

License Use Management

IBM

# Using License Use Management Runtime for Windows

Version 4.5.5 (September 6, 1999)



License Use Management

IBM

# Using License Use Management Runtime for Windows

Version 4.5.5 (September 6, 1999)

**Note**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xiii.

**ISO 9001 Certification**

This product was developed using an ISO 9001 certified quality system.

Certification has been awarded by the Italian quality system certification group, CSQ (Certification No. CISQ/CSQ 9150.IBM7).

CSQ is a member of the mutually recognized organization of European assessors, ITQS, which assesses and certifies quality systems in the field of information technology enterprises.

**Third Edition (September 1999)**

This major revision obsoletes and replaces SH19-4347-01. The major changes are described in "Summary of Changes" on page xix. In the hard copy version of this book, technical changes are marked by a vertical line in the left margin. In the .HTM version, technical changes appear in purple. Post-publication technical changes are marked in brown.

This edition applies to Version 4.5.5 of IBM License Use Management Runtime for Windows, a part of License Use Management for Windows, Program Number 5697-B23, and to Version 4.5.5 of IBM License Use Management Runtime for Windows NT Alpha, a part of License Use Management for Windows NT, Program Number 5697-E77, 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.

A form for reader's comments is provided at the back of this publication. If the form has been removed, 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.

**Copyright 1994, 1997 Isogon Corp.**

**Copyright International Business Machines Corporation 1995, 1999. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

## Contents

<b>Notices</b> . . . . .	xiii
Trademarks . . . . .	xiv
<b>About This Book</b> . . . . .	xv
Who Should Use This Book . . . . .	xv
How This Book is Organized . . . . .	xv
Where to Find More Information . . . . .	xvi
Online Books . . . . .	xvi
License Use Runtime README File . . . . .	xvii
IBM LUM Web Site . . . . .	xvii
Notational Conventions . . . . .	xviii
Notation for Drives and Directories . . . . .	xviii
<b>Summary of Changes</b> . . . . .	xix
Changes in the Third Edition . . . . .	xix
General Changes . . . . .	xix
Specific Changes . . . . .	xix
<b>Chapter 1. Introduction to License Use Runtime</b> . . . . .	1
License Use Management at a Glance . . . . .	1
License Use Management Vendor Perspective . . . . .	1
License Use Management Customer Perspective . . . . .	2
Basic Concepts of License Use Management . . . . .	2
License Use Runtime Platforms . . . . .	4
License Passwords . . . . .	5
Password Use Control Levels . . . . .	6
Vendor-Managed Use Products . . . . .	6
Customer-Managed Use Products . . . . .	7
License Types . . . . .	7
Nodelocked Licenses . . . . .	7
Simple Nodelocked Licenses . . . . .	8
Concurrent Nodelocked Licenses . . . . .	8
Use-Once Nodelocked Licenses . . . . .	8
Per-Server Licenses . . . . .	9
Network Licenses . . . . .	9
Concurrent Licenses . . . . .	10
Reservable Licenses . . . . .	10
Use-Once Licenses . . . . .	11
Per-Seat Licenses . . . . .	11
License Policies . . . . .	12
Vendor-Controlled Policies . . . . .	12
Try-and-Buy Policy . . . . .	12
Multiuse Rules . . . . .	12
Product Wait Queues . . . . .	12
License Annotation . . . . .	13

Custom Configuration	13
Customer-Controlled Policies	13
Hard Stop/Soft Stop Policy	13
User Access Restriction	14
Switching from Per-Server to Per-Seat Licenses	14
License-Enabling Models	15
Scalable Installation and Configuration	16
License Creation Tool	17
License Administration Tool	18
High-Availability Licensing	20
Backup Procedure	21
Working with Licensed Products	21
Central Registry License Server	21
Working with Nodelocked Licenses (Non-Runtime-Based Enabling)	22
Working with Nodelocked Licenses (Runtime-Based Enabling)	23
Working with Use-Once Licenses	24
Working with Concurrent Licenses	26
Working with Reservable Licenses	28
Working with Per-Server Licenses	31
Working with Per-Seat Licenses	32
<b>Chapter 2. Planning Your Network Licensing Environment</b>	<b>35</b>
Selecting Your Servers	35
Network Computing System (NCS)	36
Planning Direct Binding	37
Planning the Central Registry	40
Planning for Java Applications and Applets	40
Planning Clusters	41
Restrictions on Cluster Size and Composition	41
Examples of Cluster Size Rules	43
Cluster Membership Considerations	45
Running the NetBIOS Socket Server	45
Verifying Network Connections	46
<b>Chapter 3. Installing License Use Runtime</b>	<b>47</b>
Before You Install	47
Selecting License Use Runtime Components	48
Hardware Requirements	49
Software Requirements	49
Protocol Configuration Requirements	50
Obtaining the License Use Runtime Code	50
Deciding How to Install	51
Installing License Use Runtime Interactively	51
Installing after an Uninstallation	57
Installing License Use Runtime Noninteractively	58
Using the Installation Command	59
Creating a Response File	60
Editing the Sample Response File	60

Saving a Response File During an Interactive Installation . . . . .	61
Verifying and Updating the PATH Environment Variable . . . . .	62
Installation Using a Code Server . . . . .	62
Uninstalling License Use Runtime . . . . .	63
Installing LUM Java Client Support . . . . .	63
Before You Install . . . . .	63
Disk Space Requirements . . . . .	63
Software Requirements . . . . .	63
Obtaining LUM Java Client Support Code . . . . .	64
Installing LUM Java Client Support . . . . .	64
Uninstalling LUM Java Client Support . . . . .	64
<b>Chapter 4. Getting Started with License Use Runtime . . . . .</b>	<b>67</b>
Setting Up Your Servers and Clients . . . . .	67
Configuring to Handle Nodelocked Licenses . . . . .	67
Configuring to Handle Network Licenses . . . . .	67
Determining the Configuration Required . . . . .	68
Before You Configure . . . . .	70
Customizing Log Information . . . . .	70
Automatically Starting License Servers . . . . .	71
Disabling Remote Administration . . . . .	71
Configuring Direct Binding . . . . .	71
Using the Configuration Tool GUI . . . . .	72
Using the Configuration Tool Command-Line Interface . . . . .	73
Scenario 1: Configuring a Standalone Nodelocked License Server . . . . .	73
Command-Line Equivalent . . . . .	75
Scenario 2: Configuring a Nodelocked License Server in a Network . . . . .	75
Command-Line Equivalent . . . . .	79
Scenario 3: Configuring a Network License Server . . . . .	79
Command-Line Equivalent . . . . .	84
Scenario 4: Configuring a Network License Client . . . . .	84
Command-Line Equivalent . . . . .	87
Scenario 5: Configuring the Central Registry License Server . . . . .	88
Command-Line Equivalent . . . . .	90
Starting and Listing Your Processes . . . . .	90
License Use Runtime Processes Defined as Windows NT Services . . . . .	90
Managing License Use Runtime Processes with Service Manager Tool . . . . .	91
Verifying Connections to Servers . . . . .	92
Administering License Use . . . . .	92
Using the Basic License Tool GUI . . . . .	92
Starting the Basic License Tool GUI . . . . .	92
Refreshing License Information . . . . .	93
Selecting Servers . . . . .	93
Using the Basic License Tool Command-Line Interface . . . . .	94
Performing Basic Administration . . . . .	94
Scenario 6: Managing a Licensed Product . . . . .	94
Enrolling the Product . . . . .	94
Distributing the Licenses . . . . .	97

Generating Reports	99
Monitoring the Number of Product Users	100
Command-Line Equivalent	101
Scenario 7: Managing Reservable Licenses	102
Command-Line Equivalent	105
Exercising Customer-Controlled Policies	106
Scenario 8: Switching from Per-Server to Per-Seat Licenses	106
Command-Line Equivalent	109
Scenario 9: Using the Hard Stop/Soft Stop Policy	109
Command-Line Equivalent	112
Scenario 10: Restricting User Access	113
Administering High-Availability Licensing	114
Scenario 11: Creating and Administering a Cluster	114
Creating a Cluster	114
Activating Cluster Members	117
Adding a Cluster Member	119
Deactivating a Server	120
Viewing Licenses Being Served	121
Enrolling and Removing Licenses on a Cluster	122
Command-Line Equivalent	122
Upgrading a Custom Configuration	123
Command-Line Equivalent	126
<b>Chapter 5. License Use Runtime Commands</b>	127
i4blt - Basic License Tool	128
General Rules for the i4blt Command	128
Primary Command Options	129
-a Enroll a Product	130
-U Update a Product	132
-E Extract and Distribute Licenses	134
-d Delete a Product License	135
-R Reserve Licenses; Delete or Update Reserved Licenses	137
-C Clean Up Stale Licenses	138
-l Display a List	139
-s Display Product License Status	144
-r Generate a Report	145
-x Delete Server Log Entries	148
-m Monitor and Log Threshold Events	149
-H Administer High-Availability Licensing	150
-h Display Help	152
i4cfg - Configuration Tool	153
License Use Runtime Tools	157
i4tv - Test Verification Tool	157
i4target - Target View Tool	158
License Use Runtime Processes	160
i4ls - Network License Server Process	160
i4ls -b - Nodelocked License Server Process	161
i4ls -a - Central Registry License Server Process	163



i4nbssvr - NetBIOS Socket Server Process (Windows NT only)	164
i4lct - License Creation Tool	164
Defining Rules for Multiple-Use Concurrent Licenses	174
i4nat - Nodelocked Administration Tool (for Backward Compatibility Only)	175
-a Add a Nodelocked License	176
-d Delete a Nodelocked License	177
-l Display License Information	178
-u Update Concurrent Nodelocked License Information	178
-h Display Command Line Interface Usage	179
<b>Chapter 6. Hints and Tips</b>	<b>181</b>
Managing Time Zone	181
Using the Built-In Backup and Recovery Procedure	181
Causes for Corrupted Definition or Database Files	181
Automatic Backup Procedure	182
Recovery Procedure	182
Manual Backup	183
Manual Recovery	183
Managing the Reports Log Files	183
Managing a Custom Configuration	184
Before Requesting a License Upgrade	184
Deleting Products or Reducing Numbers	184
Deleting Keys	184
<b>Chapter 7. Troubleshooting</b>	<b>185</b>
Checking License Details	185
Troubleshooting Licenses (All Types)	187
Troubleshooting Nodelocked Licenses	187
Troubleshooting Network Licenses (All Types)	188
Troubleshooting Reservable and Reserved Licenses	189
Troubleshooting Per-Server and Per-Seat Licenses	189
Troubleshooting Licenses of Customer-Managed Use Products	189
Troubleshooting Licenses of Vendor-Managed Use Products	190
Troubleshooting Performance Problems	190
Basic License Tool Performance	190
Performance in a Direct Binding Environment	190
Troubleshooting Heavy Server Workloads	190
Troubleshooting License Use Runtime Processes	190
Starting Required Processes	190
Restart and Recovery	191
Troubleshooting Custom Configuration Licenses	192
Cannot Install a Custom Configuration License	192
Troubleshooting Network Connections	192
Troubleshooting Direct Binding	192
Troubleshooting Communication Protocols	192
NetBIOS	193
TCP/IP	193
IPX on Windows 98	194

