** *signature_information*]
[**-S** { *serial_number* | *bundle_id* }]

Parameters:

-n *server_name*

Specifies the name of the license server to which you intend to add the product. If **-n** is omitted:

- If the product is customer-managed, and the licenses are network licenses, they are enrolled on the central registry.
- If the product is customer-managed, and the license is nodelocked, it is enrolled on the local workstation.
- If the product is vendor-managed, and the enrollment certificate file specifies a target ID, the licenses are enrolled on that workstation.
- If the product is vendor-managed, and the enrollment certificate file does not specify a target ID, the licenses are enrolled on the local workstation.

-f *filename[filename]...*

(Only if you have the enrollment certificate) The complete path and file name of the enrollment certificate file containing the product license information that you intend to add. You can specify a list of file names if you want to enroll more than one certificate at a time.

-v *vendor_name vendor_id vendor_password*

(If you do not have the enrollment certificate)

vendor_name

The name of the vendor that manufactured the product that you intend to add.

vendor_id

The unique vendor ID string for the vendor that you specify in the *vendor_name* argument.

vendor_password

The unique vendor password string for the vendor that you specify in the vendor_name argument.

-p *product_name product_version license_password* **[** *license_annotation* **]**
(If you do not have the enrollment certificate)

product_name
The name of the product that you want to install.

product_version
The version of the product that you specified in the product_name parameter.

license_password
The unique license password string associated with the product.

license_annotation
The license annotation information (if any) provided by the vendor.

-R *administrator_name* **[** *company_name address additional_info* **]**
(For customer-managed use products only)

administrator_name
The name of the administrator who performs the operation. This parameter is required.

company_name
The name of your company.

address The address of your company.

additional_info
Comments, notices to future users, or other information about the initial enrollment of this product.

| If you enroll several products, the value you specify for these options
| apply to all the license certificates.

-T *enrolled_licenses*
(For customer-managed use products only) The number of licenses you have acquired from the software supplier. This parameter is required.

| If you enroll several products, the value you specify for this option
| applies to all the license certificates.

-I *signature_information*
(for customer-managed use products only) Information about the user issuing the command, to be stored with the signature stamp.

| **-S {** *serial_number* **|** *bundle_id* **}**
|

The serial number of a custom configuration license, or the bundle identifier for a bundle. The number is a string of up to 31 alphanumeric characters that uniquely identifies a custom configuration or a bundle.

Examples: Add a customer-managed use product:

```
i4blt -a
-v "Venus 4ca0fd5cf000.0d.00.02.1a.9a.00.00.00 kz5esmu69hzyw"
-p "timer 1.1 wzx3ewdfrvu4v64d53bbrkzhheaaaaa"
-R "Alex IBM Rome" -T 100
-I "'Alex Marshall'"
```

Add a vendor-managed use product:

```
i4blt -a -n thelma
-p "scena 1.0 suf0fpeixfi5v78a22xxrkzhheaaaaa"
-v "Operatix 7gp4ac8jj000.0d.00.02.1a.9a.00.00.00 lb7usud93jdna"
```

-U Update a Product

This option is valid only for customer-managed use products. It is issued for the following purposes:

- To update the number of licenses you enrolled. Use it when you acquire new licenses for an already enrolled customer-managed use product, to update the total number of licenses you are entitled to use. In the case of a network compound password, the licenses must be distributed after the update to make them available to end users.
- For a product with per-server / per-seat licenses, to switch from per-server to per-seat licenses.
- For a product with the hard stop / soft stop policy enabled, to change the hard stop / soft stop policy and to reset the high-water mark.
- To update the threshold value of a product.

Syntax:

i4blt -U
-v *vendor_name*
-p *product_name product_version*
[**-n** *server_name*]
[**-T** *enrolled_licenses*]
[**-S** *enable_switch* [**yes**]]
[**-H** *hard_soft_mode* [**yes** | **no**]]
[**-M** *hwm_reset*]
[**-t** *threshold* [**1...100**]]
[**-I** *signature_information*]

Parameters:

-v *vendor_name*

The name of the vendor that manufactured the product that you intend to update.

-p *product_name product_version*

The information on the licensed product that you intend to update.

product_name

The name of the product for which you have acquired the new licenses.

product_version

The version of the product that you specified in the product_name parameter.

-n *server_name*

The name of the license server on which you want to update product information. This parameter is required if the product has nodelocked licenses and you are updating the product on a remote nodelocked license server. It is the name of the nodelocked license server. If you are updating the product on the local nodelocked license server, omit the -n parameter. If the product has network licenses, this parameter need not be specified, because the server is the central registry license server.

-T *enrolled_licenses*

The total number of licenses you have for the specified product; that is, the number of licenses you had, plus the new licenses.

-S *enable_switch*

Use this parameter to migrate the license from per-server to per-seat. To use the per-seat license remember also to enroll the per-seat certificate. The only allowed value for -S is yes. When the licenses have been changed to per-seat, you cannot go back to per-server licenses.

-H *hard_soft_mode*

Use this parameter to switch the product behavior from hard stop to soft stop and vice versa. You can do it only on products the vendor has enabled to allow hard stop/soft stop switching. Allowed values for -H are:

no	Set the soft stop
yes	Set the hard stop

-M *hwm_reset*

Use this parameter to reset the high-water mark to 0. You can do it only on products the vendor has enabled to soft stop.

-t *threshold*
: Use this parameter to set a specific value for the threshold value of a customer-managed product. Allowed values are 1 to 100.

-I *signature_information*
: Information about the user issuing the command, to be stored with the signature stamp. Use this parameter with the -T parameter.

Examples: The Test Compiler product (version 1.1, vendor Psychosync) has network licenses. Update the number of licenses to 50.

```
i4blt -U -v "Psychosync" -p "'Test Compiler' 1.1"
-T 50 -I "'Paula'"
```

For the same product, set the soft stop policy and reset the high-water mark.

```
i4blt -U -v "Psychosync" -p "'Test Compiler' 1.1" -H no -M
```

For the ScreenPic product (version 2, vendor ArtTools), update to 5 the number of nodelocked licenses on server *Virginia*:

```
i4blt -U -n Virginia -v "ArtTools" -p "ScreenPic 2" -T 5
```

-E Extract and Distribute Licenses

Use the i4blt -E command to extract licenses from an installed network compound password and distribute them to the network license servers.

Syntax:

i4blt -E
-n *origin_server_name*
-v *vendor_name*
-p *product_name product_version*
-A *license_number_per_server*
-w *target_server_names*
[**-I** *signature_information*]

Parameters:

-n *origin_server_name*
: The name of the server where the network compound password is enrolled.

-v *vendor_name*
: The name of the vendor that manufactured the product whose licenses you want to distribute.

-p *product_name product_version*
: The information on the licensed product whose licenses you intend to distribute.

product_name
The name of the product whose licenses you want to distribute.

product_version
The version of the product that you specified in the product_name parameter.

-A *license_number_per_server*
The number of licenses for the specified product you want to distribute on each of the servers specified after the -w parameter.

-w *target_server_names*
The servers on which you want to distribute the licenses.

-I *signature_information*
(For customer-managed use products only) Information about the user issuing the command, to be stored with the signature stamp.

Examples: Extract and distribute 10 licenses to each of the servers **Louise** and **Hall**, for the Test Compiler product, Version 1.1 of vendor Psychosync, installed on server **Thelma**:

```
i4blt -E -n "Thelma" -v "Psychosync" -p "'Test Compiler' 1.1"
-A 10 -w "Louise Hall" -I "'Paula'"
```

-d Delete a Product License or Bundle

This option deletes a product license or bundle from the license database on the license server that you specify, or an application client identifier from the central registry.

Syntax:

i4blt -d
-n *server_name*
-v *vendor_name*
-p *product_name product_version*
{ **-t** *timestamp* | **-A** *ACID* | **-b** *bundle_id* }
[**-I** *signature_information*]

Parameters:

-n *server_name*
Either of the following:

- The name of the license server from which you intend to delete the product license.
- To delete a high-availability license, the name of one of the servers in the cluster on which the license is enrolled.

To delete a high-availability license, issue the i4blt command in the following format:

```
i4blt -d -n server_name -v vendor_name -p product_name
product_version -t timestamp
```

where *server_name* identifies one of the servers in the cluster on which the license is enrolled.

-v *vendor_name*
Name of the vendor whose product license you intend to delete.

-p *product_name product_version*
The information on the licensed product whose licenses you intend to delete.

product_name
Name of the product whose license you intend to delete.

product_version
Version of the product whose license you intend to delete.

-t *timestamp*
Unique timestamp of the product license that you intend to delete. To get the timestamp, issue the following command:

```
i4blt -lp -p "product_info" -i
```

Do not specify the timestamp when you delete an application client identifier.

-A *ACID*
Unique identifier of the application client identifier of an application client you want to delete from the central registry. After deletion the application client no longer has the license to use the specified product. To get the application client identifier, issue the following command:

```
i4blt -s -lpt -v "vendor_name" -p "product_info"
```

Do not specify **-A** when you delete a product.

| **-b** *bundle_id*
| The identifier of the bundle you want to delete.

-I *signature_information*
(For customer-managed use products only) Information about the user issuing the command, to be stored with the signature stamp. Use this parameter when deleting a product license.

When the last license for the only remaining product of a vendor is deleted, the vendor is automatically deleted from the license database. Vendor-managed compound passwords and use-once licenses cannot be deleted until they expire.

Examples: Delete a bundle whose ID is AZ57561C:

```
i4blt -d -b AZ57561C
```

Delete an expired license for a VectorComp Corporation product called EZ-Vectors Version 1.0. The unique timestamp of the license to be deleted from the database on server **saturn** is 781401788:

```
i4blt -d -n saturn -v "'VectorComp Corporation'" -p
"EZ-Vectors 1.0" -t 781401788
```

Delete an application client whose application client identifier is **thelma** from the central registry. After this command the application client will no longer have licenses for the product EZ-Vectors Version 1.0 of VectorComp Corporation vendor:

```
i4blt -d -v "'VectorComp Corporation'" -p "EZ-Vectors 1.0"
-A thelma
```

-R Reserve Licenses; Delete or Update Reserved Licenses

Use **i4blt -R** to reserve reservable licenses and to delete or update the reservation status of reserved licenses.

Syntax:

i4blt -R *action_type* [**r** | **d** | **u**]
-n *server_name*
-v *vendor_name*
-p *product_name product_version*
[**-t** *timestamp*]
[**-A** *license_number*]
[**-g** *end_date*]
[**-H** *end_time*]
[**-u** *user_id user_group user_node*]

Parameters:

action_type
To reserve licenses, **r**; to delete licenses, **d**; to update an existing reservation, **u**.

-n *server_name*
The name of the server where the product license is enrolled.

-v *vendor_name*
The name of the vendor that manufactured the product.

-p *product_name product_version*

The information on the licensed product whose licenses you intend to reserve, delete, or update.

product_name

The name of the product.

product_version

The version of the product.

-t *timestamp*

Unique timestamp of the product license from which you intend to reserve, or that you intend to delete or update. To get the timestamp, issue the following command:

```
i4blt -lp -p "product_info" -i
```

If you are reserving licenses (option **-R r**), the timestamp is optional. If it is omitted, the first usable reservable license is used.

-A *license_number*

The number of licenses you intend to reserve. If you are updating a reservation (**-R -u**) or deleting licenses (**-R -d**), do not specify -A.

-g *end_date*

The end date of the new or updated reservation (mm/dd/yyyy). The latest allowed expiration date of a reservation is 12/31/2037. If you are deleting licenses (**-R -d**), do not specify **-g**.

-H *end_time*

The end time of the new or updated reservation (hh:mm). If you are deleting licenses (**-R -d**), do not specify **-H**.

-u *user_id user_group user_node*

The identification of the user, group, and node for which a license is being reserved or a reservation is being changed. Any of these values may be *****, meaning “any”. If you are deleting licenses (**-R -d**), do not specify **-u**.

Examples: Reserve three licenses for the Test Compiler product, taken from the reservable license identified by the timestamp 389588975, Version 1.1 of vendor Psychosync for any member of the testers group. They expire March 2, 1998, at 11:00.

```
i4blt -R r -v "Psychosync" -p "'Test Compiler' 1.1"
-t 389588975 -A 3 -g 03/02/1998 -H 11:00 -u "* testers *"
```

-C Clean Up Stale Licenses

Use **i4blt -C** to update the number of in-use concurrent, reservable, per-server, and concurrent nodelocked licenses.

When you issue this command, LUM Runtime polls all the license servers that have granted licenses of these types and verifies that the licenses are still in use. If any stale licenses are found, they are removed from the number of in use licenses.

Syntax:

i4blt -C
[**-F** *server_type* { **l** | **w** | **a** }]
[**-n** *server_names*]
[**-v** *vendor_names*]
[**-p** *product_names*]

Parameters:

-F *server_type*
A filter on the type of server to be searched. Specify **l** for nodelocked license servers, w for network license servers, or **a** (the default) for both network license servers and nodelocked license servers.

-n *server_names*
The names of the servers where the products are enrolled.

-v *vendor_names*
The name of the vendors that manufactured the products whose licenses are in use.

-p *product_names*
The names of the products whose stale licenses you want to clean up.

Examples: Clean up stale licenses for the Graphics product of vendor Alpha on servers **Thelma**, **Hall**, and **Louise**:

```
i4blt -C -n "Thelma Hall Louise" -v "Alpha" -p "Graphics"
```

Clean up stale licenses for the Graphics product of vendor Alpha on all nodelocked license servers in the network.

```
i4blt -C -F l -v "Alpha" -p "Graphics"
```

-l Display a List

You can use this option to display a list of servers, vendors, bundles, products, or licenses. You can also use it to display details about individual products or individual licenses.

Syntax:

i4blt -l *list_type* [**n** | **s** | **v** | **p** [**-i**] |**b** | **l** [**-k**]]
[**-F** *server_type* { **l** | **w** | **a** }]
[**-n** *server_names*]

[**-v** *vendor_names*]
[**-p** { *product_name*[**?***product_version*] } ...]
[**-b** *bundle_id*]
[**-u** *user_names*]

Parameters:

list_type

Indicates the type of information that you want to list. You can specify any one of the following list types:

-ln or -ls

To display a list of active license servers.

Filters:

- You can use the **-F** filter option to display a list of active network license servers or nodelocked license servers.
- Do not specify the **–b**, **-n**, **-v**, **-p**, or **-u** filter option with this parameter.

-lv To create a vendor list.

Filters:

- To list vendor information gathered from a specific type of license server, use the **-F** filter option to specify nodelocked license servers or network license servers.
- To list vendor information gathered from servers that you specify, use the **-n** filter option followed by one or more server names.
- Do not specify the **-b**, **-v**, **-p**, or **-u** filter option with this parameter.

-lp [-i] [-k]
To create a product list.

Filters:

- To list product information gathered from a specific type of license server, use the **-F** filter option to specify nodelocked license servers or network license servers.
- To list product information gathered from servers that you specify, use the **-n** filter option followed by one or more server names.
- To list product information on products from particular vendors, use the **-v** filter option, followed by one or more vendor names.
- To list product information on particular products, use the **-p** filter option, followed by one or more product names.
- To list information on users who are currently using the products that you specify, use the **-u** filter option, followed by one or more user names.
- Do not specify the **-b** filter option with this parameter.

Specify the **-i** option to display detailed information about each product in a product list.

Specify the **-k** option to remove expired licenses from the list.

-lb [-b *bundle_id* **]**
To display a list of bundles or bundle components.

Filters:

- To list only the contents of a specified bundle, use the **–b** filter option, followed by the bundle identifier.
- To list bundles or bundle components for a specific type of license server, use the **-F** filter option to specify nodelocked license servers or network license servers.

- To list bundles or bundle components for servers that you specify, use the **-n** filter option followed by one or more server names.
- To list bundles or bundle components from particular vendors, use the **-v** filter option, followed by one or more vendor names.
- Do not specify the **-p**, **-u**, or **-t** filter option with this parameter.

-ll

To create a list of individual licenses. The output includes all the information you get by specifying **lp** with the **-i** option, plus, for products with concurrent licenses that are administered in a high-availability environment, information about the cluster and servers within the cluster.

Filters:

- To list license information gathered from a specific type of license server, use the **-F** filter option to specify nodelocked license servers or network license servers.
- To list license information gathered from servers that you specify, use the **-n** filter option followed by one or more server names.
- To list license information on products from particular vendors, use the **-v** filter option, followed by one or more vendor names.
- To list license information on particular products, use the **-p** filter option, followed by one or more product names.
- To list information on users who are currently using the licenses that you specify, use the **-u** filter option, followed by one or more user names.

High-Availability Output:

- Cluster name
- **For each server in the cluster**:
  - Server name
  - Server status:

    Serving
    Running, serving licenses

    Waiting
    Server is ready, but the cluster is in incomplete or inactive state

Unavailable
Not started

Reserve
In reserve in case a serving server becomes unavailable

Not activated
Defined as a member of the cluster but the administrator has not yet activated the server or has deactivated the server

– Percentage of licenses being served by this server

– Target ID

– Number of licenses served by this server

– Number of in-use licenses served by this server

-F *server_type*
A filter on the type of server to be searched. Specify **l** for nodelocked license servers, **w** for network license servers, or **a** (the default) for both network license servers and nodelocked license servers.

-n *server_names*
Names of the servers about which you want to display information in a vendor or product list.

-v *vendor_names*
Names of the vendors about whose products you want to display information in a product list.

-p { *product_name***[?***product_version* **] } ...**
Names of the products and, optionally, their versions about which you want to display information in a product list.

-u *user_names*
Names of users about whom you want to display license usage information in a product list.

-i
Specify the **-i** option in conjunction with a product list (**i4blt -lp**) to display the following detailed license usage information about an individual licensed product in a product list:

- Vendor name
- Vendor ID
- Product name
- Product version
- Product ID
- Licenses (total on all the selected servers)
- In-use licenses (total on all the selected servers)

- **For each license instance:**
  - Number of licenses
  - License type
  - Capacity type
  - Bundle component
  - Server on which the license is installed
  - License annotation (if any)
  - Serial number or bundle ID (if any)
  - Start date
  - Expiration date
  - Time stamp
  - Password use control level
- **For products with customer-managed use control and per-server, per-seat, or concurrent nodelocked licenses**, the following information is also displayed:
  - High-water mark licenses
  - Threshold value
  - Soft stop
  - Soft stop enabled
- **For products with customer-managed use control and use-once nodelocked licenses**, the following information is also displayed:
  - Threshold
- **For per-seat licenses**, the following information is also displayed:
  - Enablement flag
- **For reservable licenses**, the following information is also displayed:
  - Number of reserved licenses
  - Number of unreserved licenses
- **For reserved licenses**, the following information is also displayed:
  - User for whom licenses are reserved
  - Group for which licenses are reserved
  - Node for which licenses are reserved
- **For concurrent, concurrent nodelocked, and per-server licenses**, the following information is also displayed:
  - Multiuse rules (if any)
- **For try-and-buy licenses**, the following information is also displayed:
  - Try-and-buy flag
- **For compound passwords**, the following information is also displayed:
  - Derived license type
  - Aggregate duration
  - Derived start type

- Derived expiration date
- **For products with customer-managed use control and concurrent or reservable licenses**, the following information is also displayed:
  - Enrolled licenses
  - Distributed licenses
  - To be distributed licenses
  - High-water mark licenses
  - Threshold value
  - Soft stop
  - Soft stop enabled
- **For products with customer-managed use control and use-once licenses**, the following information is also displayed:
  - Enrolled licenses
  - Distributed licenses
  - To be distributed licenses
  - Threshold value

Examples: List all servers:

```
i4blt -ln
```

List all vendors on all servers:

```
i4blt -lv
```

List all vendors on all network license servers:

```
i4blt -lv -F w
```

List all vendors on server **Hall**:

```
i4blt -lv -n Hall
```

List all products on server **Hall**:

```
i4blt -lp -n Hall
```

List all products on server **mercury** provided by vendors Opticon, Inc. and Cybertronics Ltd.:

```
i4blt -lp -n mercury -v "'Opticon, Inc.' 'Cybertronics Ltd.'"
```

List detailed information for the product PsychoSynch on server **venus**:

```
i4blt -lp -n venus -p PsychoSynch -i
```

List all of the products on the server **neptune** that are currently being used by the user **Alex**:

```
i4blt -lp -n "neptune" -u Alex
```

-s Display Product License Status

Use this option to display information about current product usage on specified license servers.

Syntax:

```
i4blt -s
[ -l list_type [ c | pt | ps | ru | rr | cn ] ]
[ -n server_names ]
[ -v vendor_names ]
[ -p { product_name[?product_version ] } ... ]
[ -u user_names ]
```

Parameters:

-l *list_type*

Indicates the type of license usage you want to list. You can specify one of the following list types:

c To display information related to concurrent users of concurrent licenses.

pt To display information related to application clients that use per-seat licenses.

ps To display information related to users of per-server licenses.

ru To display information related to users of unreserved reservable licenses

rr To display information related to users of reserved licenses.

cn To display information related to users of concurrent nodelocked licenses.

If you omit **-l**, the default value **c** is used.

-n *server_names*

The name of each of the license servers for which you want to display product usage information.

If you omit the -n parameter, the display defaults to all servers in your cell. This parameter is ignored if you use **pt**, **ps**, **rr**, or **cn** as the list type.

-v *vendor_names*

The name of the vendor (or vendors) about whose products you want to display information.

-p { *product_name*[**?***product_version* **] } ...**

Names of the products and, optionally, their versions about which you want to display information.

-u *user_names*

Use the optional **-u** *user_names* argument to display product usage information for the specified products that are currently in use by the named users.

This command displays the following information for the servers, vendors, products, and users that you specify:

- Vendor name
- Product name
- Product version
- Total number of installed licenses
- Number of licenses currently in use
- Number of soft stop licenses currently in use
- Number of licenses not in use
- Number of queued users

For each user who currently holds a license, the following information is displayed:

- User name
- Node name
- Group name
- Number of licenses the user has been granted
- Check-out date for each granted license

Examples: Display current license availability and usage information for concurrent licenses of the Monolith Inc. product Megamail/2 on server **uranus**:

```
i4blt -s -lc -n "uranus" -v "'Monolith Inc.'" -p "Megamail/2"
```

-r Generate a Report

root and non-root users can use this option to list server, event, vendor, product, and user information on specified license servers.

Syntax:

i4blt -r *report_type* [**1** | **2** | **3** | **4** | **5** | **6**]
[**-b** *start_date*]
[**-g** *end_date*]
[**-e** *event_filter*]
[**-F** *server_type* { **l** | **w** | **a** }]
[**-n** *server_names*]

```
[ -v vendor_names ]
[ -p { product_name[?product_version ] } ... ]
[ -u user_names ]
```

Parameters:

-r *report_type*

Specifies the type of report to generate. The following report types are available:

1 - Standard Event Report

Displays detailed information about significant events occurring on the license servers that you specify. Available for all license types.

2 - License Request by Product Report

Displays statistical information about the use of the licenses of a product in the time interval you specify. For each product, it reports the licenses requested, the licenses granted, and the percentage of rejections. Not available for simple nodelocked or use-once (nodelocked or network) licenses.

3 - License Request by User Report

Displays statistical information about the use of products by users in the time interval you specify. For each user, it reports the licenses requested, the licenses granted, and the percentage of rejections for each product the person is using. Not available for simple nodelocked or use-once (nodelocked or network) licenses.

4 - License Use by Product Report

Displays statistical information about the use of the licenses of a product in the time interval you specify. For each product, it lists the maximum number of concurrent nodes that used the product, the maximum number of concurrent users, and the average time of use of the product. Statistical information about events appears only after the license has been released. Not available for per-seat, simple nodelocked, or use-once (nodelocked or network) licenses.

5 - License Use by User Report

Displays statistical information about the use of the licenses of a product in the time interval you specify. For each user, it lists the number of times each product was invoked, and the average time the user used each product. Statistical information about events appears only after the license has been released. Not available for per-seat, simple nodelocked, or use-once (nodelocked or network) licenses.

6 - Customer-Managed Use Audit

Reports the following information for customer-managed use product transactions:

- Vendor name

- Product name
- Product version
- Administrator information
- Time stamp of the event
- Number of licenses involved in the transaction
- Event list (product enrolled, license distributed, license deleted, license updated, per-server/per-seat license migrated)
- Signature stamp (user, group, and node)
- Signature information

Available for all license types.

-b *start_date*

Specifies the start date of a report. Express the date using the *mm/dd/yyyy* format. If you specify a start date and do not specify an end date, the report will include all information logged from the specified start date until the present.

-g *end_date*

Specifies the end date of a report. Express the date using the *mm/dd/yyyy* format. If you specify an end date and do not specify a start date, the report will include all information logged prior to (and including) the specified end date.

-e *event_filter*

You can use the event_filter argument to generate a Standard Event report on the following types of events which you specify, by number, on the command line. (Separate multiple event type arguments with a comma.)

1	All events
2	License-related events
3	Vendor messages
4	License database modifications
5	Error events
6	Server start and stop events
7	Fatal errors

-F *server_type*

A filter on the type of server to be searched. Specify **l** for nodelocked license servers, **w** for network license servers, or **a** (the default) for both network license servers and nodelocked license servers.

-n *server_names*

Names of the servers about which you want to display information.

-v *vendor_names*
Names of the vendors about whose products you want to display information.

-p { *product_name***[?***product_version* **] } ...**
Names of the products and, optionally, their versions about which you want to display information.

-u *user_names*
Names of users about whom you want to display license usage information.

Examples: Standard Event Report:

Report on license-related events (2) and server start and stop events (6) that were logged on server **neptune** since May 21, 1998:

```
i4blt -r1 -n "neptune" -b 05/21/1998 -e 2,6
```

License Use by Product Report:

Report current license usage information on server **saturn** for the products PsychoSynch, Megamail/2, EZ-Vectors, and DataVision:

```
i4blt -r4 -n "saturn" -p "PsychoSynch Megamail/2 EZ-Vectors DataVision"
```

License Request by User Report:

Report current license usage information on server **mercury** for users alex, ann, mary, christine, paul, and alby:

```
i4blt -r3 -n "mercury" -u "alex ann mary christine paul alby"
```

Customer-Managed Use Audit Report:

Report information about customer-managed use product transactions on all nodelocked license servers from May 1, 1999 to July 31, 1999:

```
i4blt -r6 -F l -b 05/01/1999 -g 07/31/1999
```

-x Delete Server Log Entries

This option deletes all entries before a specified delete date from the log file of the license servers that you specify. If one of the specified license servers has the central registry, the central registry log entries are also deleted. If the specified license server is the local node, the nodelocked license server log entries are also deleted.

Syntax:

i4blt -x *delete_date*
[**-F** *server_type* { l | **w** | **a** }]
[**-n** *server_names*]

Parameters:

-x *delete_date*
Specifies an end date for the delete operation. All log entries recorded before the delete date are removed from the log file. You must specify a delete date in the *mm*/*dd*/*yyyy* format. If you do not specify a *delete_date* value, all entries in the log file are deleted.

-F *server_type*
A filter on the type of server to be searched. Specify l for nodelocked license servers, **w** for network license servers, or **a** (the default) for both network license servers and nodelocked license servers.

-n *server_names*
Specifies the license servers from whose log file you want to delete the entries.

Examples: Delete all the log file entries recorded on server **neptune** before August 25, 1998:

```
i4blt -x 08/25/1998 -n neptune
```

Delete all the log file entries recorded on all nodelocked license servers before August 25, 1998:

```
i4blt -x 08/25/1998 -F l
```

-m Monitor and Log Threshold Events

This option displays the threshold messages and logs them if the threshold logging option is specified.

Syntax:

i4blt -m
[**-T** *percentage* [**1...100**]]
[**-A** *periodic_mode* [**yes** | **no**]]
[**-X** *frequency* [**1...1440**]]
[**-l** *log* [**yes** | **no**]]

Parameters:

-T *percentage*
Specifies the level of threshold value. It can be any number between 1 and 100. The default is 80. This is the percentage over which you want to log the level of usage of each product installed on all the servers. For instance, if you have 100 licenses of the product Icon

Editor Version 1.5, and you set the level of threshold to 10, a message appears in the vendor messages report only if more than 10 licenses are in use. If 20 licenses are being used, the message will say:

```
The 20% of licenses of Icon Editor 1.5 is in use.
```

Note that a customer-managed product may have its own threshold value, set with the **i4blt -U** command. Such a threshold value overrides the **-T** value.

-A *periodic_mode*
Specifies whether to check the license usage of the products only once, or periodically. Its values can be:

no To check the threshold conditions on the products once, immediately. If **-A** is omitted, the default is **-A no**.

yes To check the threshold conditions on the products periodically, with the frequency specified with the **-X** parameter.

-X *frequency*
Specifies the number of minutes between license usage checks. Enter a value between 1 and 1440. It is required if you set the **A** parameter to **yes**.

-l *log* Specifies whether the threshold messages must be logged on the license server to be reviewed with the report function.

Examples: Set the threshold percentage to 50% and set the check on the products' usage to every 4 hours:

```
i4blt -m -T 50 -A yes -X 240
```

-H Administer High-Availability Licensing

This option creates a cluster of network license servers, adds servers to an existing cluster, displays cluster status, and activates and deactivates servers in a cluster.

Syntax:

i4blt -H *action_type*
{ { **c** **-N** *cluster_name* **-n** *"server_names"* **-T** *initial_number_of_servers*} |
{ { **a** | **d** } **-N** *cluster_name* **-n** *"server_names"* } |
{ **s** { **-N** *cluster_name* | **-n** *"server_names"* } } }

Parameters:

action_type
Specifies the action to be taken:

c To create a cluster. With action type **c**, the **-N**, **-T**, and **-n** parameters are all required.

a To add a server to a cluster, or to activate a server in a cluster. With action type **a**, the **-N** and **-n** parameters are required.

d To deactivate a server in a cluster. With action type **d**, the **-N** and **-n** parameters are required.

s To request cluster status. With action type **s**, the **-N** or **-n** parameter is required. Use the **-N** parameter to get an overall view of cluster status. If you are having problems with cluster operation and you want to see a view of the cluster from the perspective of an individual server, use action type **s** with the **-n** parameter. In either case, the cluster status display includes the following information about the cluster:

- Cluster name
- Cluster ID (available only after the cluster switches to Active status for the first time)
- Cluster status:

  Active Running, serving licenses

  Change Pending Waiting for a change in the status of a server, or of the cluster, to be propagated to all the servers

  Inactive Not enough servers up and running

  Incomplete Not enough servers activated

- Initial number of servers
- Minimum number of members, maximum number of members, and minimum up and running for the cluster to work
- For each server in the cluster:
  - Server name
  - Server status:

    Serving Running, serving licenses

    Waiting Server is ready, but the cluster is in incomplete or inactive state

    Unavailable Not started

    Reserve In reserve in case a serving server becomes unavailable

    Not activated Defined as a member of the cluster but the administrator has not yet activated the server or has deactivated the server

| – Percentage of licenses being served

| – Target ID

| **-N** *cluster_name*

The name of the cluster to which the command is directed. The **-N** parameter is required if the action type is **c**, **a**, or **d**, and it is recommended when the action type is **s**.

-T *initial_number_of_servers*

The initial number of servers in the cluster that you are creating. The **-T** parameter is required if the action type is **c**, and is not valid if the action type is not **c**. The **-n** parameter must specify a number of servers equal to the value of **-T**.

-n *server_names*

The names of the servers to which the command is directed. The **-n** parameter is required if action type is **c**, **a**, or **d**, and it can be used when action type is **s**.

A cluster cannot contain a Windows 95 or Windows 98 workstation.

If the action type is **c**, this is the list of initial members of the cluster. You must specify a number of servers equal to the value of **-T**. After this command has been processed, the first server in the list is automatically activated. Issue `i4blt -H` again, using action type **a**, to activate each additional server.

At the time the `i4blt -H c` command is processed, all the servers specified must be up and running. If not, the command fails.

If the action type is **a** or **d**, this is the name of the server to be activated or deactivated. You must specify exactly one server. If action type is **a**, the server must be up and running when the command is processed. If not, the command fails.

If the action type is **s**, this is the name of any server that is currently activated in the cluster. The command returns cluster status from the perspective of this server.

Note: When the cluster is in **Change pending** status, different servers may return different data.

Examples: Create a cluster named **ruth** that has three members: anthony, germaine, and costanza:

```
i4blt -H c -N ruth -T 3 -n "anthony germaine costanza"
```

Activate the servers **germaine** and **costanza**:

```
i4blt -H a -N ruth -n germaine
i4blt -H a -N ruth -n costanza
```

Add the server **sandra** to the cluster:

```
i4blt -H a -N ruth -n sandra
```

Deactivate the server **germaine**:

```
i4blt -H d -N ruth -n germaine
```

Get an overall report of the status of the cluster **nobel**:

```
i4blt -H s -N nobel
```

Get a report of the status of the cluster **nobel** from the perspective of one of its activated members, **pirandello** (use this only for troubleshooting purposes):

```
i4blt -H s -n pirandello
```

-h Display Help

This option displays general syntax information for the basic license tool CLI.

Syntax:

i4blt -h

Examples: Display the i4blt syntax:

```
i4blt -h
```

i4cfg - Configuration Tool

Use the `i4cfg` command as an alternative to the configuration tool GUI to configure your workstation. Before coding the `i4cfg` command, see “Before You Configure” on page 71 to plan your configuration requirements. If issued with no options, the `i4cfg` command starts the configuration tool GUI.

Syntax

```
i4cfg
[ -a { { c,n,s,r } | { C,N,S,R } } ]
[ -e { a | { e,t,w,c,g,v,m,p,s } } ]
[ -l logfile_path ]
[ -S { a,n,s } ]
[ -R { a,n,s } ]
[ -g group_name ]
[ -b { "binding_list" | null } ]
[ -t "transport_list" ]
[ -d { option_string | all } ]
[ -start ]
[ -stop ]
[ -list ]
[ -h ]
```

Parameters

-a The roles the workstation is to play in your licensing environment. Code any combination of these values, optionally separated by commas:

- **c** Reset the current role of the workstation to network license client.
- **n** Reset the current role of the workstation to nodelocked license server.
- **s** Reset the current role of the workstation to network license server.
- **r** Reset the current role of the workstation to central registry license server.
- **C** Update the current role of the workstation to include network license client.
- **N** Update the current role of the workstation to include nodelocked license server.
- **S** Update the current role of the workstation to include network license server.