	Troubleshooting Coexistence with Other Products . . . . .	194
	Coexistence with Digital DCE for Windows NT . . . . .	194
	Troubleshooting the Hardware . . . . .	194
	Collecting Error Log Data . . . . .	195
	Running Processes in Traced Mode . . . . .	195
	Running Enabled Applications in Traced Mode . . . . .	196
	Running Tools in Traced Mode . . . . .	196
	Collecting Other Data . . . . .	196
	Troubleshooting LUM Java Client Support . . . . .	197
	Java Program Cannot Read the User Name . . . . .	197
	Incomplete View of an Applet . . . . .	197
	Troubleshooting Java Client Support Uninstallation . . . . .	198
	<b>Appendix A. License Use Runtime Configuration File . . . . .</b>	<b>199</b>
	<b>Appendix B. Using the Nodelock File . . . . .</b>	<b>209</b>
	<b>Appendix C. Features and Functions Added in Version 4 . . . . .</b>	<b>211</b>
	<b>Glossary . . . . .</b>	<b>213</b>
	<b>Index . . . . .</b>	<b>219</b>

## Figures

1.	Licensing Concepts Summary	6
2.	Using a Nodelocked License (Non-Runtime-Based Enabling)	22
3.	Using a Nodelocked License (Runtime-Based Enabling)	23
4.	Using a Use-Once License for a C-Language Program	24
5.	Using a Use-Once License for a Java Application or Applet	25
6.	Using a Concurrent License for a C-Language Program	26
7.	Using a Concurrent License for a Java Application or Applet	27
8.	Using a Reservable License for a C-Language Program	28
9.	Using a Reservable License for a Java Application or Applet	29
10.	Using a Per-Server License	31
11.	Using a Per-Seat License for a C-Language Program	32
12.	Using a Per-Seat License for a Java Application or Applet	33
13.	Direct Binding with Network License Servers and Nodelocked License Servers	38
14.	Direct Binding with Java Client Support	39
15.	Welcome Window	52
16.	READ.ME Window	53
17.	Choose Destination Location Window	54
18.	Select Components Window	55
19.	Setup Window	55
20.	Setup Complete Window	56
21.	Setup Complete Window	57
22.	Configuration Tool Notebook - Standalone Nodelocked License Server	74
23.	Configuration Tool Notebook - Log Page	75
24.	Configuration Tool Notebook - Nodelocked License Server in a Network	76
25.	Configuration Tool Notebook - User Section	77
26.	Configuration Tool Notebook - Direct Binding Section	78
27.	Configuration Tool Notebook - Network License Server	80
28.	Configuration Tool Notebook - Log Page	81
29.	Configuration Tool Notebook - Direct Binding Section	82
30.	Configuration Tool Notebook - User Section	83
31.	Configuration Tool Notebook - Network License Client	85
32.	Configuration Tool Notebook - Direct Binding Section	86
33.	Configuration Tool Notebook - User Section	87
34.	Configuration Tool Notebook - Central Registry License Server	88
35.	Configuration Tool Notebook - Direct Binding Section	89
36.	Service Manager Tool Window	91
37.	Basic License Tool Window	93
38.	Import Window	95
39.	Enroll Product Window	95
40.	Enroll Licenses Window	96
41.	Basic License Tool Window with SMARTJava Enrolled	97
42.	Distribute Licenses Window	97
43.	Set Number of Licenses Window	98
44.	Distribute Licenses Window with Number of Licenses Set	98

45.	Basic License Tool Window with Distributed Licenses . . . . .	99
46.	Reports Window . . . . .	99
47.	Concurrent Users Page . . . . .	101
48.	Basic License Tool Window with Reservable Licenses . . . . .	102
49.	Details Notebook with Reservable Tab . . . . .	103
50.	Reserving Reservable Licenses . . . . .	103
51.	Details Notebook with Reserved Licenses . . . . .	104
52.	Reserved Users Page with Reservable Licenses in Use . . . . .	104
53.	Un-Reserved Users Page . . . . .	105
54.	Reservable Page with Unreserved Licenses in Use . . . . .	105
55.	Basic License Tool Window with Per-Server Licenses Enrolled . . . . .	107
56.	Enabling Per-Seat Licensing . . . . .	108
57.	Basic License Tool Window with Per-Seat Licenses Enrolled and Enabled . . . . .	108
58.	Basic License Tool Window with Soft-Stop Licenses in Use . . . . .	110
59.	Update Licenses Window with High-Water Mark . . . . .	110
60.	Update Licenses Window - Enrolling More Licenses . . . . .	111
61.	Resetting the High-Water Mark . . . . .	111
62.	Basic License Tool Window with Licenses Updated . . . . .	112
63.	Clusters Window . . . . .	115
64.	Create Cluster Window . . . . .	115
65.	Define Cluster Members Window . . . . .	116
66.	Clusters Window with New Cluster Added . . . . .	116
67.	Details of New Cluster . . . . .	117
68.	Details of Cluster after Three Activations . . . . .	118
69.	Clusters Window with Cluster ID . . . . .	118
70.	Basic License Tool Window with HAL Test Product . . . . .	119
71.	Add Cluster Members Window . . . . .	120
72.	Details of Cluster after Adding a Server . . . . .	120
73.	Details of Cluster after Deactivation . . . . .	121
74.	Details of Servers Serving HAL Test Product . . . . .	122
75.	Initial State of the Concurrent Page of the Details Notebook for a Custom Configuration License . . . . .	123
76.	Enroll Product Window for Custom Configuration . . . . .	124
77.	Import Window for Custom Configuration . . . . .	125
78.	Enroll Product Window for Custom Configuration . . . . .	125
79.	Upgraded State of the Concurrent Page of the Details Notebook for a Custom Configuration License . . . . .	126

## **Tables**

1.	License Use Runtime Platforms . . . . .	4
2.	License-Enabling Models, License Types, and License Policies . . . . .	16
3.	Number of Servers in a Cluster . . . . .	43
4.	Example - Cluster with Three Initial Members . . . . .	44
5.	Example - Cluster with Six Initial Members . . . . .	45
6.	Required Components by Machine Role . . . . .	48
7.	Hardware Requirements . . . . .	49
8.	Disk Space Requirements . . . . .	49
9.	Transport Protocols . . . . .	50
10.	Configuration Required to Support All Types of Licenses . . . . .	69
11.	Configuration Options . . . . .	70
12.	Valid Uses of i4lct . . . . .	166
13.	License Use Runtime Processes . . . . .	191
14.	Features and Functions Added in Version 4 . . . . .	211



---

## 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

---

### Trademarks

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

AIX  
IBM  
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.

HP-UX is a registered trademark of Hewlett-Packard Company.

IRIX is a trademark of Silicon Graphics, Inc.

Solaris is a registered trademark of Sun Microsystems.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks, and Authenticode a trademark, 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 provides a guide to setting up the environment required to support licensed software products. Refer to Chapter 1, "Introduction to License Use Runtime" on page 1 for details on licensed products. This book describes License Use Management Runtime for Windows (referred to as License Use Runtime). It contains information about how to install and configure the servers where licenses will be installed and the clients that will use the products, and how to set up a network licensing environment. It explains how to manage the licenses for licensed software products you have installed.

Before reading this guide, follow the procedures described in the documentation accompanying the licensed software product you acquired.

---

## 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 using License Use Runtime.

The License Use Runtime system administrator 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 on their client machines and, if required, install and configure License Use Runtime with the support of the system administrator.

End users require only general knowledge of the Windows operating environment.

---

## How This Book is Organized

This book is divided into the following sections:

Chapter 1, "Introduction to License Use Runtime," 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 Your Network Licensing Environment," provides information on how to organize your network.

Chapter 3, "Installing License Use Runtime," gives you instructions on how to install and uninstall License Use Runtime components.

## About This Book

Chapter 4, “Getting Started with License Use Runtime,” provides scenarios for configuring License Use Runtime, administering product licenses, and exercising customer-controlled policies, using the License Use Runtime tools.

Chapter 5, “License Use Runtime Commands,” documents the License Use Runtime command line interface.

Chapter 6, “Hints and Tips,” provides hints and tips to better take advantage of License Use Runtime, and information on how to use the provided backup and recovery procedure.

Chapter 7, “Troubleshooting,” helps you to improve performance and to handle problems, should they arise when you use license-enabled products.

Appendix A, “License Use 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, “Features and Functions Added in Version 4” lists the features and functions that have been added in Version 4.

“Glossary” defines terms used in this manual.

---

## Where to Find More Information

This section lists other sources of information related to License Use Runtime.

### Online Books

The online books for License Use Runtime, listed in the rest of this section, are available in .HTM format for viewing with a Web browser.

The .HTM files are in the directory:

```
\ifor\win\bin\language
```

where *language* is the subdirectory for a language in which License Use Runtime has been installed on your machine. The directories for the languages are:

Brazilian Portuguese	PT_BR	Danish	DA_DK
English	EN_US	Finnish	FI_FI
French	FR_FR	German	DE_DE
Italian	IT_IT	Japanese	JA_JP
Korean	KO_KR	Norwegian	NO_NO
Simplified Chinese	ZH_CN	Spanish	ES_ES
Swedish	SV_SE	Traditional Chinese	ZH_TW

To view an .HTM file, open the file in your Web browser.