R Update the current role of the workstation to include central registry license server.

-b *"binding_list"*

The complete list of servers (network license servers, nodelocked license servers, and central registry license server) with which this workstation will communicate in a direct binding environment. Enclose the complete list in double quotes.

Specify the network license servers, nodelocked license servers, and central registry license server as follows:

```
'network protocol:network_address1 [ port_number1 ]
protocol:network_address1 [ port_number1 ]
protocol2:network_address2 [ port_number2 ] ... '
'nodelocked protocol:network_address1 [ port_number1 ]
protocol2:network_address2 [ port_number2 ] ... '
'registry protocol:network_address [ port_number1 ]'
```

where *protocol* is **ip**, **ipx**, or **netbios**.

Code `-b null` to delete all previously specified entries from the binding list.

-d *option_string*

Display the current configuration settings for the `i4cfg` options specified in *option_string*. Code all to see the current settings of all the options.

For example, `i4cfg -d elS` requests a display of which events are being logged, the path to the log databases, and a list of startup options showing which are enabled and which are disabled.

-e The list of events you want to be logged. Code **a** to log all events, or any combination of these values, optionally separated by commas:

e Errors

Describes server errors that do not stop the server, but return a status code and a message. This is logged by default.

t License timeout

Tells you that the server has canceled the request for a license because the check period expired. This is not logged by default.

w License wait

Tells you when a license request cannot be satisfied because no licenses are available, and the user is added to a queue. This is not logged by default.

c License check-in

Tells you when a licensed product has sent a check-in call to the server to notify that the product is running. This is not logged by default.

g License grant or release

Tells you when a license was granted or released. This is not logged by default.

v Vendor added or deleted

Tells you when a product of a new vendor was registered or deleted. This is logged by default.

m Vendor messages

Provides the log messages the vendor inserted in the enabled product. This is logged by default.

P Product added or deleted

Tells you when a new product was registered or deleted. This is logged by default.

s Server start or stop

Logs the successful start or stop of a license server. This is not logged by default.

-g *group_name*
The name of the group to which the user of this workstation belongs.

-h Displays command syntax and usage information about the configuration tool CLI.

-l *logfile_path*
The path in which you want log files to be stored.

-list Displays a list of active process.

-R Startup options that you want to disable. Code any combination of these values, optionally separated by commas:

| **a** Automatic startup of process at system startup (disabled by default). This parameter is valid only for Windows NT, Windows 2000, and Windows terminal Server workstations.

n Remote administration of nodelocked license server (disabled by default).

s Remote administration of network license server (enabled by default).

Note: This parameter is not valid for network clients.

-S Startup options that you want to enable. Code any combination of these values, optionally separated by commas:

a Automatic startup of process at system startup (disabled by default). This parameter is valid only for Windows NT, Windows 2000, and Windows terminal Server workstations.

n Remote administration of nodelocked license server (disabled by default).

s Remote administration of network license server (enabled by default).

Note: This parameter is not valid for network clients.

-start Starts all the processes you have configured to run on the workstation.

-stop Stops all the processes that are running on your workstation.

-t *"transport_list"*

Use this parameter to specify each protocol you will use and, optionally, to change the default port numbers, as follows:

```
"protocol 'netls_port,crls_port,nodls_port'"
```

where *protocol* is **ip**, **ipx**, or **netbios**. Specify one, two, or three protocols, separated by spaces, and enclose the entire transport list in double quotes.

The three subparameters for port numbers are positional. If you omit one, its value is reset to the default. For example:

```
"ip "
"ip '1516,10999,1215'"
"ip ',,11999'"
```

-u *"user_name"*

The name of the user of this workstation, enclosed in double quotes.

Examples

1. Configure a standalone nodelocked license server, specifying automatic startup of the server and customizing the path to the log files and the selection of events logged:

```
i4cfg -a n -S a -e evmps -l c:\ifor\ls\baratti
```

2. Configure a nodelocked license server in a network. Specify automatic startup of the server, make it possible to administer licenses on another nodelocked licensed server (**louise**) remotely, and customize the path to the log files and the selection of events logged.

```
i4cfg -a n -S n -e evmps -l c:\ifor\ls\baratti -g Infotech
-b 'nodelocked ip:louise'
```

3. Configure a network license server (**thelma**). Specify automatic startup of the server, customize the path to the log files and the selection of events logged, specify the user group, and configure to communicate via both TCP/IP and NetBIOS. Configure to communicate with:
   - Network license server **louise**
   - OS/2 network license server **bertini**
   - Nodelocked license server **louise**
   - Nodelocked license server **speedy**
   - Central registry license server **speedy**

   ```
   i4cfg -a s -S a,s -e cegvp -l c:\ifor\ls\baratti -g Infotech
   -t "ip netbios" -b "'network ip:thelma ip:louise
   os2@netbios:bertini' 'nodelocked ip:speedy ip:louise'
   'registry ip:speedy'"
   ```

4. Configure a network license client that will communicate with a workstation named **thelma** that is configured as both a network license server and the central registry license server. Specify the user group.

   ```
   i4cfg -a c -g Infotech -b "'network ip:thelma' 'registry ip:thelma'"
   ```

5. Configure a workstation named **thelma** as the central registry license server and as a network license server. Configure it to communicate with a network license server named **hydra**. Specify automatic startup of the servers, and specify the use group.

   ```
   i4cfg -a s,r -S a,s -g Infotech -b "'network ip:thelma ip:hydra'
   'registry ip:thelma'"
   ```

6. Cancel all entries previously made in the direct binding servers list:

   ```
   i4cfg  -b null
   ```

7. Display the command syntax and usage:

   ```
   i4cfg  -h
   ```

LUM Runtime Tools

This section contains information about LUM Runtime tools.

The tools are:

Test verification tool (i4tv)
Verifies that license servers are running properly.

Target view tool (i4target)
Displays the target ID of your workstation. The vendor of a licensed product may ask you to provide the target ID of the workstation on which the license is to be installed.

i4tv - Test Verification Tool

Use the test verification tool (i4tv) after the license servers are started to verify that they are running properly. The tool resides in the `\ifor\ls\bin` directory. A message describing a completed license transaction and a list of all license servers will be displayed.

| If a HAL cluster has been configured and activated, i4tv tries to request all 10 000 licenses of the HAL test product, to contact all the servers that are in Serving state.

Syntax:

i4tv { [**-n** *hostname*] [**-v**] | { **-h** | **-usage** | **-version** }
[**-p** *number_of_transactions*] }

Parameters:

-n *hostname* Checks that the specified workstation is running a network license server. It returns **0** if the hostname is running a network license server and **1** if the hostname is not running a network license server.

-v Displays progress messages during the license request operation.

-h Displays command usage information (same as **-usage**). This parameter is valid only when issued without other parameters.

-usage Displays command usage information (same as **-h**). This parameter is valid only when issued without other parameters.

-version Displays command version information. This parameter is valid only when issued without other parameters.

-p [*number_of_transactions***]**
Specifies the number of transactions to be completed before performance information is displayed. This information provides averages for the specified period. It can be used for tuning the system and for troubleshooting performance problems. The default value is 1000.

Examples: Run the i4tv test and verification tool:

```
i4tv
```

Check for the presence of the license server **pluto**:

```
i4tv -n pluto
```

i4target - Target View Tool

Use the target review tool (i4target) to display the target ID of your workstation. The tool resides in the `\ifor\ls\bin` directory.

Syntax:

i4target [**-O** | **-V** | **-o** | **-l** | **-h** | **-d** *devicename*] [**-z**] [**-v**]

Parameters:

-O Displays the target identifier of the workstation on which you issue the command, in the form that the license creation tool accepts.

-V Displays command version information.

-o Displays the operating system name of your workstation.

-l Displays all target IDs of the workstation, starting with the most secure (the one based on the network adapter, if it is available).

The hardware-based (win32mac) target ID is retrieved from the device driver that controls the network adaptor, rather than from the operating system. This mechanism results in the highest level of security and makes it possible for the target ID to work even when the workstation is not connected to the network.

For this mechanism to work, the following must be installed on the workstation where the license will be installed:

- A device that complies with NDIS 4.
- The latest level of the device driver. For IBM device drivers, you can check at `http://www.networking.ibm.com` to determine the latest level.
- TCP/IP.
- The i4target executable. If LUM Runtime is not yet installed on the workstation, you can download i4target from the LUM Web site (`http://www.software.ibm.com/is/lum`), or the vendor of the license-enabled products can supply it to customers.

If the prerequisites are not met, the hardware-based target ID is not generated.

-h Displays command usage information.

-d *devicename*
Specifies that the device driver *devicename* is to be used to retrieve the target ID.

-z Displays all the devices on the workstation; use it to check the available network devices before using the -d parameter to select a device driver manually.

-v Displays information in verbose mode.

The information that this parameter displays includes the NDIS level of the device driver.

The target ID is retrieved from the device driver that controls the network adapter. You may want to specify explicitly the device driver to be used to determine the target ID. You would do this if either of the following points were true:

- Your workstation has multiple network adapters and you prefer to use an adapter other than that automatically selected as the basis for the target ID.
- The target -Ol command did not return a target ID.

Perform the following steps:

1. Get the device name of the adapter you want to use, in one of the following ways:
   - For Windows NT and Windows 2000, to get the device name of the adapter you want to use, enter the command:

     ```
     ipconfig /all
     ```

     For example, for IBM token ring adapter IBMTRP1, the information returned includes the line:

     ```
     Token ring adapter IBMTRP1:
     ```
   - Alternatively for Windows NT or Windows 2000, or for Windows 95 or Windows 98, enter the command:

     ```
     i4target -z
     ```

     This displays all the devices on the workstation, and be sure the network device you want to use is listed.
2. Enter the command:

   ```
   i4target -d devicename
   ```

   For example, `i4target -d IBMTP1`.

LUM Runtime and NCS Processes

Read this section for reference information on LUM Runtime and NCS processes.

i4ls - Network License Server Process

The **i4ls** process starts the network license server on the local node. If the workstation is not configured to run the network license server, **i4ls** has no effect.

The parameters of **i4ls** override the corresponding settings in the i4ls.ini file.

Syntax:

i4ls [**-no** *event_list*] [**-v**] [**-z**] [**-l** *log_name*] [**-s**] [**-r**] [**-c**]

Parameters:

-no *event_list* Turns off logging of the events specified in *event_list*. Any combination of events is valid, but items in the list of events must not be separated by spaces or other characters. Following are the event types that you can specify:

l Grant and release licenses.

c Check in licenses. (Licensed products usually check in with the license server at regular intervals while a user is using the product.)

w Waiting events: these include wait events (a user was waiting for a license), wait grant events (a user was waiting for and then was granted a license), and wait remove events (a user was waiting for a license and then asked to be removed from the queues before a license was granted.)

v Vendor events: a vendor was added, renamed or deleted.

p Product events: a product was added, renamed, or deleted.

e Errors.

m Messages.

s Starts and stops of this license server.

t License time-out events. (When a licensed product fails to check in with the license server, it may stop running after it times out. The vendor of the product sets the time-out interval, which is how long a product can run after it has lost contact with the license server.)

-v LUM Runtime library verbose mode.

-z Debugging flag. Prints RPC debugging information.

-l *log_name* Overrides the default name and location of the file used to store log information. This allows the I/O activity to the files used by the license server to be spread across multiple file systems that may become important for large installations.

-s Instructs the license server to ignore attempts from administrators on remote systems to modify the license database. Records in the database remain readable by all instances of the LUM Runtime license administration tool.

-r Recovers files from the automatic backup version.

-c Specifies that this is a cold start, meaning that the license server restarts from scratch, as if it had granted no licenses before stopping.

Examples: Start a license server and do not log check-in, vendor, product, time-out, or message events:

```
i4ls -no cvptm
```

Start a license server, changing the default log file:

```
i4ls -l \ifor\ls\my_log
```

i4ls -b - Nodelocked License Server Process

The **i4ls -b** process starts the nodelocked license server on the local node. If the workstation is not configured to run the nodelocked license server, **i4ls -b** has no effect.

The parameters of **i4ls -b** override the corresponding settings in the i4ls.ini file.

Syntax:

i4ls -b [**-no** *event_list*] [**-v**] [**-l** *log_name*] [**-s**] [**-r**] [**-c**]

Parameters:

-no *event_list* Turns off logging of the events specified in the event list. Any combination of events is valid, but items in the event list must not be separated by spaces or other characters. Following are the event types that you can specify:

l	Grant and release licenses.
v	Vendor events: a vendor was added, renamed or deleted.
p	Product events: a product was added, renamed, or deleted.
e	Errors.
m	Messages.
s	Starts and stops of this license server.
t	Time out.

-v LUM Runtime library verbose mode.

-l *log_name* Overrides the default name and location of the file used to store log information. This allows the I/O activity to the files used by the license server to be spread across multiple file systems that may become important for large installations.

-s Instructs the license server to ignore attempts from administrators on remote systems to modify the license database. Records in the database remain readable by all instances of the basic license tool.

-r Recovers files from the automatic backup version.

-c Specifies that this is a cold start, meaning that the license server restarts from scratch, as if it had granted no licenses before stopping.

Examples: Start a nodelocked license server and do not log check-in, vendor, product, or message events:

```
i4ls -b -no cvpm
```

Start a nodelocked license server, changing the default log file:

```
i4ls -b -l \ifor\ls\my_log
```

Start a nodelocked license server, disabling remote administration from instances of the basic license tool on other workstations:

```
i4ls -b -s
```

i4ls -a - Central Registry License Server Process

The central registry is a LUM Runtime process that provides a mechanism for storing data pertaining to licensing information. There must be one and only one central registry license server. This ensures that the data is accurate and complete.

The basic license tool requires a central registry license server up and running to administer customer-managed use products.

You must double-check that you have started one and only one central registry license server in your licensing environment by issuing the following command on every LUM Runtime server:

```
i4cfg -list
```

If the workstation is not configured to run the central registry license server, **i4ls -a** has no effect.

The parameters of **i4ls -a** override the corresponding settings in the i4ls.ini file.

Syntax:

i4ls -a [**-no** *event_list*] [**-v**] [**-l** *log_name*] [**-r**] [**-c**] [**-z**]

Parameters:

-no *event_list* Turns off logging of the events specified in *event_list*. Any combination of events is valid, but items in the event list must not be separated by spaces or other characters. Following are the event types that you can specify:

l Grant and release licenses.

c Check in licenses. (Licensed products usually check in with the license server at regular intervals while a user is using the product.)

v Vendor events: a vendor was added, renamed or deleted.

p Product events: a product was added, renamed, or deleted.

e Errors.

m Messages.

s Starts and stops of this license server.

t Time out.

-v LUM Runtime library verbose mode.

-l *log_name* Overrides the default name and location of the file used to store log information. This allows the I/O activity to the files used by the license server to be spread across multiple file systems that may become important for large installations.

-r Recovers files from the automatic backup version.

-c Specifies that this is a cold start, meaning that the license server restarts from scratch, as if it had granted no licenses before stopping.

-z Debugging flag. Prints RPC debugging information.

i4nbssvr - NetBIOS Socket Server Process (Windows NT 4.0 and Windows 2000 only)

You need to run the NetBIOS socket server on any Windows NT or Windows 2000 workstation configured to communicate with an OS/2 workstation through the NetBIOS protocol. The OS/2 workstation must run the NetBIOS socket server as well.

Syntax:

```
i4nbssvr
```

i4lct - License Creation Tool

The license creation tool is intended for:

- Software vendors, to create test passwords while enabling a product
- Software vendors, to create production passwords and bundles
- Sales representatives, who can be provided with a compound password containing many licenses, from which they extract licenses for individual customers.

This tool is not intended for administrators or end users.

Run this command on a workstation where LUM Runtime is installed.

Notes:

1. The passwords you generate with the license creation tool of LUM Runtime Version 4.6 also work on LUM Runtime servers and clients of previous releases. High-availability licenses, introduced in Version 4.5.0, can be installed only on workstations running Version 4.5 or later. Licenses of types introduced in Version 4 (such as reservable and per-seat), cannot be installed on workstations running earlier releases of LUM Runtime. Custom configuration licenses, introduced in Version 4.5.5, can be installed only on workstations running Version 4.5.5 or later.

   In the enrollment certificate file, the **PasswordVersion** parameter is set as follows:

   8 If the password is for a capacity license or for bundle component licenses

   7 If the password is for a custom configuration license

   6 If the password is for a high-availability license (and can therefore be installed only on workstations running Version 4.5 or later)

   5 If the password is for a license type, or exercises a policy, introduced in Version 4 (and therefore is not installable on workstations running earlier versions)

   4 Otherwise

2. To create test passwords, use **test** as the value of the **-i**, **-k**, and **-v** parameters.

3. To create a bundle:
   - Create a license for each product in the bundle.
   - Use **b** as the value of the **-A** parameter.
   - Use the same value for the **-C** parameter for all products in the bundle.

   Optionally, you can create a separate *master license* for the bundle. The master license is used to check the consistency of the bundle. It does not appear as a product in the basic license tool.