## About This Book

### Command Reference

The *Command Reference*, in addition to being part of this book (Chapter 5, “License Use Runtime Commands” on page 127), is available separately as an online book. The file is named:

lumcmd

### Message Reference

The *Message Reference* is an online book named:

lummsg

It lists Basic License Tool and Configuration Tool error messages, with an explanation, system action, and suggested user action.

### Using License Use Runtime

This book is available as an online book, as well as in hard copy. The file is named:

lumusg

A printable copy of this book in PostScript and PDF formats is available for download from the IBM LUM Web site <http://www.software.ibm.com/is/lum>.

### Using Application Developer's Toolkit

For information about how to license-enable software products for use with License Use Management, see *Using License Use Management Application Developer's Toolkit*, SH19-4362. If Application Developer's Toolkit is installed in your environment, it is available as an online book in .HTM format. The file is named:

lumtkt

### License Use Runtime README File

For changes to License Use Runtime or to this book that were made after the book went to press, see the README.ARK file on the CD-ROM or in the download package from the LUM Web site.

### IBM LUM Web Site

Visit the IBM License Use Management Web site at <http://www.software.ibm.com/is/lum> for information and news about IBM License Use Management, and to download License Use Runtime publications and code.

## About This Book

---

### Notational Conventions

This book uses the following notation in text:

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

*Italics*              Italic print is used for variables for which you must supply a value, for introducing new terms in the text, and for emphasis.

Monospacing        Monospacing indicates system messages and examples.



This icon marks important information that can affect the operation of the product or the completion of a task.

---

### Notation for Drives and Directories

When this book shows the path to a file in the `\ifor` directory, it means the `\ifor` directory on the drive where you installed License Use Runtime.

For example, if you installed License Use Runtime on the D: drive, when this book refers to:

```
\ifor\ls\conf\i4ls.log
```

it means:

```
D:\ifor\ls\conf\i4ls.log
```

---

## Summary of Changes

This section provides an overview of major changes made to this book.

---

### Changes in the Third Edition

The third edition of this book incorporates changes documented in the README.ARK and README.JCS files for Version 4.5.1 and Version 4.5.2, and changes made to Version 4.5.5 of the product.

### General Changes

This section summarizes the general, pervasive changes made to this book.

#### Viewing Books Online

Books are no longer supplied in INF format. View the HTML-format versions of books in your web browser.

#### Additional Operating Systems

Support has been added for the following operating systems:

Windows NT 4.0 Server, Terminal Server Edition (Windows Terminal Server), on the x86 platform. The behavior of License Use Runtime on Windows Terminal Server is the same as on Windows NT (x86).

Windows NT 4.0 on the Alpha platform. On this platform, License Use Runtime functions are available only from the command line.

Windows NT 4.0 Server, Terminal Server Edition, on the Alpha platform. On this platform, License Use Runtime functions are available only from the command line.

For information, see "License Use Runtime Platforms" on page 4 and Chapter 3, "Installing License Use Runtime" on page 47.

#### New Versions and Releases of Operating Systems

Support for previously supported operating systems has been extended to include the following new versions and releases:

HP-UX 11.0 32-bit  
SGI IRIX 6.5  
Sun Solaris 2.7

For information, see "License Use Runtime Platforms" on page 4 and Chapter 3, "Installing License Use Runtime" on page 47.

### Specific Changes

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

## Summary of Changes

### **Custom Configuration**

Vendors can now offer combinations of products, tailored to the needs of each user, under a single custom configuration license.

A section has been added to Chapter 3, Installing License Use Runtime that introduces the concept of custom configuration.

A section has been added to Chapter 6, Hints and Tips that suggests how you might better manage a custom configuration.

A section has been added to Chapter 7, Troubleshooting that suggests what you might do should you have a problem with a custom configuration.

### **Installation on Windows NT**

Installation and uninstallation of License Use Runtime on Windows NT (on the x86 and Alpha platforms) use InstallShield 5.1. See Chapter 3, "Installing License Use Runtime" on page 47.

On Windows NT and Windows Terminal Server, after installing License Use Runtime, you need not shut down and restart your machine. See "Installing License Use Runtime Interactively" on page 51.

### **Uninstallation Enhancements on Windows NT**

Uninstallation of License Use Runtime on Windows NT has been enhanced to remove definitions of system and user variables and NT LUM services. The I4\_INSTALL\_DRIVE and I4\_LANG variable are removed. LUM entries in the BOOKSHELF, HELP, IPF\_PATH32, NLSPATH, and PATH variables are removed. See Chapter 3, "Installing License Use Runtime" on page 47.

### **License Creation Tool Enhancement**

The License Creation Tool enables you to create licenses whose start date is one day earlier than the date the tool is run. This makes it possible for licenses to be used immediately in any time zone. See "i4lct - License Creation Tool" on page 164 for details.

### **Editing a Nodelock File**

A new appendix, Appendix B, "Using the Nodelock File" on page 209, explains how to edit and use a nodelock file.

### **List of Features and Functions Added in Version 4**

A new appendix, Appendix C, "Features and Functions Added in Version 4" on page 211, lists the features and functions added in Version 4. Do not use obsolete commands or APIs, which are supported for backward compatibility, with these newer features and functions.

---

## Chapter 1. Introduction to License Use Runtime

License Use 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:

The License Use Management Application Developer's Toolkit contains the tools that are needed to implement licensing technology in an application program (called *license-enabling* the application). To do the enablement, vendors code API calls in their products and embed the code that services the API calls. The products thus become *license-enabled*. Vendors can license-enable C-language programs, Java applications, and Java applets.

The Application Developer's Toolkit offers the vendor great flexibility in:

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

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

License Use Management Runtime (License Use Runtime) contains the tools that are needed in an end user environment to manage licenses and to get up-to-date information about license usage. The License Use Runtime software is free of charge and is available for download from the IBM License Use Management (LUM) 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 at a Glance

### 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 objective of License Use Runtime is to control the use of licenses in a customer's environment.

(Note that the term *license* does not refer to the license agreement that governs use of and rights to a product.)

In the license-enabling process, the vendor can:

- Select among various types of licenses (see “License Types” on page 7).
- Decide whether 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 “Password Use Control Levels” on page 6).
- Impose, or allow the user to impose, various types of control over administration of licenses (see “License Policies” on page 12).

Vendors deliver licenses to customers in the form of a *license password*. The password contains an encryption of some terms of the acquisition of the software product. For example, the password may specify:

- How many licenses or concurrent copies of the product the customer can use
- The expiration date of the licenses
- The type of license

License Use Runtime checks that customers have a license that authorizes them to use the product when the product is executed, not when it is installed.

Depending on the terms for software product acquisition, vendors can implement licenses in two fundamental ways: *node-locked* licenses and *network* licenses.



## License Use Management at a Glance

A nodelocked license is stored on the workstation where the license-enabled product is installed, for the exclusive use of that node.

With network licenses, you set up a client/server configuration for License Use Management. Many License Use 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, License Use Runtime at the license server determines whether a license is available.

License-enabled Java applications and applets must have network licenses. A Web server machine, rather than the end user machine where the application or applet runs, serves as the network license client. See “Planning for Java Applications and Applets” on page 40.

License Use Runtime includes an administration tool, called the *Basic License Tool*, which manages both nodelocked and network licenses on all the license servers in your network. 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. For more information about what the Basic License Tool does, see “License Administration Tool” on page 18.

## License Use Runtime Platforms

---

### License Use Runtime Platforms

Table 1 shows which platforms License Use Runtime supports, and how to get the License Use Runtime code:

---

*Table 1. License Use Runtime Platforms*

---

<b>AIX 4.3.3</b>	License Use Runtime 4.5.5 base code is part of the base operating system, and is installed on every machine when the operating system is installed. Optional packages and filesets can be installed from the AIX installation media. Alternatively, you can install Version 4.5.5 from the License Use Management Version 4.5.5 CD-ROM or from the product package downloaded from the IBM LUM Web site <a href="http://www.software.ibm.com/is/lum">http://www.software.ibm.com/is/lum</a> .
<b>AIX 4.3.2</b> <b>AIX 4.3.1</b> <b>AIX 4.3.0</b>	On AIX 4.3.2, License Use Runtime 4.5.0 base code is part of the base operating system, and is installed on every machine when the operating system is installed.  On AIX 4.3.1, License Use Runtime 4.0.1 base code is part of the base operating system, and is installed on every machine when the operating system is installed.  On AIX 4.3.0, License Use Runtime 4.0 base code is part of the base operating system, and is installed on every machine when the operating system is installed.  Optional packages and filesets can be installed from the AIX installation media. To upgrade to License Use Runtime Version 4.5.5 without upgrading to AIX 4.3.3, download the code from the IBM LUM Web site <a href="http://www.software.ibm.com/is/lum">http://www.software.ibm.com/is/lum</a> .
<b>AIX 4.1</b> <b>AIX 4.2</b>	The iFOR/LS license management product is part of the base operating system in AIX 4.1 and 4.2, and is installed on every machine when the operating system is installed. To upgrade to License Use Runtime Version 4.5.5, download the code from the IBM LUM Web site <a href="http://www.software.ibm.com/is/lum">http://www.software.ibm.com/is/lum</a> .
<b>Windows NT 4.0 (x86)</b> <b>Windows NT 4.0 Alpha</b> <b>Windows NT Server 4.0,</b> <b>Terminal Server Edition (x86)</b> <b>Windows NT Server 4.0,</b> <b>Terminal Server Edition Alpha</b> <b>Windows 98</b> <b>Windows 95</b> <b>OS/2 Warp Version 4</b> <b>Sun Solaris 2.6 and 2.7</b> <b>HP-UX 10.20 and 11.0</b> <b>Silicon Graphics IRIX 6.3, 6.4, and 6.5</b>	License Use Runtime Version 4.5.5 is normally incorporated into the license-enabled product. Otherwise, it is available for download from the IBM LUM Web site <a href="http://www.software.ibm.com/is/lum">http://www.software.ibm.com/is/lum</a> .

---

## License Passwords

On AIX and OS/2, License Use Management Version 4.0 replaced previous License Use Management releases. On Windows, HP-UX, IRIX, and Solaris, Version 4.0 was the first IBM License Use Management release.