If you create a master license:

- Use **b** as the value of the **-A** parameter.
- Use the same value for the **-C** parameter as you used for each product in the bundle.
- Use 102 as the value for the **-F** parameter.

If you create a master license, the bundle appears in the basic license tool as **incomplete** when either of the following conditions is true:

- The number of products in the package, or the number of licenses in a bundle, is different from the value specified in the **-n** parameter.
- The master license is not enrolled.

4. To extract licenses for individual customers from a compound password assigned to a sales representative, use **supplier** as the value of the **-k** parameter.
5. To create production licenses, vendors must acquire the license for this tool from IBM or from Isogon Corp.

   The address of Isogon Corp. is:
   Isogon Corporation
   330 Seventh Avenue
   New York, New York 10001
   U.S.A.
   Tel: (+1) 212-376-3200
   Fax: (+1) 212-376-3280

   Table 13 summarizes the valid combinations of license type, password use control level, password type, and enabled policies the vendor can specify with `i4lct`.

Table 13. Valid Uses of i4lct

License Type	Password Use Control Level	Password Type	Policies
Concurrent (**-l c**)	Customer-Managed (**-R c**)	Compound (**-w c**)	Hard Stop/Soft Stop (**-A s**)† Multiuse Rules (**-m**) License Annotation (**-a**) Capacity (**-F**)†
Concurrent (**-l c**)	Vendor-Managed (**-R v**)	Simple (**-w l**)	Soft Stop (**-A s**) Multiuse Rules (**-m**) License Annotation (**-a**) Custom Configuration (**-C**) Capacity(**-F**) Bundle (**-A b**)

Table 13. Valid Uses of i4lct (continued)

License Type	**Password Use Control Level**	**Password Type**	**Policies**
Concurrent (**-l c**)	Vendor-Managed (**-R v**)	Compound (**-w c**)	Soft Stop (**-A s**) Multiuse Rules (**-m**) License Annotation (**-a**) Capacity(**-F**)
Reservable (**-l r**)	Customer-Managed (**-R c**)	Compound (**-w c**)	Hard Stop/Soft Stop (**-A s**) License Annotation (**-a**)
Reservable (**-l r**)	Vendor-Managed (**-R v**)	Simple or Compound (**-w l** or **-w c**)	License Annotation (**-a**)
Use-Once (**-l u**)	Customer-Managed (**-R c**)	Compound (**-w c**)	License Annotation (**-a**)
Use-Once (**-l u**)	Vendor-Managed (**-R v**)	Simple or Compound (**-w l** or **-w c**)	License Annotation (**-a**)
Per-Seat (**-l pt**)	Customer-Managed (**-R c**)	Simple (**-w l**)	Hard Stop/Soft Stop (**-A s**) License Annotation (**-a**)
Per-Server (**-l ps**)	Customer-Managed (**-R c**)	Simple (**-w l**)	Hard Stop/Soft Stop (**-A s**) Multiuse Rules (**-m**) License Annotation (**-a**)
Simple Nodelocked (**-l n**)	Vendor-Managed (**-R v**)	Simple (**-w l**)	License Annotation (**-a**) Custom Configuration (**-C**) Capacity (**-F**) Bundle (**-A b**) Product-Specific Program (**-I**)
Simple Nodelocked (**-l n**)	Vendor-Managed (**-R v**)	Compound (**-w c**)	License Annotation (**-a**) Capacity (**-F**)
Simple Nodelocked (**-l n**)	Vendor-Managed (**-R v**)	Compound Nodelocked (**-w cn**)	Try-and-Buy (**-A t**)* License Annotation (**-a**) Capacity (-F)
Concurrent Nodelocked (**-l cn**)	Customer-Managed (**-R c**)	Simple (**-w l**)	Hard Stop/Soft Stop (**-A s**)† Multiuse Rules (**-m**) License Annotation (**-a**) Capacity (**-F**)† Bundle (**-A b**)
Concurrent Nodelocked (**-l cn**)	Vendor-Managed (**-R v**)	Simple (**-w l**)	Soft Stop (**-A s**) Multiuse Rules (**-m**) License Annotation (**-a**) Capacity(**-F**) Bundle (**-A b**)
Use-Once Nodelocked (**-l un**)	Customer-Managed (**-R c**)	Simple (**-w l**)	License Annotation (**-a**) Capacity (**-F**)

Table 13. Valid Uses of i4lct (continued)

License Type	Password Use Control Level	Password Type	Policies
Use-Once Nodelocked (**-l un**)	Vendor-Managed (**-R v**)	Simple (**-w l**)	License Annotation (**-a**) Capacity (**-F**)

* **When -w is set to cn, the try-and-buy attribute is required**. † Soft-stop does not work with capacity for customer-managed use licenses.

Syntax

i4lct *parameters*

Parameters required to generate a license:

-i {*vendor_id* | **create** | **test** }

-k { *vendor_key* | **test** | **supplier** }

-v { *vendor_name* | **test** }

-l *license_type*

-F *capacity_type*

-A b (required only for a product that is part of a bundle)

-C { *serial_number* | *bundle_id* } (required only for a custom configuration or for a product that is part of a bundle)

-p *product_id*

-N *product_name*

-w *password_type*

{ **-d** *duration* | **-e** *expiration_date* }

{ **-T** *target_id* **-t** *target_type* | **-X** *extended_target_id* **-x** *extended_target_type* }

Parameters required to generate a master license for a bundle:

-F 102

-N *bundle_name*

-r *revision*

-A b

-C *bundle_id*

Parameters valid only if **-w** is set to **c** or **cn** (compound or compound nodelocked passwords):

[**-S** *derived_start_date*]

[**-E** *derived_expiration_date*]

Parameter valid only if **-R** is set to **v** (vendor-managed product):

[**-n** *number_of_licenses* | *capacity_units*]

Parameter valid only if **-w** is set to **c** or **cn** (compound or compound nodelocked passwords) and **-R** is set to **v** (vendor-managed product):

[**-D** *aggregate_duration*]

Parameter valid only if **-l** is set to **c**, **cn**, or **ps** (concurrent, concurrent nodelocked, or per-server license):

[**-m** *multi-usage_specification*]

Optional parameters:

[**-a** *annotation*]

[**-A** *attributes*] (optional only for a product that is not part of a bundle)

[**-c** *customer_information*]

[**-C** *serial_number*] (optional only for products that are not a custom configuration or part of a bundle)

[**-L** *log_file*]

[**-O**]

[**-P** *16_bit_flag*]

[**-r** *revision]*

[**-R** *password_registration_level*]

[**-s** *start_date*]

Parameters valid only when entered without any other parameters:

[**-f** *batch_file_name*]

[**-h**]

[**-V** *version*]

[**-u** *upgrade_flag*]

[**-U**]

Parameters

-a *annotation* The license annotation string. Up to 255 characters for simple nodelocked licenses, up to 80 characters for all others.

-A *attributes* Possible values are:

s To enable the end user to modify the product policy from the default hard stop to soft stop. Valid only for customer-managed products (**-R** set to **c**).

t To specify a try-and-buy license. Valid only for vendor-managed products (**-R** set to **v**) with nodelocked licenses (**-l** set to **n**) and password type compound nodelocked (**-w** set to **-cn**).

b To specify a product that is part of a bundle, or to specify the master license for a bundle.

-c *customer_information*
Specifies additional customer details for logging purposes. This parameter is useful only if used with the **-L** *log_file* parameter.

-C { *serial_number* **|** *bundle_id* **}**
Specifies the serial number of a custom configuration license or the identifier of a bundle. This parameter is a string of up to 31 alphanumeric characters.

-d *duration*

The duration of the password. If the password type is license, this value indicates the number of days for which the licenses are valid. If the password type is compound, this value indicates the number of days during which license passwords can be derived from the compound password. Its maximum allowed value is 32 767.

For vendor-managed compound passwords, the product obtained by multiplying **-d** (duration) and **-n** (number_of_licenses) cannot exceed 2 147 483 647.

For example, if **-n** is 70 000, the maximum duration is 30 678 days (2 147 483 647/70 000).

You must specify at least one of **-d** and **-e**.

-D *aggregate_duration*
Valid only for vendor-managed products (**-R** set to **v**) and compound or compound nodelocked passwords (**-w** set to **c** or **cn**). This is the maximum aggregate duration, in days, of all licenses that are to be derived from a compound password. Its maximum allowed value is 2 147 483 647.

In the case of a try-and-buy license (**-w** set to **cn**, **-A** set to **t**, and **-l** set to **n**), this represents the duration of the try-and-buy license extracted from the compound password.

For example, a compound password from which 100 licenses may be derived might have an aggregate duration of 36 500 days. From this password there can be derived 100 1-year licenses, or 50 6-month licenses and 50 18-month licenses, and so on.

-e *expiration_date*

The end date of the password. The date format is *mm/dd/yyyy*. If the password type is license, this value indicates the end date beyond which the licenses are no longer valid. If the password type is compound, this value indicates the end date beyond which license passwords can no longer be derived from the compound password.

The latest expiration date that can be specified with the -e parameter is 02/05/2106. Note, however, that the standard time functions of the operating system do not properly handle expiration dates later than 12/31/2037, so it is recommended that you not create licenses that expire after that date. Note also that the current version of the operating system does not allow system dates later than 01/18/2038.

You must specify at least one of **-d** and **-e**.

Valid combinations of the start, duration, and end options are as follows:

-d	The start date defaults to the current date. `i4lct` calculates the expiration date for you.
-s and -d	`i4lct` calculates the expiration date for you.
-e and -d	`i4lct` calculates the start date for you.
-s and -e	`i4lct` calculates the duration for you.

-E *derived_expiration_date*

Valid only with compound or compound nodelocked passwords (**-w** set to **c** or **cn**). The date format is *mm/dd/yyyy*. This is the derived license end date, the date after which no license password derived from the compound password is valid.

-f *batch_file_name*

Specifies the fully qualified path and file name of a batch file containing the full `i4lct` command to issue the full `i4lct` command contained in such a file.

-F *capacity_type*

Specifies the capacity of the license.

capacity_type

One of the following values:

100 The units specified are a number of configured processors.

101 The units specified are a number of online processors.

102 The units specified are the number of components in a bundle. This unit code is used only for the master license of a bundle.

103 The units specified are a number of physical disks.

104 The units specified are a number of megabytes. Used only for physical memory.

xxx The units are vendor-specified, in the range 128 through 255.

-h Displays help for the `i4lct` command.

-i *vendor_id* Specifies the vendor ID. It can also take the following values:

create Generate a new vendor ID while generating a production password.

test Create test passwords.

-k *vendor_key* Specifies the vendor key. This must be an integer in the range 1 through 2 147 483 647, or one of the following values:

test Create test passwords.

supplier When you specify this value the license server must be up and running, and there must be a compound password enrolled for a vendor-managed use product.

By specifying this value you create an enrollment certificate file for a simple password extracted from the existing compound password. You specify the compound password by means of the other `i4lct` parameters.

The following example creates the certificate file for 497 concurrent licenses with duration 10 days, extracted from the compound password of the vendor-managed use product **cmpLev3** of the vendor Operatix:

```
i4lct -i 6pw4ci1xw000.0n.00.03.4g.5y.00.00.00 -k supplier
-n 497 -l c -d 10 -N "cmpLev3" -p 317 -r 1.0 -t any -T any
-v "Operatix" -w 1
```

Use this parameter when you need to provide sales representatives in other locations with a compound password for several licenses, for example. The representatives enroll the compound password, then generate the licenses for customers, extracting simple passwords from it. Thus you do not need to generate the enrollment certificate files for all the customers or to supply the production `i4lct` to all your representatives.

-l *license_type*

The license type. Use one of the following keywords:

c	Concurrent
cn	Concurrent nodelocked
n	Nodelocked
u	Use-once
un	Use-once nodelocked
ps	Per-server
pt	Per-seat
r	Reservable

Multiuse rules, **-m**, can be specified only if this parameter is set to concurrent, concurrent nodelocked, or per-server.

-L *log_file*

Specify the `i4lct` log file path and name. If you do not specify it the default is:

```
i4lct.log
```

in the **\ifor\ls\conf** directory.

-m *multi-usage_specification*

This argument is optional and is used to define multiuse rules for concurrent, concurrent nodelocked, and per-server licenses.

You can define conditions for multiuse of a single concurrent license as any combination of the following key letters: **u** (same user), **n** (same node), **g** (same group or same display, depending on the license-enabled application), **j** (same job ID), **a** (same accessor ID).

You can define conditions for multiuse of a single concurrent nodelocked or per-server license as any combination of the following key letters: **u** (same user), **g** (same group or same display, depending on the license-enabled application), **j** (same job ID), **a** (same accessor ID).

Specify the letters without spaces, commas, or other separators. For example, **-m un** means that if the user and node are the same as those associated with a previously granted license, granting a new concurrent access license is not required.

-n { *number_of_licenses* **|** *capacity_units* **}**
For a compound password, this is the maximum number of licenses or capacity units that can be derived from the password. It is valid only for vendor-managed products (**-R** set to **v**). Its maximum allowed value is 65 534. For customer-managed products, you cannot specify this parameter, and the value is set to 65 535.

For vendor-managed compound passwords, the product obtained by multiplying **-d** (duration) and **-n** (number_of_licenses) cannot exceed 2 147 483 647.

For example, if -n is 70 000, the maximum duration is 30 678 days (2 147 483 647 / 70 000).

-N *product_name*
The name of the product. It can be up to 31 alphanumeric characters long. All product name specifications must be enclosed within double quotation marks (*"product_name"*). Product name specifications are case-sensitive.

-O
Specify this option to generate, at the top of the enrollment certificate file, the command the end user issues to enroll the password. If the license is a type supported in releases of LUM Runtime earlier than Version 4.0, two commands are generated: the i4blt command for use with LUM Runtime Version 4 and the ls_admin command for use with previous releases. Otherwise, only the i4blt command is generated.

-p *product_id*
The product ID. This is an integer between 1 and 2 147 483 647 that identifies a vendor's licensed software product. Product IDs are used by the license server to distinguish between different products from the same vendor. Product ID must be unique among all the products you create licenses for.

-P *16_bit_flag*
The *product_id* field in the password is limited to 16 bits.

-r *revision*
A string that identifies a particular version of a product; by means of version identifiers, the license server can distinguish between products that use the same product ID. It can be up to 11 characters long.

-R *password_registration_level*
Specifies the password registration level. Its allowed values are:

c
Specify that the password is for a customer-managed use product.

v
Specify that the password is for a vendor-managed use product.

Issue the `i4lct -h` command and see the notes at the end of the displayed text for information about the valid values of this parameter.

-s *start_date*
Specifies the start date of the password. The date format is *mm/dd/yyyy*. If the password type is license, this value indicates the effective start date of the licenses; if the password type is compound, this value indicates the start date at which you can create license passwords that are derived from the compound password.

To provide concurrency of licensing across the international date line, you can specify a date value of current date - 1 day. If you specify a date earlier than that, `i4lct` issues an error message and does not create a license certificate file.

The maximum start date you can specify is 4095 days from the current date.

This parameter is optional. If it is omitted, the start date of the password defaults to the current date.

-S *derived_start_date*
Valid only with compound or compound nodelocked passwords (**-w** set to **c** or **cn**). The date format is *mm/dd/yyyy*. This is the derived license start date, the date before which no license password derived from the compound password is valid.

To provide concurrency of licensing across the international date line, you can specify a date value of *current date - 1 day*. If you specify a date earlier than that, `i4lct` issues an error message and does not create a license certificate file.

This parameter is optional. If it is omitted, the derived start date of the password defaults to the current date.

-t *target_type*

The target type of the license server on which the licenses are to be installed.

Valid values are **any** (valid only for customer-managed licenses)**, aix, dg[ux], do[main], h[pux] i[ntergraph], m[sdos], ne[xt], no[vell], os2, os2mac, sco, sgi, sun, svr4, u[ltrix], v[ms], apollo, open, sun, vax, hposf, clipper, osfl, win32, win32mac, hiux, nec**.

The **win32mac** and **os2mac** parameters specify that the target ID to be used is based on the network adapter. The **win32** and **os2** parameters specify a software-based target ID.

For customer-managed licenses, the value of **-t** must be set to **any**.

-T *target_id*
: Specifies the target ID of the license server where the license password is to be installed. The target ID can be either the old style (32-bit) or the new style (64-bit).

If the target type, **-t**, is set to **any (valid only for customer-managed licenses)**, the target ID, **-T**, is set to **any** by default.

-u
: The upgrade flag for a custom configuration license. This flag indicates whether the customer's initial configuration and password have been modified. The replacement password is used thereafter. For concurrent network licenses, the initial password is deleted, leaving only the replacement password available. For simple nodelocked licenses, the initial password remains in the file and must not be deleted, though only the most recent replacement password is used.

-U
: Display the command line usage information.

-v *vendor_name*
: Specifies the vendor name. It can be up to 31 characters long. All vendor name specifications must be enclosed within double quotation marks ("***vendorname***"). Vendor name specifications are case-sensitive.

If you are generating test passwords, specify the value **test**.

-V
: Display the `i4lct` version string.

-w *password_type*
: The type of password to be created; supply one of the following keywords:

l Simple password

c Compound password

cn Compound nodelocked password; valid only in conjunction with the try-and-buy attribute (**-A** set to **t**)

-x *extended_target_type*
: The type of target for an extended target ID. In LUM Runtime Version 4.5.x and 4.6, the only valid value for **-x** is **cluster**.

-X *extended_target_id*
: The ID of the extended target on which the password is to be installed. In LUM Runtime Version 4.5.x and 4.6, **-X** is the ID of a cluster.

Examples

The following command creates an enrollment certificate that contains the password to test a vendor-managed use product. It represents 100 concurrent access licenses, with one year of duration, and with multiuse rules specified.

```
i4lct -i test -k test -v "test"
-N "Example Licensed Product" -p 1 -r 1.0 -R v
-w l -l c -t any -a "Example Product" -s 01/01/1998 -d 365 -n 100 -m ug
```

The following command creates the enrollment certificate that contains the password to test a customer-managed use product with a per-seat license.

```
i4lct  -i test -k test
-v test -N "Example Licensed Product6"
-p 6 -r 1.1 -w l -l pt -a "Example Product Core Package"
-s 1/1/1998 -d 365  -t aix -T any -R c
```

The following command creates an enrollment certificate that contains an initial custom configuration key for a nodelocked license:

```
i4lct -i 5242378dbf8d.02.c0.09.c8.93.00.00.00 -k 53989 -l n -p 50
-N "Mechanical Design" -d 730 -t aix -T 152c234 -v "Mechanical Systems"
-w l -r 1.2 -C 85AB2215691 -a "MD2"
```

The following command creates an enrollment certificate that contains a replacement custom configuration key for the nodelocked license in the preceding example. In this example, the duration of the license is extended from the initial 730 days to 5000 days. The other values remain unchanged.

```
i4lct -i 5242378dbf8d.02.c0.09.c8.93.00.00.00 -k 53989 -l n -p 50
-N "Mechanical Design" -d 5000 -t aix -T 152c234 -v "Mechanical Systems"
-w l -r 1.2 -C 85AB2215691 -a "MD2" -u
```

The following commands create a bundle that contains two products. Three licenses are created (two for the products and one for the master license).

To create the license for the first product (20 concurrent network licenses):

```
i4lct -i 92cd55ef995d.02.92.54.b1.74.00.00.00 -k 53989 -d 100 -w l
-A b -l c -n 20 -N "Mechanical Design Product1" -p 54 -r 2.0
-C BNDL016D -t any -v "Mechanical Systems"
```

To create the license for the second product (nodelocked license):

```
i4lct -i 92cd55ef995d.02.92.54.b1.74.00.00.00 -k 53989 -d 100 -w l
-A b -l n -N "Mechanical Design Product2" -p 56 -r 2.5
-C BNDL016D -t any -v "Mechanical Systems"
```

To create the master license for the bundle:

```
i4lct -i 92cd55ef995d.02.92.54.b1.74.00.00.00 -k 53989 -d 100 -w l
-A b -F 102 -l c -n 2 -N "Mechanical Design Bundle" -p 52 -r 1.0
-C BNDL016D -t any -v "Mechanical Systems"
```

Chapter 7. Tips and Hints

This chapter provides information intended to help you better manage your licensing environment.

Managing the Time Zone

To display the correct time in administration tool panels under Windows 95 or Windows 98, set the environment variable TZ. To do this, add the following line to the autoexec.bat file:

```
SET TZ='value'
```

where *value* is the same as that in step 3 in the following procedure:

1. Click **Settings** --> **Control Panel** --> **System** --> **Environment**.
2. In the **Variable** entry field, enter **TZ**.
3. In the **Value** entry field, enter:

   ```
   XXXsnnYYY
   ```

   where:

   XXX **and** *YYY*
   Characters you choose to remind you of the setting

   s A sign: + or -

   nn The difference in hours from Greenwich mean time (GMT). A positive difference means the Greenwich is west of your location; a negative difference means that Greenwich is east of your location. For example, you might specify Italian standard time (one hour east of Greenwich mean time, or GMT+1) as:

   ```
   IST-1GMT
   ```

Remember to adjust this time to account for standard time and daylight savings time in different countries.

Using the Built-In Backup and Recovery Procedure

Because the breakdown of license servers may have a potentially severe impact on production, it is important to be prepared in case definitions and database files are corrupted.

The minimum backup activity the administrator should do is to keep the enrollment certificate files (or e-mail or hard copy equivalents) received from the license provider in a secure place.

LUM Runtime implements a backup procedure of all databases on license server workstations.

Causes for Corrupted Definition or Database Files

Since LUM Runtime uses the database files dynamically, any disk-related problems such as the following may cause the database files to become corrupted:

- Hardware failures (media surface errors)
- File-system problems (for example, file system full)
- Synchronization errors during writing of data (that is, loss of electrical power)

When a LUM Runtime database is corrupted, after the database has been recovered, try to find out the real cause of the problem.

The contents of the definition and database files used by LUM Runtime are changed only by defined administrative commands and tools.

Automatic Backup Procedure

LUM Runtime does an automatic periodic backup on license servers by copying all files and databases in the directory:

```
\ifor\ls\conf\backup
```

You can choose to get the backup on any other device by changing the **BackupPath** parameter in the configuration file (i4ls.ini). You can set the automatic backup to occur daily, at a certain time (the default); weekly, on a certain day; or at every change on the license database, according to the **BackupMode** and **BackupParm** parameters specified in the configuration file. You can also disable the automatic backup procedure by setting the **BackupMode** parameter to **none**.

For detailed information on the configuration file (i4ls.ini) see “Appendix A. LUM Runtime Configuration File” on page 209.

Make sure that the **BackupMode** and **BackupParm** parameters have the same value on all servers in the licensing environment.

The objects listed in:

```
\ifor\ls\conf\db_back.cmd
```

are backed up if found.

Recovery Procedure

To recover the files and databases saved with the automatic procedure described in “Automatic Backup Procedure” on page 186:

1. Stop the LUM Runtime services either by selecting **Stop** from the Service pull-down menu of the service manager tool, or by issuing the following command:

   ```
   i4cfg -stop
   ```

2. Issue the following command:

   ```
   i4ls -r
   ```

This replaces the current objects with those saved with the backup procedure.

Important: If a file is damaged, run this command according to the following rules:

- If the **BackupMode** in the configuration file (i4ls.ini) is set to **changes**, run the recovery command only on the server where corruption occurred.
- If the **BackupMode** is set to **daily** or **weekly**, first check that the backup copies have the same date on all the servers of your licensing environment, then run the backup command on all the servers.

Manual Backup

On Windows NT, you can backup manually by running:

```
\ifor\ls\conf\db_back.cmd
```

On a workstation configured only as a network license client, only manual backup is available.

The command copies the files and databases to a backup diskette (the a: drive).

Manual Recovery

To start the recovery procedure for damaged files on Windows NT, enter on the failing workstation:

```
\ifor\ls\conf\db_recov.cmd
```

This command restores, from the a: drive, the files and databases that were saved by the `db_back.cmd` command.

Managing the Reports Log Files

When you ask for a report, the basic license tool reads the current log files:

```
\ifor\ls\conf\logdbnn_ (network license server)
\ifor\ls\conf\crlognn_ (central registry license server)
\ifor\ls\conf\llmlgnn_ (nodelocked license server)
```

The names of the current log files end with an underscore.

The files `logdb`*nn*, `crlog`*nn*, and `llmlg`*nn* contain all the collected LUM Runtime events. You can specify which events are to be collected when you configure each license server. See the examples in “Scenario 1: Configuring a Standalone Nodelocked License Server” on page 75, and “Scenario 3: Configuring a Network License Server” on page 80.

nn can assume values from 00 to 99. When a file is full, a new one is started. You determine the maximum value *nn* can assume and the maximum size of each file by setting the **NumberOfLogFile** and **MaxLogFileSize** parameters in the configuration file, `i4ls.ini`.

When the maximum value for *nn* is reached, LUM Runtime wraps to 00. The filled log files are retained so that you can archive them if you wish before the numbering wraps. For details see “Appendix A. LUM Runtime Configuration File” on page 209.

The numbering of log files starts from 00. Suppose you have the following files on the workstation:

```
logdb00
logdb01_
logdb02
```

The second file is the current, the first is the previous, and the third is the oldest.

If the current files, marked with the underscore, get too big, do not delete them. You can decrease the size of the current files with the following command:

```
i4blt -x delete_date -n server_name
```

where:

delete_date Specifies an end date for the delete operation. All log entries recorded up to the delete date are removed from the log files. If you do not specify a date all the entries are deleted.

server_name Specifies the license server where you want to delete the entries of `logdb`*nn*`_`, `crlog`*nn*`_`, and `llmlg`*nn*`_` if they exist on the server.

Tuning the Time-Out Interval to Detect a Nonresponding Server

You can increase or decrease the time-out interval, using a scaling factor. You can modify the time-out interval for connections between client and server, by changing on the TIMEOUT_FACTOR environment variable on the client side of the connection.

The default value of TIMEOUT_FACTOR is 1000; values greater than 1000 increase the time-out interval, while values below 1000 decrease it.

Managing a Custom Configuration

This section offers advice about custom configurations and their licenses.

Before Requesting a License Upgrade

Before you request an upgrade to your current custom configuration license, double-check the serial number.

Deleting Products or Reducing Numbers

When you upgrade a custom configuration, you can add products and increase the number of seats; however, you can neither delete products nor reduce the number of seats.

Deleting Keys

The initial key is always required. Do not delete it from either the network license server or the nodelock file. You can, however, delete intermediate upgrade keys from a nodelock file. (These intermediate keys are deleted automatically on network license servers.)

Chapter 8. Troubleshooting

This chapter provides suggestions for improving performance, problem determination, and debugging when using products managed with LUM Runtime. This chapter assumes you have read the preceding chapters in this book.

Checking License Details

Before you proceed, be sure you know the following details about the product that is not starting properly. Check the product enrollment certificate file for all these details.

- Product name (ProductName tag)
- Product version (ProductVersion tag)
- Vendor name (VendorName tag)
- Target type (TargetType tag)
- Whether the product implements customer-managed or vendor-managed use control (RegistrationLevel tag; 1=customer-managed, 3=vendor-managed)
- Whether the product is enabled for a custom configuration policy (SerialNumber tag)
- Whether the password is simple or compound (LicenseStyle tag=compound, or LicenseStyle tag=license type if the password is simple)
- License type (LicenseStyle tag if the password is simple; DerivedLicenseStyle tag if the password is compound)
- Whether the product is enabled for the hard stop / soft stop policy (SoftStop tag)
- When the license becomes valid and when it expires (LicenseStartDate and LicenseEndDate tags)
- Whether the password specifies a license type or a policy introduced in Version 4 (PasswordVersion tag: 8=new in Version 4.6, 7=new in Version 4.5.5, 6=new in Version 4.5.0 5=new in Version 4.0, 4=not new in Version 4)

This is an example of an enrollment certificate file:

```
i4blt -a
-v "'IBM Corporation' 8499f53d66dd.8d.01.51.32.4c.00.00.00 673mr5dfvm2w6"
-p "'DataWare' '2.1a' fgm63s65v4edb9y5fjujtbnce2dmhjhqgajuupibaaaaa "
-S 00102293AE8

[LicenseCertificate]
CheckSum=ED729ADF511FE95361CF28FD87B943A1
```

```
TimeStamp=961160135
PasswordVersion=8
VendorName=IBM Corporation
VendorPassword=673mr5dfvm2w6
VendorID=8499f53d66dd.8d.01.51.32.4c.00.00.00
ProductName=DataWare
ProductID=2222
ProductVersion=2.1a
ProductPassword=fgm63s65v4edb9y5fjujtbnce2dmhjhqgajuupibaaaaa
ProductAnnotation=
LicenseStyle=concurrent
LicenseStartDate=10/01/2000
LicenseDuration=13606
LicenseEndDate=12/31/2037
LicenseCount=24
MultiUseRules=Node/Accessor
RegistrationLevel=3
TryAndBuy=No
SoftStop=No
Bundle=No
Registration=No
Uniqueness=No
Capturable=No
TargetType=ANY
TargetTypeName=Open Target
TargetID=ANY
ExtendedTargetType=
ExtendedTargetID=
SerialNumber=00102293AE8
Upgrade=No
InstallProgram=
CapacityType=Processors configured
DerivedLicenseStyle=
DerivedLicenseStartDate=
DerivedLicenseEndDate=
DerivedLicenseAggregateDuration=
```

The i4blt command at the top of the certificate file is the command that could be used to enroll the password. In the actual enrollment certificate file it would appear on one line; here it is shown on four lines because of space constraints.

Migrating from Windows 9x to Windows 2000 on a LUM Runtime System

If, after installing LUM Runtime on a Windows 95 or Windows 98 workstation, you upgrade your operating system to Windows 2000, reinstall LUM Runtime, to register its services.

Migrating from Windows NT 4.0 to Windows 2000 on a LUM Runtime System

If you install LUM Runtime 4.6 on a Windows 2000 workstation on which LUM Runtime is not currently installed, the .ini file is installed in the directory `x:\Documents and Settings\All Users\Application Data` directory, where *x* is the drive letter of the logical drive on which Windows is installed.

If you install LUM Runtime 4.6 on Windows 2000 over a previous installation of LUM Runtime 4.5.8 or 4.5.9, the .ini file is left in the `x:\ifor\ls\conf` directory.

If the system has only one network adapter, the registry entry I4DRIVER is automatically updated to the Windows 2000 format.

If the system has more than one network adapter, update the registry manually, as follows:

1. Run `i4target -z`.

   This displays information about all the devices in the workstation.
2. Run `i4target -d` *devicename*

   where *devicename* identifies the device driver to be used to retrieve the target ID of the workstation.

For more information about the i4target command, see "i4target - Target View Tool" on page 162.

Troubleshooting Licenses (All Types)

If a user tries to start a license-enabled product and it does not start, some of the first things to check are:

- First, check the product documentation.
- Check to be sure the license for the application you are running is installed, and, if not, install it. See "Enrolling the Product" on page 97 for information on how to install a license.
- Check that the license you have installed is the correct license for the version of the software you are trying to run.
- Check that the date and time on the workstation are set correctly. Each license has a start date and an end date built in. If the date or time is set incorrectly on the workstation where you are trying to run the product or on a license server, the license may not be recognized as active.
- Check that the time zone and daylight saving time settings are correct.
- Check that the start date of the enrolled license is not later than the current date, and that the license has not expired.

Troubleshooting Nodelocked Licenses

If a workstation with a nodelocked license does not allow an end user to use a license-enabled product, try the suggestions in this section.

If the product uses non-runtime-based enabling:

- Check that the enrollment certificate file is in the path specified by the vendor of the product and that its permissions are set so that all users can read it.
- Check that the license is correctly installed in the nodelock file specified by the vendor and that its permissions are set so that all users can read it.

  The default location of the nodelock file is as follows:

  ```
  c:\'COMMON_APPDATA'\IBM\LUM\nodelock
  ```

  where:

  c The drive identifier

  COMMON_APPDATA
  The initial part of the path that is returned by the SHGetFolderPath API

  Examples of typical paths for the nodelock file across the Windows operating systems are as follows:

  Windows 95:
  `c:\windows\All User\Application Data\IBM\LUM\nodelock`

  Windows 98:
  `c:\windows\Profiles\All User\Application Data\IBM\LUM\nodelock`

  Windows NT:
  `c:\winnt\Profiles\All User\Application Data\IBM\LUM\nodelock`

  Windows 2000:
  `c:\Documents and Settings\All User\Application Data\IBM\LUM\nodelock`

  If the file is not in the default directory, use the SHGetFolderPath API to retrieve it.

If the product uses runtime-based enabling:

- Check that the nodelocked license server (i4llmd) is up and running (see “Starting Required Processes” on page 200).
- If the request waits for some time and then fails with error message:

  ```
  Inter process communication failure: check log file i4ipc.out
  ```

it may be that the maximum wait time for an application to receive a response from the nodelocked license server via Interprocess Communications, as specified in the configuration file, is too short. Edit the configuration file and increase the value of the **ReadTimeout** parameter, for example to 20:

```
ReadTimeout=20
```

Running Nodelocked Applications on Windows 2000 Terminal Services

To run nodelocked applications and monitor the related licenses by means of the basic license tool, on a Windows 2000 Terminal Services system, follow these steps:

1. Select **Start** --> **Administrative Tools** --> **Terminal Services Configuration**.
2. Check the values of the settings shown in Table 14.

Table 14. Windows 2000 Terminal Services Configuration for Nodelocked Applications

Setting	Value
Terminal Server Mode	Application Server
Permission Compatibility	Terminal Server 4.0 User

3. Double-click any incorrect setting, to open it for editing.
4. Change its value. Ignore any warning message about LUM Runtime working.
5. Save and close the file.

Running the Basic License Tool Graphical User Interface on Windows 2000 Terminal Services

To run the basic license tool graphical user interface on a Windows 2000 Terminal Services system, follow these steps:

1. Select **Start** --> **Administrative Tools** --> **Terminal Services Configuration**.
2. Check the values of the settings shown in Table 15.

Table 15. Windows 2000 Terminal Services Configuration Settings for the Basic License Tool GUI

Setting	Value
Terminal Server Mode	Application Server
Permission Compatibility	Terminal Server 4.0 User

3. Double-click any incorrect setting, to open it for editing.
4. Change its value. Ignore any warning message about LUM Runtime working.