---

### License Passwords

A license password (or *license key*) is an encrypted character string that specifies the characteristics of the license. This information, determined by the vendor, includes:

- The specific number and type of license contained in the password
- The date when the licenses become active
- The date when the licenses expire

Vendors can create two types of password: *simple* and *compound*.

A simple password, once enrolled on a license server, represents one or more licenses that the license server can grant when an end user starts the product.

A compound password, once enrolled on a license server, is a single password from which you can extract multiple simple passwords. Each extracted simple password represents one or more licenses. The compound password is a means of:

- Efficiently distributing multiple licenses from the vendor to the customer.
- Distributing licenses to different license servers, when required. The compound password must be installed on a specific license server. Extracted passwords can be distributed as required to other license servers that are not specified in the compound password.
- Providing a sales representative with a set of licenses that the representative can distribute to different customers.

A compound password contains an expiration date that the vendor sets. The duration of extracted licenses cannot be longer than the time remaining before the compound password expires.

The vendor includes the password, along with other information about the application, in the *enrollment certificate file* (ECF).

Figure 1 on page 6 summarizes the relationship among the license, the license password, the compound password, and the enrollment certificate file.

## Password Use Control Levels

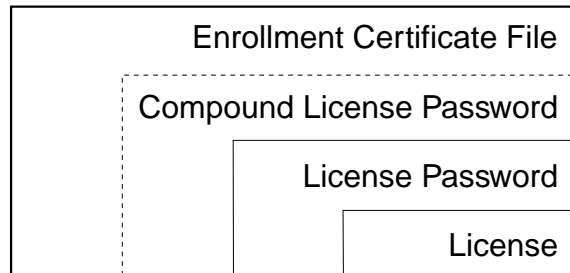


Figure 1. Licensing Concepts Summary

---

## Password Use Control Levels

Vendors can enable their products according to either of the following predefined use control levels:

- Vendor-Managed Use Control (VMU)
- Customer-Managed Use Control (CMU)

## Vendor-Managed Use Products

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

When you acquire licenses for a vendor-managed use product, the product vendor will ask you to supply the unique identifier (target ID) of each machine 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. The password, tied to the specified workstation, cannot be used on another workstation. You must also supply the number of licenses you want to acquire. The vendor uses this information to create the password that you use to install and activate the licenses.

Vendors typically ship a vendor-managed use product with a simple password. They can also use compound passwords for this purpose. You can extract and distribute a limited number of licenses from the compound password, up to the maximum the vendor set in the compound password.

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 machines 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 machine. 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

## License Types

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, License Use Runtime supports *customer-managed use* 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.

License Use 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 License Use 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 allows the use of a product on the particular machine for which the license was created for as long as the license remains valid. Vendors typically use nodelocked licenses for standalone, rather than client/server, applications.

A vendor who is enabling a product that uses nodelocked licenses can choose between two license-enabling models: *non-runtime-based* and *runtime-based*.

If the vendor chooses non-runtime-based enabling, the license-enabled product itself, rather than License Use Runtime, manages use of the nodelocked license. The password for such a product is stored in a file that is called the *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 machine. You interact with the nodelocked license server through the Basic License Tool. It enables you to view and update information about the nodelocked licenses on the machine and get reports about their use.

See “License-Enabling Models” on page 15 for more information about license-enabling models.

## License Types

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 machine. 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



As with a simple nodelocked license, 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 machine during the period the license is valid. 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. The vendor designs the product so that when all concurrent nodelocked licenses for the product are in use, a user can request an available use-once license.

## License Types

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 exactly like concurrent nodelocked licenses, except that at any time, you can change them into per-seat licenses (see “Per-Seat Licenses” on page 11 and “Switching from Per-Server to Per-Seat Licenses” on page 14).

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 machine 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 will probably 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 machine, are stored on a network license server and shared among multiple network license clients.

## License Types

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 Types

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. The vendor designs the product so that when all concurrent licenses for the product are in use, a user can request an available use-once license.

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.

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, 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 9.)

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

## License Policies

---

### 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 *try-and-buy*, *multiuse rules*, *product wait queues*, and *license annotation* license policies.

#### Try-and-Buy Policy



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, 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, or node, a second invocation of the product does not require a second license. 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.

- 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.

For information about how to implement multiuse rules, see “Defining Rules for Multiple-Use Concurrent Licenses” on page 174.

#### Product Wait Queues

Some license-enabled products with concurrent licenses may use wait queues.

When a user invokes such a product, and there are no concurrent licenses currently available, the product can be enabled to ask if the user wants to wait for a license. If

## License Policies

the user responds affirmatively, the user is added to the wait queue on each License Use Runtime network license server that provides concurrent licenses for the product. User names are added to the wait queues in chronological sequence. When a license becomes available, it is granted to the first user in the queues. The next user in the queues becomes eligible for the next available license.

### 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 either a concurrent network license or a 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.

## Customer-Controlled Policies

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

### Hard Stop/Soft Stop Policy

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.

## License Policies

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*).

Only applications with customer-managed use licenses can be enabled for the hard stop/soft stop policy. Vendor-managed use licenses are always hard stop licenses.

When you enroll a product enabled for hard stop/soft stop, the default is soft stop. You can use the Basic License Tool to change the policy to hard stop and back again at any time.

License Use Runtime keeps track of the soft stop licenses in use, so that you can get meaningful information about license use.

When the soft stop policy is set, License Use 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 licenses 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.

### 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).

See “Scenario 10: Restricting User Access” on page 113 for details.

### 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.

See “Per-Server Licenses” on page 9 and “Per-Seat Licenses” on page 11 for information to help you decide between per-server and per-seat. See “Scenario 8: Switching from Per-Server to Per-Seat Licenses” on page 106 for information about how to perform the switch.

## License-Enabling Models

---

### License-Enabling Models

To summarize, 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 License Use Runtime on the machine 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 License Use Runtime on the machine 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 License Use Runtime on the machine where the product runs and requires some limited configuration on that machine. The licenses are stored on one or more network license servers. When the user at a client starts a licensed program, License Use Runtime at the license server determines whether or not a license is available.

Table 2 on page 16 summarizes the license-enabling models, license types, and license policies.

## Scalable Installation and Configuration

Table 2. License-Enabling Models, License Types, and License Policies

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
Nodelocked* License-Enabled Products (Runtime-Based Enabling)	Simple Nodelocked	Try-and-Buy License Annotation
	Use-Once Nodelocked	User Access Restriction License Annotation
	Concurrent Nodelocked, Per-Server	Hard Stop/Soft Stop Multiuse Rules User Access Restriction License Annotation
Network License-Enabled Products	Concurrent	Hard Stop/Soft Stop Multiuse Rules User Access Restriction Product Wait Queues* License Annotation 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

**Note:** \* This applies only to C-language applications.

## Scalable Installation and Configuration

License Use Runtime consists of separate installable components, so that you can install exactly what you need on each workstation.

For example, in AIX 4.3.3, all required License Use Runtime components are automatically installed on every AIX workstation as part of AIX 4.3.3 installation. You can install optional components, such as the graphical user interface, either when you install AIX or later.

On OS/2 and Windows, you can select the appropriate components depending on the role the workstation is to play in your licensing environment. On OS/2, Windows NT (x86), Windows NT Alpha, Windows Terminal Server (x86), Windows Terminal Server

## License Creation Tool

Alpha, Windows 95, and Windows 98, there are components for runtime, communications, and online documentation. On OS/2, there is also a component for namespace binding support. Vendors can, optionally, incorporate the communications component 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 should download and install License Use Runtime.

When you configure License Use Runtime, the configuration tool recognizes which components are installed and presents only the options consistent with the installed component.

The configuration tool, for configuring License Use Runtime license servers and clients, has a command-line interface on all platforms, a graphical user interface on AIX, Windows, and OS/2, and an interactive script interface on all UNIX platforms.

---

### License Creation Tool

License Use 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.

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

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.

## License Administration Tool

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 164.

---

## License Administration Tool

License Use Runtime includes a license administration tool, which is called the *Basic License Tool*.

The Basic License Tool has a command-line interface on all platforms, and a graphical user interface on AIX, OS/2, Windows NT (x86), Windows NT Server Terminal Server Edition (Windows Terminal Server) (x86), Windows 95, and Windows 98. 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 license server database; update information about existing customer-managed use licenses.

### **Display information**

Display a notebook of information about the licenses that are 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 machine, 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 machines. 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 Administration Tool

**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 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.

## High-Availability

### 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 and, by default, to customer-managed products. You can change that threshold, and you can also set a separate threshold for each customer-managed 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.

---

## 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 machine. However, only AIX, HP-UX, IRIX, Solaris, Windows NT, Windows NT Alpha, and Windows Terminal Server network license servers can be members of a cluster: no OS/2, 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.

## Working with Licensed Products

See “Planning Clusters” on page 41 for planning information, and “Scenario 11: Creating and Administering a Cluster” on page 114 for an example.

---

### Backup Procedure

On license servers, there is an automatic backup procedure for License Use Runtime databases and files.

---

### Working with Licensed Products

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

- License Use Runtime is installed properly.
- License Use Runtime is configured properly.
- A Web server machine is set up properly for Java applications and applets.
- 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 special server, which is called the *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.

Some of the uses of the central registry are:

- All the per-seat licenses in the licensing environment are installed on the central registry.
- The list of application clients to which per-seat licenses have been granted is maintained in the central registry.
- Soft stop licenses are tracked in the central registry.
- The high-water mark is recorded in the central registry.
- Reserved licenses that have not yet been granted to a user are kept in the central registry.

## Working with Licensed Products

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

Figure 2 shows what happens when an end user invokes an application with nodelocked licenses for which the vendor chose non-runtime-based enabling. The licenses must be simple nodelocked licenses.

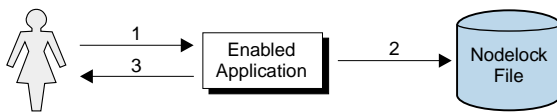


Figure 2. Using a Nodelocked License (Non-Runtime-Based Enabling)

- 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 209.

## Working with Licensed Products

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

Figure 3 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.

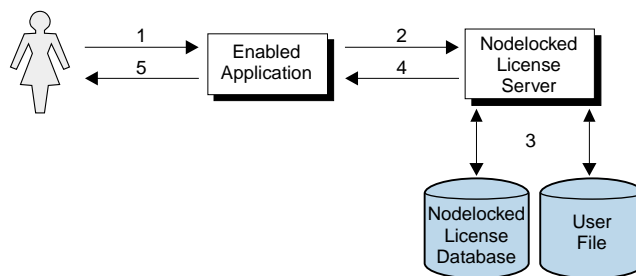


Figure 3. Using a Nodelocked License (Runtime-Based Enabling)

- 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 machine 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 Licensed Products

### Working with Use-Once Licenses

Figure 4 shows what happens when an end user invokes a C-language application with use-once licenses.

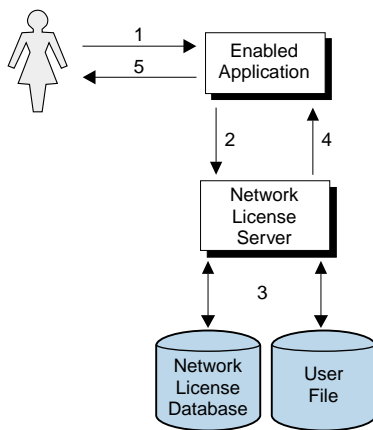


Figure 4. Using a Use-Once License for a C-Language Program

- 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 5 on page 25 shows what happens when an end user invokes a Java application or applet with use-once licenses. The primary difference between usage of use-once licenses for C and Java programs is that in the Java case, the Web server machine, rather than the end user's machine, serves as the network license client.

## Working with Licensed Products

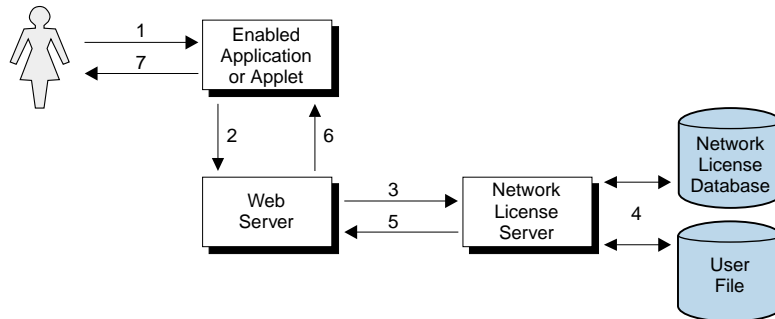


Figure 5. Using a Use-Once License for a Java Application or Applet

- 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 Licensed Products

### Working with Concurrent Licenses

Figure 6 shows what happens when an end user invokes a C-language application with concurrent licenses.

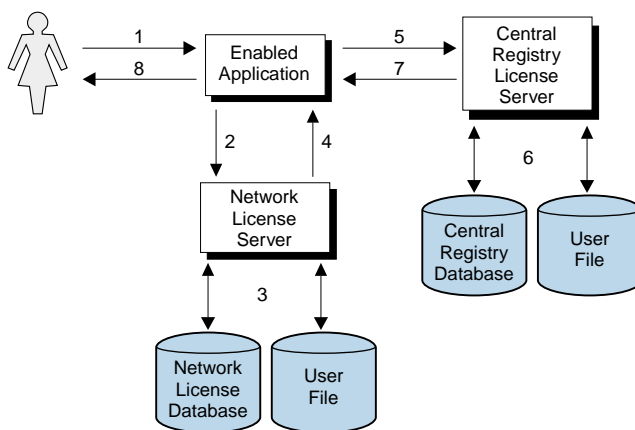


Figure 6. Using a Concurrent License for a C-Language Program

- 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 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 7 on page 27 shows what happens when an end user invokes a Java application or applet that has concurrent licenses. The primary difference between usage of concurrent licenses for C and Java programs is that in the Java case, the Web server machine, rather than the end user's machine, serves as the network license client.



## Working with Licensed Products

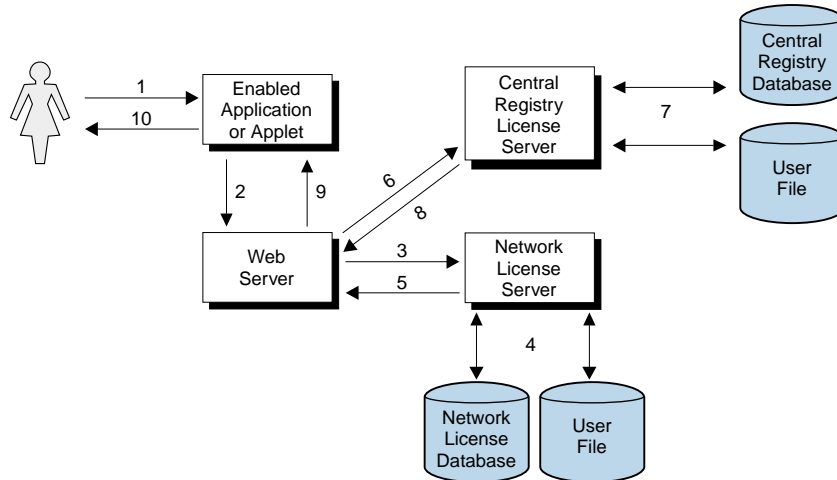


Figure 7. Using a Concurrent License for a Java Application or Applet

- 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 Licensed Products

### Working with Reservable Licenses

Figure 8 shows what happens when an end user invokes a C-language application with reservable licenses.

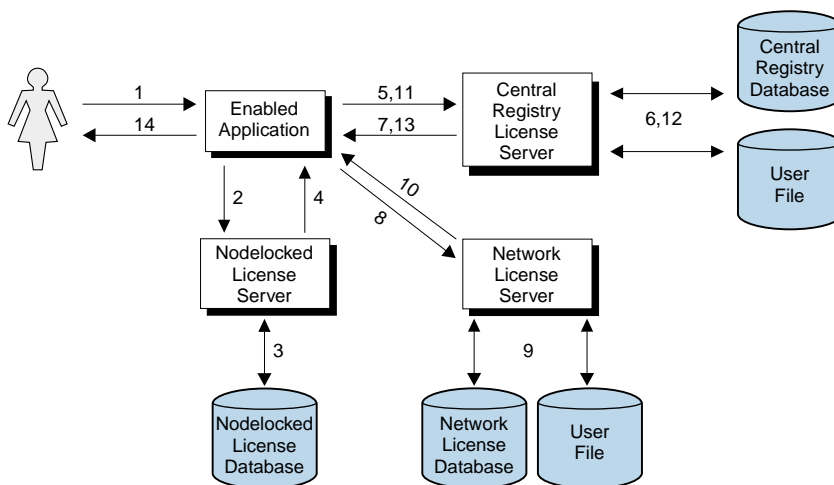


Figure 8. Using a Reservable License for a C-Language Program

- 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 machine, 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.

## Working with Licensed Products

- 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 9 shows what happens when an end user invokes a Java application or applet with reservable licenses. The primary differences between usage of reservable licenses for C and Java programs are that in the Java case:

The Web server machine, rather than the end user's machine, serves as the network license client.

Reserved licenses, when granted, are not moved to the nodelocked license server.

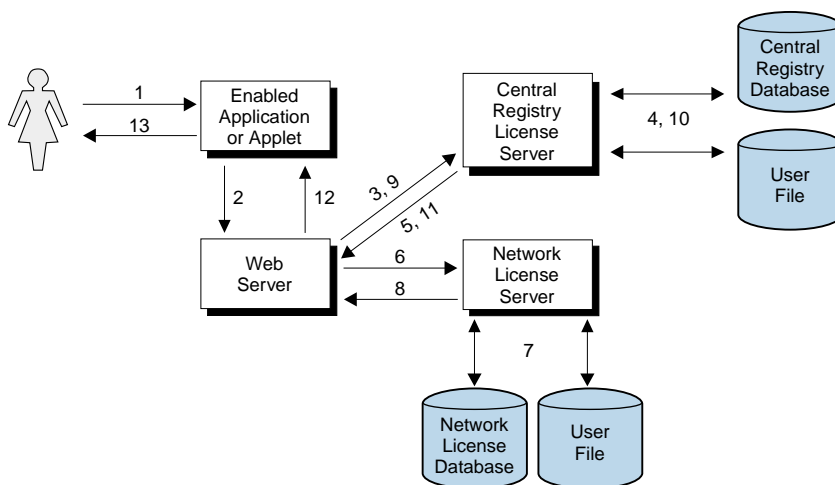


Figure 9. Using a Reservable License for a Java Application or Applet

- 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.

## Working with Licensed Products

- 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 for anyone.
- 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 Licensed Products

### Working with Per-Server Licenses

Figure 10 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.

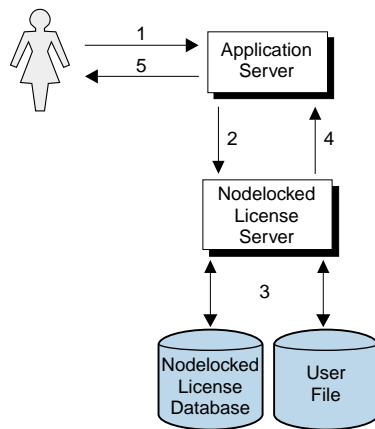


Figure 10. Using a Per-Server License

- 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 Licensed Products

### Working with Per-Seat Licenses

Figure 11 shows what happens when an end user invokes a C-language application with per-server/per-seat licenses when per-seat has been enabled. In the figure, the application server is license-enabled.

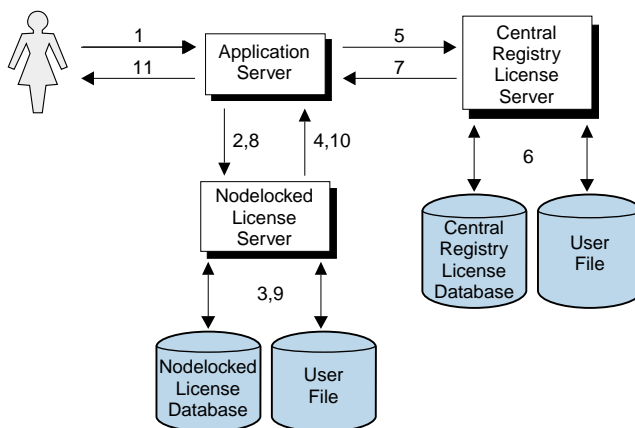


Figure 11. Using a Per-Seat License for a C-Language Program

- 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 machine.
  - 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.