5. Save and close the file.
6. Add the system host name to the License Use Management direct binding list.

Troubleshooting Network Licenses (All Types)

If a user tries to start a product with a network license and the product does not start, try the following steps. These suggestions apply to concurrent, use-once, reservable/reserved, and per-seat licenses.

- Use the `i4tv` command from the client workstation to verify the connection to the license server where you have the licenses installed.
  - If `i4tv` shows no active servers, check that the network license server is running on the server workstations where you have the licenses installed.
  - If `i4tv` does display active servers, check that they include a workstation where licenses for the product are installed. Use `i4blt -lp` to display the licenses installed on each server. You may need to reconfigure the client to connect to the proper servers.
  - In direct binding, verify that the client is configured to connect to the correct servers.
- If you get the error message:

  ```
  Time disparity is too large
  ```

  check that the date and time on the servers and client are synchronized. If server and client are in different time zones, be sure that time zone and daylight saving time have been set correctly.
- If an enabled application requests more than one license to run, be sure the requested number of licenses is available on one server.

  LUM Runtime does not combine licenses installed either on different servers or on the same server but with multiple enrollment actions, to satisfy the same request.

  Similarly, if you received the licenses in a compound password, check that you have distributed, in one single distribution, on one server, at least the number of licenses requested. LUM Runtime does not combine licenses distributed either on different servers, or on the same server but with multiple distribution actions, to satisfy the same request.

  For the same reason, if the product is enabled for soft stop, you may see soft stop licenses in use even if there are still some available licenses.

Troubleshooting Reservable and Reserved Licenses

Reservable licenses are enrolled on a network license server. When reserved they are moved to the central registry, and when granted they are moved to the nodelocked license server on the client workstation. If a license has been reserved for a user but, when that user tries to use the product, it does not start:

- Check that the central registry license server is up and running (see “Starting Required Processes” on page 200).

- Check that the client workstation can reach the central registry.
- Check that the nodelocked license server is up and running at the client (see “Starting Required Processes” on page 200).
- Check that the date and time set on the central registry are the same as that set on the network license client. It is possible that, according to the date and time set on the central registry, the license is not yet valid or has expired.
- Double-check the name of the user, group, or node for which licenses are reserved. Be careful with leading and trailing blanks. Note that the domain is part of the node specification.

Troubleshooting Per-Server and Per-Seat Licenses

- If an application with per-server licenses fails to start, make sure the nodelocked license server is running on the workstation where the application server runs (see “Starting Required Processes” on page 200).
- If an application with per-seat licenses fails to start:
  - Make sure the central registry license server is up and running, and that the nodelocked license server is running on the workstation where the application server runs (see “Starting Required Processes” on page 200).
  - Make sure the per-seat license is enrolled and that per-seat licensing has been enabled (see “Scenario 8: Switching from Per-Server to Per-Seat Licenses” on page 108).
  - Make sure the workstation where the application server runs can reach the central registry.

Troubleshooting Licenses of Customer-Managed Use Products

If you are unable to enroll, update, or distribute licenses for a customer-managed product, a customer-managed use product fails to start, if soft stop does not work, or the high-water mark does not work:

- Make sure that the central registry license server is up and running, and that you have defined only one central registry license server in the direct binding servers list (see “Starting Required Processes” on page 200).
- Make sure the workstation where you are working can reach the central registry.
- If you received the licenses in a compound password, make sure you have distributed the licenses. See “Distributing the Licenses” on page 100 for information on how to distribute licenses.
- If soft stop does not work, make sure the soft stop policy is enabled.

Troubleshooting Licenses of Vendor-Managed Use Products

If enrollment of a vendor-managed use product fails, check that the target ID and the target type in the license match the target ID and target type of the workstation where the license is installed. To get the target ID of the workstation, run the target view tool (see “i4target - Target View Tool” on page 162) on that workstation.

If there is a mismatch, it is possible that:

- The vendor put the wrong target ID or target type into the license.
- You are trying to use the license on the wrong workstation.
- The target ID of your target workstation has changed (see “Troubleshooting the Hardware” on page 204).

Troubleshooting License Databases

If the basic license tool fails to display enrolled licenses, try to remove invalid records from the license databases. To do so, use the `-clean` option of the license server, as follows:

1. Stop the license services by entering the command:

   ```
   i4cfg -stop
   ```

2. Backup the following files:

   ```
   \ifor\ls\conf\licdb.dat
   \ifor\ls\conf\licdb.idx
   ```

3. Issue the command:

   ```
   \ifor\ls\bin/i4lmd -clean –v
   ```

4. When you see on the screen the string `Listening` ...., press **Ctrl+C** to stop the command.
5. Restart the license services by entering the command:

   ```
   i4cfg -start
   ```

Troubleshooting Performance Problems

Read this section for assistance with optimizing performance.

Basic License Tool Performance

In a network with many users, or when you are creating large reports, if performance is consistently slow when you use the basic license tool graphical user interface, consider switching to the command line interface.

Performance in a Direct Binding Environment

In a direct binding environment, careful configuration can help you to optimize performance. See the performance notes under “Planning Direct Binding” on page 39.

Troubleshooting Heavy Server Workloads

When a License Use Management server is stressed by a heavy workload, performance could deteriorate to the point that the server can no longer manage licenses. To avoid this situation, spread the workload over two or more servers.

Troubleshooting LUM Runtime Processes

This section covers problems that could arise if LUM Runtime processes are not started, or go down.

Starting Required Processes

When a license-enabled product fails to start, the problem may be that a required LUM Runtime process is not running.

To get a list of the LUM Runtime processes that are running on a workstation, use the `i4cfg -list` command or double-click the **Service Manager Tool** icon in the License Use Runtime folder.

For an overview of which license servers are required for each license type, see Table 16.

Table 16. LUM Runtime Processes

Name	Process
Nodelocked License Server	i4llmd
Network License Server	i4lmd
Central Registry License Server	i4gdb
NetBIOS socket server	i4bnssvr

To start the processes, select **Start** from the **Service** pull-down menu.

If the process fails to start when you issue the command, check the error messages in the i4ls.log, i4lmd.log, l4llmd.log, and l4gdb.log files in the `x:\ifor\ls\conf` directory, where *x* is the installation directory.

Restart and Recovery

- If a network license server, a nodelocked license server, or the central registry license server goes down, a record of users who currently have licenses is kept on disk. When the server is restarted, the record is reinstated and the licenses are still assigned to those users.

  If you want a cold start (that is, if you want the server to restart as if it had granted no licenses before going down), use the **-c** parameter on the command used to restart the server (`i4lmd`, `i4llmd`, or `i4gdb`, all described in “Chapter 6. Command Line Interface” on page 125). To change the default permanently to cold start, edit this parameter of the i4ls.ini file:

```
ColdStart=yes
```

and then restart services (`i4cfg -start`).

Cold start is not possible for reserved and per-seat licenses.

- If the client workstation goes down or the network fails, the licenses it was using become *stale* (after a check period expires, if the application is enabled using concurrent access or reservable licenses, and the application is programmed to check in with the server after a specified check period). In this case, the licenses are available to be granted to other clients. Note that those licenses will still be displayed as in use until you perform the **Clean up stale licenses** function or until a license is newly requested and none is available, in which case the server does its own cleanup of stale licenses.

  The client behavior depends on the software product that is in use.

Troubleshooting Custom Configuration Licenses

If, for a custom configuration, you are unable to install a network concurrent license or nodelocked license from the certificate file:

- Check the serial number.
- Check whether another license with the same serial number is already installed. For a concurrent license, use `i4blt`. For a nodelocked license, use the nodelock file.
  - If you are installing the initial key, no other key can already be installed.
  - If you are installing a replacement key, another key must already be installed.

Troubleshooting Network Connections

If connections to license servers seem not to be working properly, use the test verification tool (`i4tv`) to verify that the license servers are up and running, or use the `i4blt -ln` command to get a list of active servers (network license servers and the central registry). For more information about these commands, see “Chapter 6. Command Line Interface” on page 125.

Troubleshooting Direct Binding

If servers and clients are not communicating correctly in a direct binding environment (the `i4tv` command reports `No servers found`), check that exactly the same server names and direct binding port numbers were configured for each client, each network license server, and the central registry license server.

For example, if you use TCP/IP and have a server called **louise** running the network license server and the central registry license server, and clients connected to **louise**, the direct binding configuration for each workstation, including **louise** itself, must contain the entries:

```
ip:louise[10999]
ip:louise[1515]
```

The port numbers must match the values of the **ipPort** and **ipGDBPort** parameters in the configuration file of **louise**. **louise** and its clients all must have TCP/IP installed.

Troubleshooting Communication Protocols

If you have communication problems, first verify that the protocol you use is installed on the workstation and configured properly, as described in "Protocol Configuration Requirements" on page 53.

NetBIOS

The NetBIOS protocol is not supported on Windows 95 or Windows 98 workstations.

If you use NetBIOS, the following brief checklist should help you to make sure your NetBIOS system is working correctly:

1. Make sure you entered the workstation names correctly when you configured the direct binding servers list at license servers and clients of your environment. Note that the workstation name and the server name fields of the configuration tool are case-sensitive:
   - On Windows 2000 and Windows NT, the workstation names are the computer names. They must be entered in uppercase.
   - On OS/2, the workstation names must be entered in lowercase.
2.

   Check that the NetBIOS socket server is up and running on every workstation that uses NetBIOS to communicate with an OS/2 workstation.
3. Check that the NetBIOS socket server is up and running on every Windows NT workstation that your system needs to communicate with.

   For instructions, refer to "Running the NetBIOS Socket Server" on page 46.
4. Check whether normal NetBIOS communications are working between the nodes you want to be connected.
5. The **Port** and **CrLS Port** values must be greater than 31 and less than 255.
6. Check that the LAN adapter number of the Network Route starting with Nbf is set to 0 (see "Protocol Configuration Requirements" on page 53).

TCP/IP

If you use TCP/IP, the following is a brief checklist to help you make sure your TCP/IP system is working correctly:

1. Check that the TCP/IP system is up and running.
2. Check whether IP addresses or network interfaces have been changed.
3. Check whether normal TCP/IP communications are working between the nodes you want to be connected (for example, using `ping` or FTP).
4. Make sure your routing setup definition is valid.

   The netstat command shows the local definition. To see the hubs, use the `route print` command.
5. Is name resolution working?

   Name resolution is very often the reason for long startup times or many sorts of problems in large networks. Use the DNS (Domain Name System), and spend some time developing a good layout.
6. Is the MTU (Maximum Transmission Unit - Internet protocols) size equal on all hosts?
7. Is the token ring speed equal on all hosts?
8. NCS and LUM Runtime are based on universal datagram protocol (UDP). In a very highly loaded network, UDP connections may receive time-outs before data is delivered. This is normal behavior; you need to reduce the total network load.
9. If your LUM Runtime server is on a workstation configured with the loopback facility of TCP/IP, when you use the `i4blt -a` command to enroll a product on the server, you must specify:

   ```
   -n ip:local host
   ```

   in the `i4blt` command.

IPX on Windows 98

The IPX communication protocol does not work on the initial release of Windows 98. Use TCP/IP instead.

Troubleshooting the Hardware

Following is a brief checklist to help you make sure your hardware is working correctly.

1. If you get the error message:

   ```
   Invalid target ID
   ```

   check that your target ID has not changed.
2. Check that the cables are still where they should be.
3. Check whether you have reached the Ethernet length limitations on your LAN.
4. Check whether a security feature has been enabled on a router.

   Some routers allow enabling of security features. It is possible to block certain TCP/IP ports.

   In direct binding, the ports are predefined in the configuration file.
5. Check whether any adapters or network definitions have been changed. Because the NCS definition and database files are linked to network addresses, changes may lead to connection errors.

Collecting Error Log Data

To help IBM help you in problem determination, gather additional information to send to your IBM representative when you request support. LUM Runtime processes and tools can be run in traced mode as explained in the following sections.

Running Processes in Traced Mode

To run LUM Runtime processes in traced mode, collect trace data, and view the trace messages, follow these steps:

To run the processes in traced mode:

1. Stop all active processes, either by selecting **Stop** from the **Service** pull-down of the **Service Manager Tool** window, or by issuing the following command:

   ```
   \ifor\win\bin\i4cfg -stop
   ```

   ```
   \ifor\ls\bin\i4cfg -stop
   ```
2. Start traced mode either by selecting **Trace Level**, then **To File** from the **Options** pull-down menu of the **service manager tool** window, or by editing the i4ls.ini file and setting the following tags to **yes**:

   ```
   DebugProc=yes
   DebugNCS=yes
   DebugToFile=yes
   ```
3. Issue the following commands:

```
set IFOR_DEBUG=Y
set I4_IPC_DEBUG=4
```

4. From the same window, restart the processes by issuing the following command:

   ```
   \ifor\win\bin\i4cfg -start
   ```

To collect the trace data:

1. Produce the error condition again.
2. Stop all active processes again. The output trace information is collected in the files with extension trc.

To view the trace data using the GUI:

1. In the **Service Manager Tool** window select **Stop** from the **Service** pull-down menu.
2. Select the line corresponding to a specific process in the list.
3. Select **View trace** from the **Selected** pull-down menu. The View trace window is displayed. It shows the trace messages.

Running Enabled Applications in Traced Mode

To run enabled applications in traced mode set the environmental variables I4LIB_VERB and IFOR_LT_DEBUG as follows:

```
set I4LIB_VERB=Yes
set IFOR_LT_DEBUG=Yes
set RPC_DEBUG=Yes
```

Then start the enabled application from the same window.

Trace messages will be displayed in the same window where you have set the variable and from which you run the application.

Running Tools in Traced Mode

To run tools in traced mode, first set the following environmental variables:

```
set I4_DEBUG_FL=Yes
set IFOR_LT_DEBUG=Yes
set I4_IPC_DEBUG=4
set RPC_DEBUG=Yes
```

Then start the tools from the same window.

Collecting Other Data

Other information concerning LUM Runtime servers is automatically collected by the license server processes (i4llmd, i4lmd, and i4gdb). This data is stored in the following files:

```
\ifor\ls\conf\i4ls.log
\ifor\ls\conf\i4lmd.log
\ifor\ls\conf\i4llmd.log
\ifor\ls\conf\i4gdb.log
```

Most of the messages you find in these files and the related return codes are not documented.

Also provide IBM with:

- The default location of the configuration file, i4ls.ini, is as follows:

  `c:\'COMMON_APPDATA'\IBM\LUM\i4ls.ini`

  where:

  c The drive identifier

  COMMON_APPDATA
  The initial part of the path that is returned by the SHGetFolderPath API

  Examples of typical paths for the nodelock file across the Windows operating systems are as follows:

  Windows 95:
  `c:\windows\All User\Application Data\IBM\LUM\i4ls.ini`

  Windows 98:
  `c:\windows\Profiles\All User\Application Data\IBM\LUM\i4ls.ini`

  Windows NT:
  `c:\winnt\Profiles\All User\Application Data\IBM\LUM\i4ls.ini`

  Windows 2000:
  `c:\Documents and Settings\All User\Application Data\IBM\LUM\i4ls.ini`

  If the file is not in the default directory, use the SHGetFolderPath API to retrieve it.

- `\ifor\ls\conf\userfile` (the user file)
- LUM Runtime databases
  - License databases

    ```
    \ifor\ls\conf\licdb.dat
    \ifor\ls\conf\licdb.idx
    \ifor\ls\conf\llmdb.dat
    \ifor\ls\conf\llmdb.idx
    \ifor\ls\conf\crpdb.dat
    \ifor\ls\conf\crpdb.idx
    ```

  - Log Databases

```
\ifor\ls\conf\logdbnn_.dat
\ifor\ls\conf\logdbnn_.idx
\ifor\ls\conf\llmlgnn_.dat
\ifor\ls\conf\llmlgnn_.idx
\ifor\ls\conf\crlognn_.dat
\ifor\ls\conf\crlognn_.idx
```

Troubleshooting LUM Java Client Support

If you are having trouble with LUM Java Client Support under Windows NT:

- Check the WebSphere servlet_log and error_log files in:

  ```
  WEBSPHERE_root\logs\servlet\servletservice
  ```

- To enable native DLL plug-in logging in the `WEBSPHERE_root\logs\native.log` file, edit the file:

  ```
  WEBSPHERE_ROOT\properties\server\servlet\servletservice\jvm.properties
  ```

  and change **ncf.native.logison** from **false** to **true**.

- To enable Java virtual workstation logging in the `WEBSPHERE_root\logs\ncf.log` file, change both **ncf.jvm.stdoutlog.enabled** and **ncf.jvm.stdoutlog.file** from **false** to **true**.

Web Server Fails

If the Web server fails because it lacks a permission, do one of the following tasks:

- **For Lotus Domino Go**:
  - In the file:

    ```
    /etc/httpd.conf
    ```

    set:

    ```
    UserID=root
    GroupID=system
    ```

  - In the file:

    ```
    opt/IBMWebAS/properties/server/servlet/server.properties
    ```

    set:

    ```
    server.user=root
    ```

- **For Netscape FastTrack and Netscape Enterprise:**

  During installation of these products set:

  ```
  UserID=root
  GroupID=system
  ```

- **For Lotus Domino Go, Netscape FastTrack, or Netscape Enterprise**:
  1. Create a new user name and a new group name with the required permission.

2. Set the UserID variable to the value of the new user name and the GroupID variable to the value of the new group name.

Java Program Cannot Read the User Name

If a Java applet, loaded on Netscape Communicator, cannot read the user name, install the latest version of Netscape Communicator.

Incomplete View of an Applet

If, when you run LicenseTest as an applet, you cannot see the whole applet window in your Web browser, change the window's width or height, or both. These are specified in the LicenseTest.htm file.

Troubleshooting Java Client Support Uninstallation

If you uninstall LUM Runtime, some environment variables may remain set. You can check this by opening a DOS shell and entering:

```
set i4
```

If the i4 environment variable remains set, the variable and its value are reported.

To install Java Client Support and to install silently the LUM Runtime Communication feature without encountering problems, make sure that no LUM Runtime environment variables are set. To remove environment variables, do one of the following:

- After uninstalling LUM Runtime, shut down and restart the workstation.
- Use regedit to remove the LUM Runtime environment variable from the registry.
- Remove the LUM Runtime environment variable from the Environment tab in the System Properties window.

Microsoft Web Server Cannot Find i4netj.dll

If, after installing LUM Runtime Java Client Support, Microsoft Web Server issues the message:

```
Cannot find the i4netj.dll in path path
```

reboot the system.

Appendix A. LUM Runtime Configuration File

This appendix describes the LUM Runtime i4ls.ini configuration file.

The default location of the configuration file, i4ls.ini, is as follows: `c:\'COMMON_APPDATA'\IBM\LUM\i4ls.ini`

where:

c The drive identifier

COMMON_APPDATA
The initial part of the path that is returned by the SHGetFolderPath API

Examples of typical paths for the nodelock file across the Windows operating systems are as follows:

Windows 95:
c:\windows\All User\Application Data\IBM\LUM\i4ls.ini

Windows 98:
c:\windows\Profiles\All User\Application Data\IBM\LUM\i4ls.ini

Windows NT:
c:\winnt\Profiles\All User\Application Data\IBM\LUM\i4ls.ini

Windows 2000:
c:\Documents and Settings\All User\Application Data\IBM\LUM\i4ls.ini

If the file is not in the default directory, use the SHGetFolderPath API to retrieve it.

You should normally use the configuration tool to configure LUM Runtime. In case you have no access to the configuration tool or you want to change just a few parameters of your configuration, the information in this appendix will enable you to modify the parameters by editing the file. Also, some parameters (designated in this appendix) can be changed only by editing the configuration file.

If a parameter has a default value, it is shown with the parameter name (for example, **BackupMode=daily**).

`[iFOR/LS Machine-Configuration]`

ConfigureAs=client
Obsolete; supported only for backward compatibility with earlier versions of LUM Runtime.

Specifies whether the workstation is a server or a client only. Possible values are **server** and **client**.

Transport= *Not used on the Windows platform.*

MachineName=
Not used on the Windows platform.

NCSCell= *Not used on the Windows platform.*

UserName= The name of the user of this workstation.

GroupName= The name of the group to which the user of this workstation belongs.

DebugProc=no
Specifies whether or not the LUM Runtime processes must be started in debug traced mode. Possible values are **yes** and **no**. This parameter can be changed only by editing the configuration file.

DebugNCS=no
Specifies whether or not the LUM Runtime processes must be started in debug traced mode and additional communication-related information collected. Possible values are **yes** and **no**. This parameter can be changed only by editing the configuration file.

DebugToFile=no
Specifies whether or not the LUM Runtime processes are started in debug traced mode and the output is redirected to a file. Possible values are **yes** and **no**. This parameter can be changed only by editing the configuration file.

ConcurrentNodelock=No
Obsolete; supported only for backward compatibility with earlier versions of LUM Runtime.

Specifies whether or not the LUM Runtime concurrent nodelock process has to be started. Possible values are **Yes** and **No**. This keyword is set by the system to **Yes** when the first concurrent nodelocked password is installed and is set to **No** when the last concurrent nodelocked password is deleted from the nodelock file.

LogLevel=1 *Obsolete; supported only for backward compatibility with earlier versions of LUM Runtime.*

Level of logging of the concurrent nodelock process:

0 No logging

1 Logging of license add, delete, change events

2 Logging of license not granted events

3 Logging of all level 1 and 2 events plus initializations, requests and releases of licenses. Error conditions are logged too.

LogMsgsMaxNum=1000
Obsolete; supported only for backward compatibility with earlier versions of LUM Runtime.

Maximum number of messages logged by the concurrent nodelock process.

LogFile= *Obsolete; supported only for backward compatibility with earlier versions of LUM Runtime.*

Log file path and name of the concurrent nodelock process log.

| **CommunVersion=V4R5**
| Version of LUM Runtime communication subcomponent.

| *Supported only for backward compatibility.*

| **RuntimeVersion=V4R5**
| Version of LUM Runtime runtime subcomponent.

| *Supported only for backward compatibility.*

NCSSupportVersion=
Not used on the Windows platform.

Communication=yes
The workstation is configured to communicate in a network.

NamespaceBindingSupport=
Not used on the Windows platform.

AdvancedConfiguration=no
The user selected **Advanced Configuration** when configuring the workstation.

[iFOR/LS GLBD-Configuration]

Create= *Not used on the Windows platform.*

CreateFrom= *Not used on the Windows platform.*

Family= *Not used on the Windows platform.*

DefaultCell= *Not used on the Windows platform.*

SelfClean= *Not used on the Windows platform.*

Frequency= *Not used on the Windows platform.*

Timeout= *Not used on the Windows platform.*

`[iFOR/LS LMD]`

BackupMode=daily

The mode of the LUM Runtime database backup procedure. Possible values are:

daily The backup is started at the time specified in the **BackupParm** parameter.

weekly The backup is started at approximately midnight (00:00) of the day specified in the **BackupParm** parameter.

changes The backup is made each time the database is changed, such as when an object is added or deleted.

none No backup takes place.

This parameter value must be the same on all servers within your licensing environment. It can be changed only by editing the configuration file.

BackupParm=0

If **BackupMode** is **daily**, the hour when the backup occurs (midnight=0). If **BackupMode** is **weekly**, the day of the week when the backup occurs (Sunday=0).

This parameter value must be the same on all servers within your licensing environment. It can be changed only by editing the configuration file.

BackupPath=drive:\ifor\ls\conf\backup

The path where the server files and databases are copied during the automatic backup procedure. This parameter can be changed only by editing the configuration file.

NumberOfLogFile=2

The number of log files LUM Runtime writes. For example, if `logdb` is the log file name, and **NumberOfLogFile** is set to **2**, LUM Runtime changes the name to `logdb00_`. When it is full, it starts logging events on `logdb01_`. When this is full, it restarts writing on `logdb00_`. This parameter can be changed only by editing the configuration file.

MaxLogFileSize=10

The maximum length of the log files, in tens of kilobytes.

After that size is reached, LUM Runtime starts writing on another log file. This parameter can be changed only by editing the configuration file.

ValidityPeriod=15
Internal period, in days, to validate per-seat licenses stored on the nodelocked license server against the central registry. This parameter can be changed only by editing the configuration file.

HALFrequency=30
The length, in seconds, of the interval at which servers in a cluster synchronize data among themselves. You can increase this number if you have performance problems, but doing so delays synchronization between members of a cluster.

```
[iFOR/LS NCS-Server]
```

llbd= *Not used on the Windows platform.*

glbd= *Not used on the Windows platform.*

ipPort=1515 The TCP/IP port number the license server listens to when supporting its clients.

ipGDBPort=10999
The TCP/IP port number the central registry license server listens to when supporting its clients.

ipNDLPort=12999
The TCP/IP port number the nodelocked license server listens to for remote administration.

ipHALPort=11999
The TCP/IP port number used for internal communication between by servers in a cluster. Change this number only if port 11999 is already used for some other purpose. If you change this value, change it also on cluster members.

netbiosPort=115
The NetBIOS port number the license server listens to when supporting its clients.

netbiosGDBPort=109
The NetBIOS port number the central registry license server listens to when supporting its clients.

netbiosNDLPort=12999
The NetBIOS port number the nodelocked license server listens to for remote administration.

ipxPort=1515 The IPX port number the license server listens to when supporting its clients.

ipxGDBPort=10999
The IPX port number the central registry license server listens to when supporting its clients.

ipxNDLPort=12999
The IPX port number the nodelocked license server listens to for remote administration.

RunGLBD= *Not used on the Windows platform.*

RunGDB=no Whether the central registry license server is to be started on this workstation. Possible values are **yes** and **no**.

DisableRemoteAdmin=no
Whether the administration of this network license server is to be disabled when using the administration tool started on a different server. Possible values are **yes** and **no**.

DisableRemoteNDLAdmin=yes
Whether the administration of this nodelocked license server is to be disabled when using the administration tool started on a different server. Possible values are **yes** and **no**.

LogAllEvents=no
Whether all the events are to be logged on the license servers. Possible values are **yes** and **no**.

LogFile=drive:\ifor\ls\conf\log_file
Obsolete; supported only for backward compatibility with earlier versions of LUM Runtime.

Log file path and name of the license server process log.

LogPath=drive:\ifor\ls\conf
Log file path of the license server process log.

ColdStart=no Whether the license servers restart from scratch, with no record of licenses in use granted before stopping (**yes**), or not (**no**). Cold start is not possible for reserved and per-seat licenses. This parameter can be changed only by editing the configuration file.

DCEDWAITTIME=
Not used on the Windows platform.

RunNDL=yes Whether the nodelocked license server is to be started on this workstation. Possible values are **yes** and **no**.

RunLMD=no Whether the network license server is to be started on this workstation. Possible values are **yes** and **no**.

UseHostTable=no
Not used on the Windows platform.

PassiveTime=300
Not used on the Windows platform.

MaxActivities=512
Not used on the Windows platform.

MaxActivitiesThreshold=100
Not used on the Windows platform.

TraceActivities=no
Not used on the Windows platform.

`[iFOR/LS Server Logging]`

LogGrant=no Log when a license was granted or released. Possible values are **yes** and **no**.

LogCheckin=no
Log when a licensed product has sent a check-in call to the server to notify it that the product is running. Possible values are **yes** and **no**.

LogWait=no Log when a license request cannot be satisfied because no licenses are available, and the user is added to a queue. Possible values are **yes** and **no**.

LogVendor=yes
Log when a new vendor was added or deleted. Possible values are **yes** and **no**.

LogProduct=yes
Log when a product of a new vendor was registered or deleted. Possible values are **yes** and **no**.

LogTimeout=no
Log when the server has canceled the request for a license because the check period has expired. Possible values are **yes** and **no**.

LogErrors=yes Log server errors that do not stop the server, but return a status code and a message. Possible values are **yes** and **no**.

LogVendorMsg=yes
Log error messages the vendor inserted in the product. Possible values are **yes** and **no**.

LogSvrStartStop=no
Log the successful start or stop of the license server. Possible values are **yes** and **no**.

`[iFOR/LS NetBIOS-Configuration]`

LanAdaptor=0 The LAN adapter your workstation uses to communicate with the NetBIOS protocol. (Windows NT only)

NCBS= *Not used on the Windows NT platform.*

HasOS2Clients=no
Change it to **yes** on a Windows NT workstation that must support OS/2 clients using NetBIOS.

```
[iFOR/LS Client]
```

Threshold_Level=80
The default value of the threshold level used in the basic license tool GUI. Used for vendor-managed use products, and for customer-managed use products when no specific threshold value is specified.

Threshold_Automatic=0
Whether the threshold is automatically checked periodically. Possible values are **0** (no) or **1** (yes).

Refresh_Automatic=0
Whether the basic license tool is to do a periodic automatic refresh. Possible values are **0** (no) or **1** (yes).

Refresh_Frequency=60
The frequency, in minutes, of the automatic refresh.

ReadTimeout=4
The maximum wait time, in seconds, for an application to receive a response from the nodelocked license server via Interprocess Communications. The minimum is 1 and the maximum is 60. You may need to increase this value if performance on your workstation is poor. This parameter can be changed only by editing the configuration file.

CleanupAutomatic=No
Whether the basic license tool automatically cleans stale licenses. Possible values are **yes** and **no**. You can change this parameter only by editing the configuration file.

```
[iFOR/LS NCS-Client]
```

UseDirectBindingOnly=yes
Whether the client licensed applications running on this workstation are to locate the license servers using direct binding only. Always **yes** on the Windows platform.

FilterNDL=no Whether nodelocked licenses are to be excluded from the set of licenses administered by the basic license tool.

FilterNet=no Whether network licenses are to be excluded from the set of licenses administered by the basic license tool.

NumDirectBindServers=3
The number of direct binding servers the client applications

are configured to point to directly, using just transport protocol family, name and port number. Specify the **DirectBindServer** keyword for each server the client points to. If you need to contact the central registry license server, there must also be an entry for it. The default ip port numbers are 1515 for the license server and 10999 for the administration server. Make sure you insert the correct ones if you are not using the defaults.

DirectBindServer1=ip:thelma.rnsl.ibm.com[1515]
The format is *family:servername[port]*.

where *family* can be:

ip Any platform

netbios
Windows NT or Windows Terminal Server

ipx Windows NT, Windows Terminal Server, or Windows 95

DirectBindServer2=ip:louise.rnsl.ibm.com[1515]
The format is *family:servername[port]*.

where *family* can be:

ip Any platform

netbios
Windows NT or Windows Terminal Server

ipx Windows NT, Windows Terminal Server, or Windows 95

DirectBindServer3=ip:louise.rnsl.ibm.com[10999]
The format is *family:servername[port]*.

where *family* can be:

ip Any platform

netbios
Windows NT or Windows Terminal Server

ipx Windows NT, Windows Terminal Server, or Windows 95

OS2NumServers=0
The number of OS/2 NetBIOS servers the client applications are configured to communicate with.

OS2NetbiosServer1=no
Set this parameter to **yes** to indicate that **DirectBindServer1** is an OS/2 NetBIOS server. There can be one OS2NetbiosServer entry corresponding to any **DirectBindServer** entry.

GDBServer=ip:louise.rnsl.ibm.com[10999]
The format is *family:servername[port]*.

where *family* can be:

ip Any platform

netbios Windows NT or Windows Terminal Server

ipx Windows NT, Windows Terminal Server, or Windows 95

NumDirectBindNDLServers=2
The number of nodelocked license servers whose licenses can be administered remotely from this workstation.

DirectBindNDLServer1=ip:lab68082.rome.lab.tivoli.com[12999]
The format is *family:servername[port]*.

where *family* can be:

ip Any platform

netbios Windows NTor Windows Terminal Server

ipx Windows NT, Windows Terminal Server, or Windows 95

DirectBindNDLServer2=ip:lab68084.rome.lab.tivoli.com[12999]
The format is *family:servername[port]*.

where *family* can be:

ip Any platform

netbios Windows NT or Windows Terminal Server

ipx Windows NT, Windows Terminal Server, or Windows 95

Appendix B. Using the Nodelock File

This appendix explains how to prepare the nodelock file manually and how to use it. You might need to modify the nodelock file for configurations without a nodelocked license server or for backward compatibility.

To prepare and use the nodelock file:

1. Create or edit the nodelock file, which is located in the following directory:

   ```
   c:\'COMMON_APPDATA'\IBM\LUM
   ```

   directory:

   where:

   c The drive identifier

   COMMON_APPDATA
   The initial part of the path that is returned by the SHGetFolderPath API

   Examples of typical paths for the nodelock file across the Windows operating systems are as follows:

   Windows 95:
   `c:\windows\All User\Application Data\IBM\LUM`

   Windows 98:
   `c:\windows\Profiles\All User\Application Data\IBM\LUM`

   Windows NT:
   `c:\winnt\Profiles\All User\Application Data\IBM\LUM`

   Windows 2000:
   `c:\Documents and Settings\All User\Application Data\IBM\LUM`

   where *c* is the drive letter of the drive on which Windows is installed.

   The format of the nodelock file is:

   ```
   # comment
   vendorID  productPassword  Annotation  version  [serialNumber]
   ```

   where:

   - The first line starts with a comment character, #, and is included for information only. It indicates the product name and license expiry date.

- The second line is the product license. Its fields and their content are as follows:

  vendorID The vendor ID.

  productPassword
  The long alphanumeric password that enables the nodelock license.

Annotation The annotation field, which is used by the application developer to provide any unique enablement options of the license. This optional field, which is set to null ("") in the example, can contain up to 255 alphanumeric characters.

version The version number of the product.

serialNumber The serial number of a custom configuration license. This field can contain up to 31 alphanumeric characters.

Initially, this file could have entries similar to the following lines:

```
# nodelock example for the licensed product
expires 12/25/2003
543b0f87c093.02.81.87.92.34.00.00.00
gganccupqb5dauxabdw "" "2.0" "85AB2215691"
```

2. To help yourself and others identify the license in the future, because there may be other nodelocked software on the same computer, you should enter a comment above the license. That comment should include the full product name, version, and any expiration date.
3. Double-check the information to ensure that it is the same as that supplied by the vendor.
4. Test the product.