## Working with Licensed Products

- 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 12 shows what happens when an end user invokes a Java application or applet with per-server/per-seat licenses when per-seat has been enabled. The primary differences between usage of per-seat licenses for C and Java programs are that in the Java case:

The Web server machine, rather than the end user's machine, serves as the network license client

When a per-seat license is granted no shadow copy is stored on the nodelocked license server.

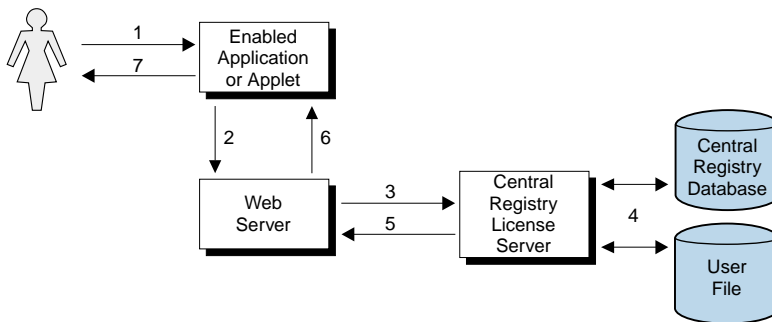


Figure 12. Using a Per-Seat License for a Java Application or Applet

- 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.

## Working with Licensed Products

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 Your Network Licensing Environment



This chapter is for the administrator who is setting up an environment to allow multiple client machines to 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 machine to administer licenses on other machines

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 License Use Runtime in large networks or across subnetworks.

The decisions you need to make include:

- How many network license servers you will set up

- Which machines, 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

- Whether all servers will serve all clients, or whether you will set up independent groups of servers and clients

- Which machine will be the central registry (if required)

- Whether you will make use of License Use Runtime high-availability licensing; 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 machine the network and nodelocked licenses on other machines

- Which machines 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.

## NCS

In designating machines 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. Machines 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.

A computer that is already acting as a file server may well be a good choice, because it satisfies these criteria.

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 37 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 which are included in the License Use Runtime components.

The Remote Procedure Call (RPC) Runtime Library of NCS 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

## Direct Binding

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 67).

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

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

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

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

**Note:** Only TCP/IP is available for the Windows NT Alpha operating system.

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 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 machine configured only as a nodelocked license server. When you configure a nodelocked license server, you

## Direct Binding

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 13 shows a simple 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 three license servers.

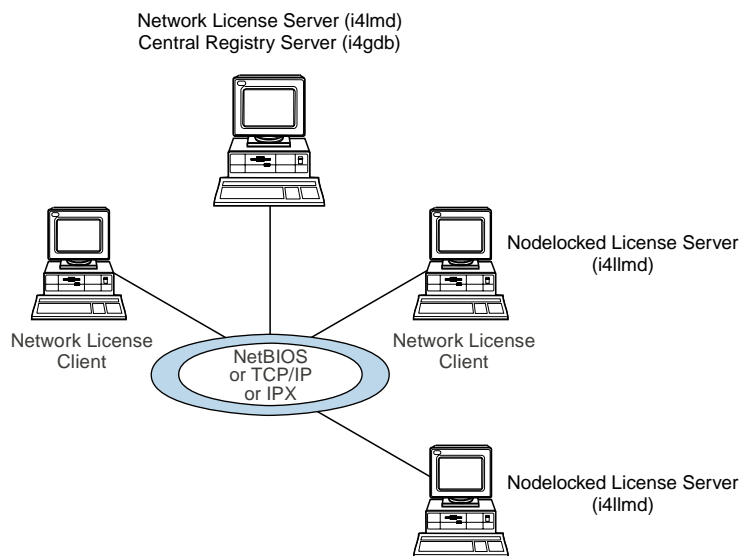


Figure 13. Direct Binding with Network License Servers and Nodelocked License Servers. (Only TCP/IP is available under the Windows NT Alpha operating system.)

Figure 14 on page 39 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. The Java-enabled applet or application communicates with Server A by means of the Hypertext Transfer Protocol (HTTP), if necessary being protected by a firewall.

## Direct Binding

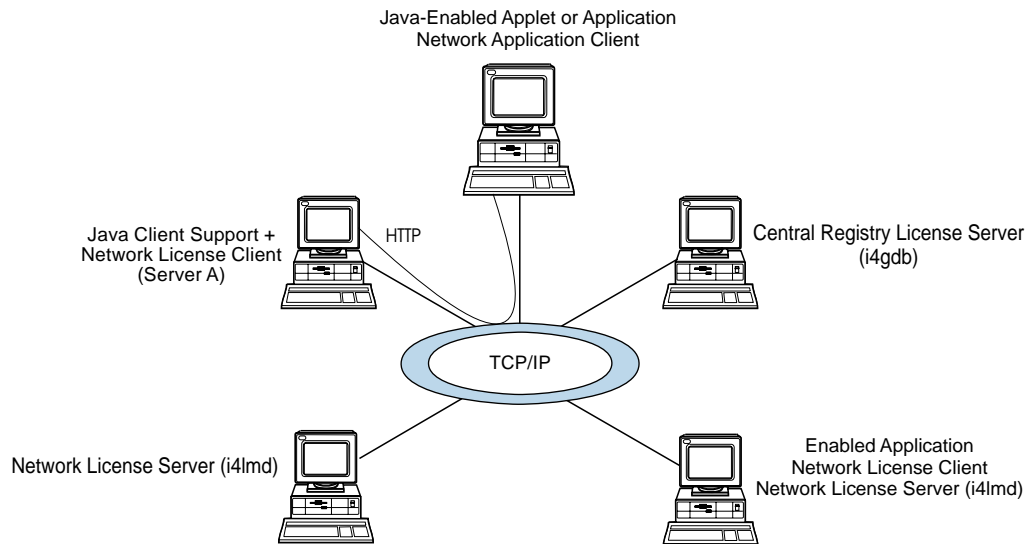


Figure 14. Direct Binding with Java Client Support

Before you begin configuring machines 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 67. 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 machines that will not actually function as servers. If there are any extra machines 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 for Java Applications and Applets

---

### 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. There must be one and only one central registry subsystem running in a direct binding servers list. This ensures that the data is accurate and complete.



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 machine 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 Applications and Applets

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

On the Web server machine, you must:

Run one of the following operating systems:

- AIX 4.3.1, 4.3.2, or 4.3.3
- Windows NT 4.0 (x86 processor)
- OS/2 Warp 4.0
- Solaris 2.6 (with the native threads package) or 2.7

Install a Web server and IBM WebSphere. For details about software requirements, see "Installing LUM Java Client Support" on page 63.

Install License Use Runtime Communications component Version 4.5.5 and LUM Java Client Support.

Configure the machine as a network license client, to communicate with network license servers where the licenses for the Java applications and applets are stored.

On each machine where license-enabled Java applications run, in the Java home directory, you must create a file named `LicenseClient.properties`. The contents of this file must be `url=http://hostname`, where `hostname` is the TCP/IP hostname of the Web server machine. This file identifies the Web server to which license requests are to be directed.

## Planning Clusters

---

### Planning Clusters

To take advantage of License Use Runtime 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 machine. However, only AIX, HP-UX, IRIX, Solaris, Windows NT, Windows NT Alpha, and Windows Terminal Server network license servers can be members of a cluster: no OS/2, 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.



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.

### 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 for as long as it exists.

When you create a cluster (using the GUI or the command-line interface; see “Scenario 11: Creating and Administering a Cluster” on page 114), 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

## Planning Clusters

If you want to replace a server machine 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.



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 3 on page 43.

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.



## Planning Clusters

Table 3. 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 & Running for Cluster to Work
3	2	1	2	2
			3	2
			4	3
4	4	2	4	3
			5	3
			6	4
5	4	1	4	3
			5	3
			6	4
6	6	2	6	4
			7	4
			8	5
7	6	1	6	4
			7	4
			8	5
8	8	2	8	5
			9	5
			10	6
9	8	1	8	5
			9	5
			10	6
10	10	2	10	6
			11	6
			12	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 machine.

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.

## Planning Clusters

Table 4 on page 44 shows how the servers are deployed, depending on how many are activated and how many are up and running.

*Table 4. 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

**Note:** \* 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 machines.

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 5 on page 45 shows how the servers are deployed, depending on how many are activated and how many are up and running.

## NetBIOS Socket Server

Table 5. 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

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

### Cluster Membership Considerations

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.

Do not disable remote administration on a server that is part of a cluster. If you do, 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 machine that is configured to communicate with an OS/2 machine through NetBIOS. The OS/2 machine 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 machine is configured as a server that provides licenses to OS/2 clients connected through NetBIOS, in `i4ls.ini` set:

```
HasOS2Clients=yes
```

## Verifying Connections

|                   **2** If your machine needs to bind an OS/2 License Use Management 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

License Use Runtime provides the i4tv tool to verify that license servers are running properly. For a detailed explanation of how to start the tool and how to use it, see Chapter 5, "License Use Runtime Commands" on page 127.

---

## Chapter 3. Installing License Use Runtime

This chapter is intended for:

The software vendor who needs to install License Use Runtime on a Windows NT, Windows NT Alpha, Windows Terminal Server, Windows 95, or Windows 98 machine for use in license-enabling a product

The customer of a license-enabled product, if the vendor did not incorporate the License Use Runtime components into the product

This chapter explains:

The License Use Runtime components you can select (“Selecting License Use Runtime Components” on page 48)

License Use Runtime hardware and software requirements (“Hardware Requirements” on page 49 and “Software Requirements” on page 49)

How to get the License Use Runtime code (“Obtaining the License Use Runtime Code” on page 50)

How to install License Use Runtime interactively, using the installation GUI (“Installing License Use Runtime Interactively” on page 51)

How to install License Use Runtime noninteractively, using a response file (“Installing License Use Runtime Noninteractively” on page 58)

How to install License Use Runtime from a code server (“Installation Using a Code Server” on page 62)

How to uninstall License Use Runtime components (“Uninstalling License Use Runtime” on page 63)

How to install and uninstall LUM Java Client Support on Windows NT (“Installing LUM Java Client Support” on page 63)

---

### Before You Install

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



To install License Use 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 License Use Runtime from another user ID, the installation program will issue the message:

Installation stopped. Only the user administrator can install License Use Runtime.