Appendix C. Operating Systems and License Use Management Support

This appendix identifies the operating systems supported by License Use Management Versions 4.5.2 and 4.5.5 (Table 18 on page 225), and Versions 4.5.8, 4.5.9, and 4.6.0 (Table 18 on page 225). It also indicates which have been tested and the level of support provided.

Table 17. Operating Systems and LUM Versions 4.5.2 and 4.5.5 Support

Operating System	**LUM Version 4.5.2**		**LUM Version 4.5.5**	
	Application Developer’s Toolkit	**Runtime**	**Application Developer’s Toolkit**	**Runtime**
AIX 4.3.x	Static link only	✓	Static link only	✓
AIX 4.1 and 4.2	✓	✓	✓	✓
HP-UX 10.20	✓	✓	✓	✓
HP-UX 11.0				✓
OS/2	Shared link only	✓	Shared link only	✓
Solaris 2.6	✓	✓	✓	✓
Solaris 2.7	✓	✓	✓	✓
Solaris 2.8				
IRIX 6.3 and 6.4	Static link only	✓	✓	✓
IRIX 6.5	Static link only	✓	✓	✓
Windows 95 and 98		✓		✓
Windows NT 4.0 Windows 2000	Shared link only	✓	✓	✓

Table 18. Operating Systems and LUM Versions 4.5.8, 4.5.9, and 4.6.0 Support

Operating System	**LUM Versions 4.5.8 and 4.5.9**		**LUM Version 4.6.0**			
	Application Developer's Toolkit	**Runtime**	**Application Developer's Toolkit**	**Runtime**	**Application Developer's Toolkit 64-Bit**	**Java GUI**
AIX 4.3.x	✓	✓	✓	✓	Static link only	
AIX 4.1 and 4.2	✓	✓	✓	✓		
HP-UX 10.20	✓	✓	✓	✓		
HP-UX 11.0		✓	Static link only	✓		✓
OS/2	Shared link only	✓				
Solaris 2.6	✓	✓	✓	✓		✓
Solaris 2.7	✓	✓	✓	✓	Static link only	✓
Solaris 2.8				✓		✓
IRIX 6.3 and 6.4	✓	✓	✓	✓		
IRIX 6.5	✓	✓	✓	✓	Static link only	✓
Windows 95 and 98		✓		✓		
Windows NT 4.0	✓	✓	✓	✓		
Windows 2000	✓	✓	✓	✓		
AIX 5.0 for IA-64				✓		✓

Glossary

A

accessor ID. In License Use Management, a generic string identifying the requestor of a license. Vendors can define their own multiuse rule based on the accessor ID.

ACID. See application client identifier.

application client identifier (ACID). In License Use Management, the unique identifier of the application client.

When a license is granted to a client, the ACID of the client is recorded in the central registry, which is checked at any new license request. This avoids granting a license twice to the same application client.

administrator. In License Use Management, the person who is responsible for setting up the LUM Runtime environment. The tasks of the administrator include:

- Installing and configuring nodelocked license servers, network license servers, network license clients, and the central registry
- Installing the software product licenses on the servers
- Monitoring the software products use through the Basic License Tool
- Configuring the network

application client. A computer that runs a software product and plays the role of the client in the traditional client-server model.

application server. A computer where an enabled product is installed, which provides shared access to the product to workstations (the application clients) over the network. In License Use Management, the application server is the LUM Runtime client. It requests the licenses for all its application clients.

annotation. See license annotation.

B

Basic License Tool. In License Use Management, the administration tool included in LUM Runtime, which enables the administrator to add or delete licenses from the server database, display the licenses installed, distribute the licenses among the servers available on the network, and generate reports on license usage and server events.

binding. In License Use Management, one of two methods by which a network license client can locate a server in order to request a license. See direct binding and namespace binding.

bundle. In License Use Management, a set of product licenses with a single serial number, called the *bundle ID*. The set of licenses can include any combination of license types. All products that are part of a bundle are enrolled together when the bundle is enrolled. A product that is part of a bundle cannot be removed separately: it can be removed only by removing the bundle.

C

capacity type. In License Use Management, a number that specifies the type of unit that a license applies to. The type of unit can be: users, processors, or another unit chosen by the vendor. If no capacity type is specified by the vendor, the license applies to users.

cell. See NCS cell.

central registry. In License Use Management, a database that contains information about:

- The enrollment and distribution of customer-managed use control products.
- Which application clients already have a per-seat license.
- Reservation of reservable licenses.

check period. In License Use Management, a time period during which a product holding a concurrent or unreserved reservable license must check in with the network license server. If the product does not check in during this period, the network license server assumes that the product is not running, and may release a granted license to another user.

cluster. In License Use Management, a group of network license servers that jointly serve vendor-managed concurrent licenses that are tied to the cluster rather than to an individual server. While some servers in the cluster are serving licenses, one or more servers remain in reserve, ready to take over should an active server fail.

compound password. In License Use Management, a password from which it is possible to extract multiple simple passwords, each representing one or more licenses. Enabled applications cannot use the compound password directly.

concurrent license. In License Use Management, a type of license, administered by the network license server, that can be used by different users from any node that is connected to a network license server. Concurrent licenses enable as many users to use a particular software product concurrently as there are licenses.

concurrent nodelocked license. In License Use Management, a nodelocked license that allows a limited number of concurrent uses of the licensed product on the node where the license is installed. Concurrent nodelocked licenses enable as many concurrent uses of a particular software product as there are licenses.

custom configuration. A selected combination of products, tailored by a vendor to the needs of one or more users. Each custom configuration is identified by unique serial number, which is incorporated into the custom configuration license.

custom configuration license. A special case of either a concurrent network license or a simple nodelocked license that contains a unique serial number identifying a custom configuration. See also license.

customer-managed use control. In License Use Management, a level of password use control in which the customer manages compliance with the terms of the software product acquisition. It is the customer’s responsibility to set the upper limit on the number of licenses that can be extracted and distributed, based on the terms of the software product acquisition.

D

default NCS cell. A cell that is identified by the default GLB object UUID. Machines in the default cell do not have the glb_obj.txt file.

direct binding. In License Use Management, a type of binding between network license servers and clients in which client applications locate license servers by means of a local text file that contains network addresses of the license servers.

direct binding servers list. In License Use Management, a set of network license servers and a central registry license server that collectively serve a set of network license clients.

dynamic nodelocking. In License Use Management, a way of using licensing APIs in which a compound password installed on a network license server carries simple nodelocked licenses. Upon first invocation of the product at a client, a simple nodelocked license is extracted from the compound password and installed on the client machine.

E

end user. In License Use Management, a user of license-enabled software products. The tasks of the end users may include:

- Installing LUM Runtime with the help of the administrator.
- Configuring LUM Runtime as a network license client.

enrollment certificate. In License Use Management, a mechanism for the distribution of licenses to end users. It is usually in the form of an electronic file, and contains all the information that is related to the licenses acquired for a license-enabled product.

G

gdb server. See central registry.

GLB. See global location broker.

glbd replica. In License Use Management, a copy, on a newly configured network license server, of a global location broker database that already exists on another server.

glb_obj.txt. A file that specifies the object UUID of the global location broker. The glb_obj.txt file makes it possible to override the default value by specifying a different GLB object UUID for a particular machine. The glb_obj.txt file is used only in special configurations that require several disjoint GLB databases (each of which is possibly replicated). In most networks and internets, there is only one GLB database (possibly replicated), and machines do not need to have a glb_obj.txt file. If a machine has a glb_obj.txt file, the UUID in the file identifies the GLB object to which that machine directs lookups and updates.

global location broker. Part of the Network Computing System (NCS) that enables clients to locate servers in a network or internet. It is a process that manages a database that stores the locations (network addresses and port numbers) where server processes are running. The global location broker process maintains this database and provides access to it.

H

hard stop. In License Use Management, a policy according to which, if the end user starts the product and there are no licenses available, the product does not start.

high-availability licensing. In License Use Management, an option that makes it possible for a cluster of network license servers to jointly serve concurrent licenses, with one or more servers in reserve in case a server goes down. The software vendor must create passwords to be enrolled on the cluster rather than on an individual server.

high-water mark. In License Use Management, the maximum number of soft stop licenses that have been granted for a given product, over the number of licenses enrolled for that product. It is updated when the soft stop policy is set. In hard stop policy no updating of the high-water mark occurs, since it is assumed that the product stops its execution if no licenses are available.

I

internet. A set of two or more connected networks. The networks in an internet do not necessarily use the same communications protocol. License Use Management supports the following protocols on OS/2:

- NetBIOS
- TCP/IP
- IPX

License Use Management supports the following protocols on Windows NT:

- NetBIOS
- TCP/IP
- IPX

On Windows 95 and Windows 98, NetBIOS is not supported. On Windows 98, IPX is not supported. On AIX, HP-UX, IRIX, and Solaris, License Use Management supports only TCP/IP.

initial key. A license key for a custom configuration license generated without using the Upgrade flag. It is an encrypted character string that specifies some terms of the acquisition of the selected combination of software products in a customer's initial custom configuration. Contrast with replacement key.

IPX. A communication protocol that creates, maintains, and terminates connections among network devices (workstations, file servers, or routers, for example).

J

Java. An object-oriented programming language for portable interpretive code that supports interaction among remote objects. Java was developed and specified by Sun Microsystems, Incorporated.

JavaBeans. The platform-independent, component architecture for the Java programming language. JavaBeans enables software developers to assemble pieces of Java code (″Beans″) into a graphical drag-and-drop development environment.

K

key. See password.

L

license. Permission to use an instance of a licensed software product or service, according to the basis on which the vendor charges for the product or service. Sometimes, a user needs more than one license to make full use of a particular product features.

The term *license* as used in the context of License Use Management does not refer to the license agreement that governs use of and rights to a product.

license annotation. A string that the vendor can use to modify the use of a license.

license database. In License Use Management, the database of licenses that a license server maintains.

license-enabled product. A product that is enabled for license use management.

A vendor provides a license-enabled product together with a password that authorizes use of the product. The password contains an encryption of certain terms of the acquisition of the product (such as how many licenses the customer can use, the expiration date of the licenses, and the type of license).

license information. In License Use Management, the information that describes licenses. This information consists of product name, product version, number of licenses, license type, start and end dates for the licenses, and a time stamp.

license key. See password.

licensed product. See license-enabled product.

license server. A program that provides the license services, administering licenses for software products. It may be a network license server or a nodelocked license server.

local location broker. Part of the network computing system (NCS). It manages the local location broker (LLB) database, which stores information about NCS-based server programs that run on the local host.

location broker. See local location broker and global location broker.

log file. A database that records messages and errors from the license server, and sometimes from licensed products as well.

M

multiuse rules. In License Use Management, rules that define the conditions under which multiple invocations of a product require only a single license. These rules are applicable only to concurrent access, concurrent nodelocked, and per-server licenses. The vendor of the product defines multiuse rules.

N

namespace binding. In License Use Management, a binding mechanism in which the network license servers register themselves with the global location broker, which locates an appropriate license server when a client requests a license. Namespace binding is not available on Windows platforms.

NCS. A set of software components, developed by Apollo Computer Inc., that conform to the

Network Computing Architecture. These components include the Remote Procedure Call (RPC) runtime library and the Location Brokers.

NCS cell. A logical grouping of clients and servers; a subset of a network. Machines in one cell cannot communicate with machines in other cells. Machines cannot be in more than one cell at a time. Machines in the same cell are identified by the same global location broker (GLB) object Universal Unique Identifier (UUID).

network. A group of nodes and the links that interconnect them.

network license. In License Use Management, a license that is maintained on a network license server for use upon request by a LUM Runtime client.

network license client. In License Use Management, a node configured to make use of licenses by requesting them from a network license server.

network licensed product. In License Use Management, a licensed product that is enabled such that the licenses are maintained on a server for use upon request by a LUM Runtime client.

network license server. In License Use Management, a node in the network on which network licenses are stored for use by LUM Runtime clients.

node. A machine in the network. In License Use Management, it can be configured as a nodelocked license server, a network license client, a network license server, the central registry license server, or a combination

nodelocked license. In License Use Management, a type of license locked to a specific node, so that the product can be used only at that node. The nodelocked license is installed on the machine for which it was created.

nodelocked license server. In License Use Management, a server on a node that manages nodelocked licenses on that node.

non-runtime-based enablement. In License Use Management, a type of license enablement for a product with simple nodelocked licenses that does not make use of LUM Runtime on the end user's machine. The password is stored in a special file when the enabled product is installed. When the enabled product is started, it checks the file to ensure that there is a valid license.

O

object. In the Network Computing System, an entity that is manipulated by well-defined operations. Databases, files, directories, devices, processes, and processors are all objects.

P

password. An encrypted character string that specifies some terms of the acquisition of a software product. See also simple password, compound password.

password use control level. In License Use Management, a level of control of compliance with the terms of the acquisition of a license-enabled product. The password use control levels are:

- customer-managed use control
- vendor-managed use control

per-seat license. In License Use Management, a license used to enable client / server applications that are constructed for multiple-server solutions. Assignment of a per-seat license to an application client is permanent. Unused application client licenses are kept in a central repository, which all the application servers share. They also share a central list of application clients that have an assigned license. If an application client connects to multiple application servers, only one license is assigned to it.

per-server license. In License Use Management, a license used to enable client / server applications that are constructed for multiple-server solutions. Each server license is associated with a specific number of clients. This number represents the maximum number of clients that may concurrently request that server application services at any given time. Assignment of a per-seat license to an application client is temporary. If an application client connects to multiple application servers at the same time, it is assigned more than one license.

product ID. In License Use Management, a number that identifies a vendor licensed software product. By means of product IDs, the license server can distinguish between products from the same vendor.

Q

queue. In License Use Management, a sequence of users who are waiting for a concurrent license to become available so they can run a product. The administrator can monitor the number of users in queue through the Basic License Tool.

R

replacement key. A license key for a custom configuration license generated using the Upgrade flag. It is an encrypted character string that specifies some terms of the acquisition of the selected combination of software products in a customer's upgraded custom configuration. Contrast with initial key.

replica. See glbd replica.

report. In License Use Management, a summary of the events related to the licenses that are installed on the selected servers, filtered as the administrator specified. Examples of events are:

- Requests for licenses for a product in a given interval of time.
- Server startup.

reservable license. In License Use Management, a network license that the administrator can reserve for the exclusive use of a user, a group, or a node. The reservation is for a specified time period.

reserved license. In License Use Management, a license that the administrator has reserved for the exclusive use of a user, a group, or a node.

runtime-based enablement. In License Use Management, a type of license enablement for a product with nodelocked licenses that uses LUM Runtime on the end user's machine to manage the licenses.

S

selected servers. In License Use Management, the servers that the administrator is working with through the Basic License Tool. All the products whose licenses are installed on the selected servers are displayed in the Basic License Tool main window.

serial number. In License Use Management, a unique number that identifies a specific custom configuration.

simple nodelocked license. In License Use Management, a nodelocked license that allows an unlimited number of simultaneous uses of the licensed application on the local machine.

simple password. In License Use Management, a password that, once enrolled on a license server, represents one or more licenses. Enabled applications can use the simple password directly.

socket server. The process that allows LUM Runtime clients and servers to communicate among themselves through the NetBIOS protocol.

soft stop. A policy according to which, if the end user starts the product and there are no licenses available, the product starts.

T

target. In License Use Management, the node at which a password is to be installed. If the password specifies a nodelocked license, the target is the node where the licensed product is run. If the password specifies multiple nodelocked license (that is, a compound password for nodelocked licenses) or network licenses, the target is a node at which the network license server (i4lmd) is running.

target ID. In License Use Management, a unique identifier of a node. A vendor can generate a password that can be installed only on a node that has a specific target ID. The target ID can be based on hardware or generated by LUM Runtime.

TCP/IP. Transmission Control Protocol/Internet Protocol. A set of communication protocols that support peer-to-peer connectivity functions for both local and wide area networks.

threshold. In License Use Management, a percentage of licenses; if more than this percentage of licenses for a product are in use, messages about the level of use are logged.

time stamp. In License Use Management, a number that identifies the date and time at which a set of licenses was created.

try-and-buy license. In License Use Management, a nodelocked license that has a fixed duration and a start date equal to the date when the license is enrolled. A try-and-buy license is made available for purposes of evaluating the application, and can be replaced by a production license after evaluation.

U

universal unique identifier. An identifier that is used by NCS to identify interfaces, objects, and types.

use-once license. In License Use Management, a type of license, administered by the license server, that is effective for only a single instance of starting a product or using a service. The license server decrements the number of available use-once licenses each time the product or service is used.

use-once nodelocked license. In License Use Management, a license that is valid for only a single instance of starting a product or using a service, on the node where the license is installed.

user file. In License Use Management, a flat ASCII file, which the administrator creates with a text editor, that lists users who specifically are or are not allowed to use specified products.

UUID. See universal unique identifier.

V

vendor ID. In License Use Management, the identifier of a vendor of licensed products. Vendor IDs are a LUM Runtime specific usage of NCS Universal Unique Identifiers (UUIDs).

vendor-managed use control. In License Use Management, a level of password use control in which the vendor manages compliance with the terms of the software product acquisition. The customer of a vendor-managed use product supplies a unique identifier (target ID) of each machine where product licenses are to be installed. The vendor uses this information to create the password, which is tied to the target workstation and cannot be used on another workstation.

Index

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SH19-4347-03