To determine whether License Use Runtime Version 4.5.5 is already installed on your machine, refer to the following files:

**Communications component**      `c:\ifor\ls\conf\comver`  
**Runtime component**              `c:\ifor\ls\conf\runver`

where *c* is the drive letter.

## Selecting License Use Runtime Components

License Use Runtime consists of these components:

- Runtime
- Communications
- Documentation
- LUM Java Client Support (which has a separate installation procedure)

Before you install on any machine, you must decide which components to install. Your selection of components depends on the role the machine will play in your licensing environment, as shown in Table 6. If your machine will play more than one role in the licensing environment, be sure you install the components required for all the roles. See "Setting Up Your Servers and Clients" on page 67 for information about configuring your machines to act in these roles.

**Attention:** If you are upgrading License Use Runtime from an earlier version, release, or modification level of License Use Runtime for which the Documentation component is installed, either uninstall the old Documentation component or install the new Documentation component. Otherwise, your License Use Runtime documentation will be incorrect for the installed product.

*Table 6. Required Components by Machine Role*

Role of the Machine	License Administration	Required Components
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 NT only)

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

## Software Requirements

### Hardware Requirements

The hardware requirements for the various Windows platforms are shown in Table 7.

Table 7. Hardware Requirements

Platform	Minimum		Recommended	
	Processor	RAM	Processor	RAM
Windows 95 Windows 98	33MHz 486	16 MB	Pentium 100MHz or higher	32 MB
Windows NT (x86) Windows Terminal Server (x86)	Pentium 100MHz	32 MB	Pentium 100MHz or higher	64 MB
Windows NT Alpha Windows Terminal Server for Alpha	Digital Alpha 433A	64 MB	Digital Alpha 433A	64 MB

Before installing License Use Runtime, allocate enough disk space for the components, as shown in Table 8.

Table 8. Disk Space Requirements

Component	Windows NT (x86) Windows Terminal Server (x86) Windows 95 Windows 98	Windows NT Alpha Windows Terminal Server for Alpha
Runtime	10 MB	5 MB
Communications	4 MB	6 MB
Documentation	2 MB	2 MB

### Software Requirements

One of the following operating systems:

- Windows NT 4.0 (x86) with Service Pack 3, 4, or 5 (Service Pack 4 recommended)
- Windows NT 4.0 Alpha with Service Pack 3, 4, or 5 (Service Pack 4 recommended)
- Windows NT 4.0 Terminal Server Edition (x86) with Service Pack 3 or 4 (Service Pack 4 recommended)
- Windows NT 4.0 Terminal Server Edition for Alpha with Service Pack 3 or 4 (Service Pack 4 recommended)
- Windows 95
- Windows 98

For all supported Windows platforms:

## Obtaining the License Use Runtime Code

- 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 License Use Runtime is not yet installed on the machine, 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 customers.

Depending on your operating system or operating environment, one or more of the transport protocols shown in Table 9:

*Table 9. Transport Protocols*

Platform	TCP/IP	NetBIOS	IPX
Windows NT 4.0 (x86)			
Windows NT 4.0 Terminal Server Edition (x86)			
Windows NT Version 4 Alpha			
Windows NT 4.0 Terminal Server Edition for Alpha			
Windows 95			
Windows 98			

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, be sure that the NetBEUI protocol is installed, and be sure you have done the following:

In 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.

## Obtaining the License Use Runtime Code

You can get the License Use Runtime code:

From the product CD-ROM.



## Installing Interactively

You can use the CD-ROM directly at the machine where you are installing License Use Runtime, or you can use it to put the License Use Runtime code images on a code server and then install your target machines from the code server (see “Installation Using a Code Server” on page 62).

By downloading License Use Runtime from the Web. Download the self-extracting EXE file from <http://www.software.ibm.com/is/lum>. Enter the name of the file as a command, using the /d option. This will:

- Extract the License Use Runtime code images, and
- Create the same directory structure that exists on the CD-ROM.

### Deciding How to Install

Before you install, you need to decide:

Whether to install directly at the target machine or to put the License Use Runtime code images on a code server (“Installation Using a Code Server” on page 62) so that multiple users can use them for installation

Whether to install interactively, using the installation GUI (“Installing License Use Runtime Interactively”) or noninteractively, using a response file (“Installing License Use Runtime Noninteractively” on page 58)

---

### Installing License Use Runtime Interactively

To install License Use Runtime interactively, follow these steps:

- 1 In Windows NT, log on as Administrator.
- 2 Shut down any currently running programs, such as virus scanning programs.
- 3 Start the installation program:
  - a If you are installing License Use Runtime on a non-Windows Terminal Server machine, switch to the `\runtime` directory on the CD-ROM drive or to the directory on the disk drive to which you have downloaded the code, and enter:  
**setup**  
Alternatively, to save to a response file the decisions you make during installation, enter:  
**setup -r**  
For more information about response files, see “Saving a Response File During an Interactive Installation” on page 61.
  - b If you are installing License Use Runtime on a Windows Terminal Server machine:
    - 1 Select **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**).

## Installing Interactively

- 3 Select the **Finish** pushbutton.

The Welcome window appears (Figure 15).



Figure 15. Welcome Window

- 4 In the Welcome window, select **Next>**.
- 5 The READ.ME file for License Use Runtime is displayed (Figure 16 on page 53).

## Installing Interactively



Figure 16. 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, OS/2, Solaris, and Windows NT; it is not available on the Windows Terminal Server, Windows NT Alpha, Windows 95, or Windows 98 platforms.

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

## Installing Interactively

- 6 Select **Next>**. The Choose Destination Location window appears (Figure 17).

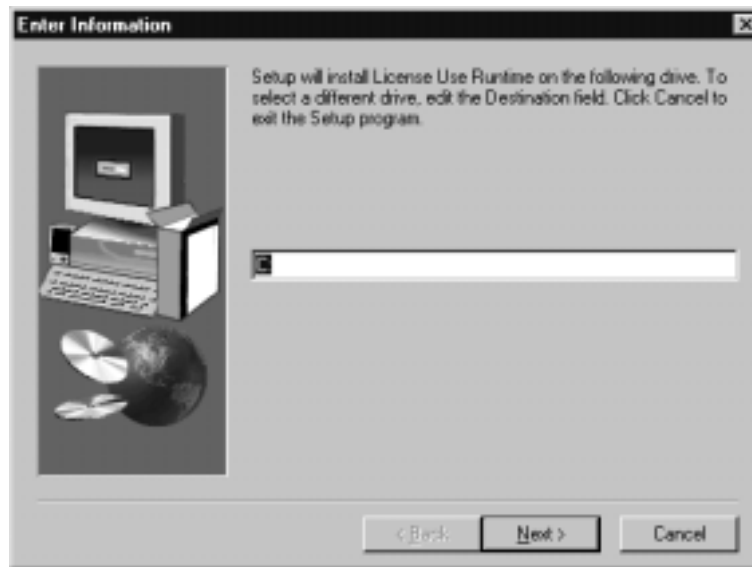


Figure 17. Choose Destination Location Window

Specify a destination drive for License Use Runtime files in the **Destination** field. The default is **C:**.



You can specify the drive only. The product is always installed in the ifor directory.



If Application Developer's Toolkit is installed on the machine, install License Use Runtime on the same drive.

## Installing Interactively

- 7 Select **Next>**. The Select Components window appears (Figure 18).

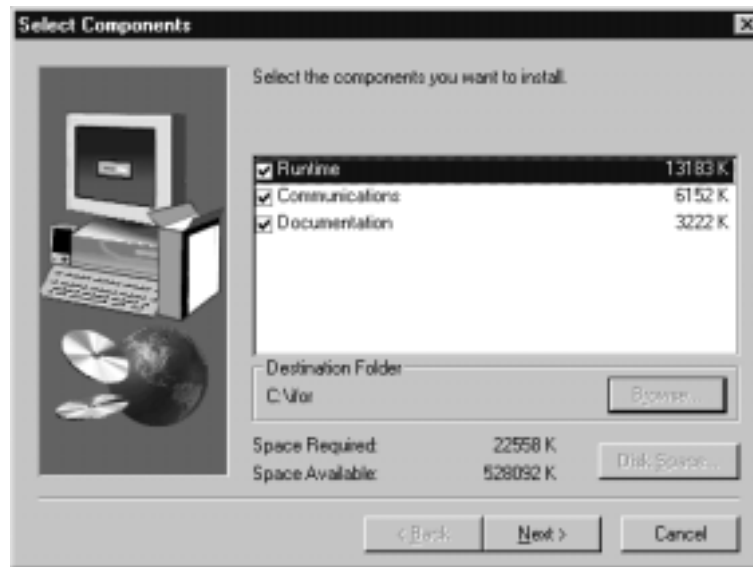


Figure 18. Select Components Window

- 8 Select the License Use Runtime components you want to install.
- 9 Select **Next>**. The Setup window appears, as shown in Figure 19.

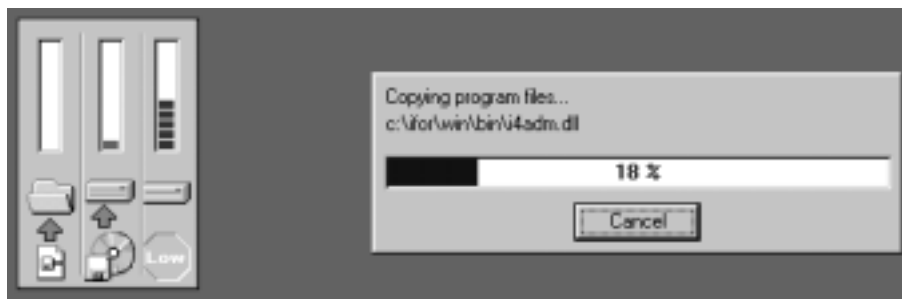


Figure 19. Setup Window

License Use Runtime starts to copy files. You can monitor the progress by watching the progress bar.

If you get the message:

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

## Installing Interactively

where:

*c* Is the installation drive  
*language* Is the installation language

Windows NT has not updated the PATH environment variable, because the sum of the length of the current environment variable plus the length of the License Use Runtime environment variable would exceed 1024 bytes. See Verifying and Updating the PATH Environment Variable for information about how to manually add the License Use Runtime path information.

- 10 The Setup Complete window is displayed. The contents of the window, and the actions you must take, depend on which operating system you are using:

If you have installed License Use Runtime on a Windows 95 or Windows 98 machine, the Setup Complete window looks like that shown in Figure 20.



Figure 20. Setup Complete Window

- a Choose either of these options:
  - Yes, I want to restart my computer now.
  - No, I will restart my computer later.

In either case, you must restart your computer before you can use License Use Runtime.

- b Select **Finish**.

License Use Runtime is installed on your computer, and a folder called License Use Runtime is displayed on the Program menu.

- c Shut down and restart your machine.

If you have installed License Use Runtime on a Windows NT or Windows Terminal Server machine (x86 or Alpha), the Setup Complete window looks like that shown in Figure 21 on page 57.



Figure 21. Setup Complete Window

**a** Select **Finish**.

If you have installed License Use Runtime on a Windows NT machine, setup is now complete. You can start using License Use Runtime immediately. If you have installed License Use Runtime on a Windows Terminal Server machine, continue to step 10b.

**b** If you have installed License Use Runtime on a Windows Terminal Server machine (x86 or Alpha), run the **reglum.cmd** script, to register the License Use Runtime DLLs.

After installation has completed, you must configure License Use Runtime, as explained in Chapter 4, “Getting Started with License Use Runtime” on page 67.

---

### Installing after an Uninstallation

If you uninstall License Use Runtime 4.x.y, your existing license databases are not deleted. They remain in the directory ifor\ls\conf. If you then install License Use Runtime Version 4.5.5, the installation process renames the databases to have an extension of .old. If you want to use the databases with License Use Runtime Version 4.5.5, enter the following sequence of commands:

## Installing Noninteractively

```
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
i4mig -r
i4mig -r -c
i4mig -r -n
```

---

## Installing License Use Runtime Noninteractively

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

You can create a response file in advance, as explained in “Creating a Response File” on page 60. The response file contains your responses to the decisions that are made during an interactive installation. 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.

If you get the message:

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

where:

*c*            Is the installation drive  
*language*    Is the installation language

Windows NT has not updated the PATH environment variable, because the sum of the length of the current environment variable plus the length of the License Use Runtime environment variable would exceed 1024 bytes. See Verifying and Updating the PATH Environment Variable for information about how to manually add the License Use Runtime path information.

- 1 From the CD-ROM drive or the directory where you have downloaded the code, switch to the `\runtime` directory.
- 2 Run the installation program:

If you are installing License Use Runtime on a non-Windows Terminal Server machine, from the CD-ROM drive or the directory to which you have downloaded the code, switch to the `\runtime` directory and enter the **setup -s** command, according to the syntax described in “Using the Installation Command” on page 59.

If you are installing License Use Runtime on a Windows Terminal Server machine:



## Using the Installation Command

- a** Select **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 -s**), according to the syntax described in “Using the Installation Command.”
- c** Select the **Finish** pushbutton.

- 3** If you installed the Runtime component under Windows 95 or Windows 98, shut down and restart the machine.
- 4** If you installed License Use Runtime on a Windows Terminal Server machine, run the **reglum.cmd** script to register the License Use Runtime dynamic link libraries.
- 5** Follow the instructions in Chapter 4, “Getting Started with License Use Runtime” on page 67.

### Using the Installation Command

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

```
setup -s -f1<response_file> -f2<log_file>
```

where:

**-s**

Specify this to execute a silent installation.

If you omit this parameter, the installation starts interactively with all windows displayed.

**-f1**<response\_file>

The full path and name of the response file containing the installation parameters. It is mandatory if you specify the **-s** parameter.

**-f2**<log\_file>

The full path and name of the log file created by the silent installation. If you do not specify it, by default a SETUP.LOG log file is created and stored in the same directory as that of SETUP.EXE. The SETUP.LOG file for a successful silent installation of the *Examp* product is shown below.

## Creating a Response File

```
[InstallShield Silent]
Version=v3. .
File=Log File

[Application]
Name=Examp
Version=3. .

Company=Stirling

[ResponseResult]
ResultCode=
```

To check if the installation was successful, check ResultCode in SETUP.LOG. If ResultCode is 0, the installation was successful; if it has any other value, the installation was unsuccessful.

Note that for -f1 and -f2, you must not leave a blank between the parameter and the path you specify after it.

After the installation is complete:

- 1 If you installed License Use Runtime on a Windows NT machine (x86 or Alpha), go to step 2. Otherwise:
  - If you installed License Use Runtime under Windows 95 or Windows 98, shut down and restart your machine.
  - If you installed License Use Runtime on a Windows Terminal Server, run the **reglum.cmd** script to register License Use Runtime dynamic link libraries.
- 2 Configure your machine as explained in Chapter 4, "Getting Started with License Use Runtime" on page 67.

## Creating a Response File

This section explains two ways to create a response file: editing the sample response file provided with License Use Runtime, and saving the results of an interactive installation to a response file for subsequent use on other machines.

### Editing the Sample Response File

The response file is divided into several sections, each started by a name included in brackets. There are two types of line in each section:

- |                       |   |
|-----------------------|---|
| <b>Comment lines</b>  | Have a semicolon as the first nonblank character on the line.   |
| <b>Response lines</b> | Determine the options and configurations to install on the target system. Response lines have the following syntax:<br><code>keyword=value</code><br>Keywords are case-sensitive. |

## Creating a Response File

An example of how to install License Use Runtime using a response file follows:

```
setup -s -flc:\mydir\setup.iss
```

where **setup.iss** is the response file. This sample file is included in the License Use Runtime package in the same directory as the setup.exe command.

Use this response file to customize License Use Runtime installation. You can change the values of the keywords identified by the string:

```
| ;
```

They are:

### **szText**

The target drive for License Use Runtime. Example:

```
| szText C:
```

### **szDir**

The target drive for License Use Runtime. Repeat here the same value specified for szText.

### **Component-count**

The number of components you want to install. Valid values are from 1 to 3.

### **Component-*n***

The names of the components you want to install. Do not use quotation marks around the component name. Specify as many components as the number you specify in Component-count. For example, if you want to install all three License Use Runtime components, set Component-count to 3, and then:

```
Component- =Runtime  
Component-1=Communications  
Component-2=Documentation
```

If you want to install only the Communications and Documentation components, set Component-count to 2, and then:

```
Component- =Communications  
Component-1=Documentation
```

Do not modify the other keyword values.

## **Saving a Response File During an Interactive Installation**

When you do an interactive installation, you can start the installation by entering the command **setup -r**. If you do so, the installation process saves the decisions you make in a response file called **setup.iss** in the `WINDOWS` directory. You can then use the response file to replicate the same installation on other machines.

## Using a Code Server

---

### 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\language
```

where:

*c* Is the installation drive  
*language* Is the installation language; for example, EN\_US for US English

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

- 1 Go to Settings Control Panel System Environment.
- 2 If necessary, delete unused path information. The License Use 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\language
```
- 4 Press the **Set** or **OK** pushbutton.

---

### Installation Using a Code Server

A *code server* is a workstation that contains the files (CD-ROM images) to be used during the installation.

To install License Use Runtime on a non-Windows Terminal Server machine, 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 License Use Runtime directory name, and type `SETUP`. Type the command without parameters for an interactive installation, or with parameters as indicated in the section "Using the Installation Command" on page 59 for a noninteractive installation.

To install License Use Runtime on a Windows Terminal Server machine:

- 1 Select **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**).
- 3 Select the **Finish** pushbutton.

## Installing LUM Java Client Support

Proceed with the installation as explained in “Installing License Use Runtime Interactively” on page 51 or “Installing License Use Runtime Noninteractively” on page 58. When you have successfully installed License Use Runtime, a message indicates that the installation is complete.

If you have installed the Runtime component of License Use Runtime under Windows 95 or Windows 98, shut down and restart the machine.

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

For non-Windows Terminal Server and for Windows Terminal Server installation, follow the instructions in Chapter 4, “Getting Started with License Use Runtime” on page 67.

---

### Uninstalling License Use Runtime

You can remove License Use Runtime from your computer only in its entirety. You cannot selectively uninstall components.

To remove License Use Runtime from your machine, go to Settings Control Panel Add/Remove Programs and select License Use Runtime.

Note, however, 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.

---

### Installing LUM Java Client Support

LUM Java Client Support is a separately installable component, which you can install only on Windows NT.

#### Before You Install

This section presents disk space and software requirements.

#### Disk Space Requirements

LUM Java Client Support requires approximately 200 KB of disk space.

#### Software Requirements

The following software is required:

One of the following Web servers:

- Lotus Domino Go Webserver 4.6.x
- Netscape Enterprise Server 3.0.1 or 3.5.1; 3.5.1 is recommended
- Netscape FastTrack Server 2.0.1 or 3.0.1
- Microsoft Internet Information Server 2.x, 3.x, or 4.0

## Installing LUM Java Client Support

Either Java Development Kit (JDK) 1.1.4 or 1.1.6, or Java Runtime Environment (JRE) 1.1.4 or 1.1.6. JDK 1.1.6 is shipped with the IBM WebSphere Application Server.

IBM WebSphere Application Server 1.1 or 2.0

The Communications component of License Use Runtime Version 4.5.5

TCP/IP

## Obtaining LUM Java Client Support Code

You can get the LUM Java Client Support code:

From the product CD-ROM.

By downloading LUM Java Client Support from the Web. Download the self-extracting EXE file from <http://www.software.ibm.com/is/lum>. Enter the name of the file as a command, using the /d option. This will:

- Extract LUM Java Client Support code images, and
- Create the proper directory structure.

With an enabled application, if the vendor chose to redistribute the LUM Java Client Support code.

## Installing LUM Java Client Support

To install LUM Java Client Support interactively, follow these steps:

- 1 In Windows NT, log on as Administrator.
- 2 From the CD-ROM drive or the directory where you have downloaded the code, switch to the `\servlet` directory and type **setup**.
- 3 Follow the panels that are displayed. LUM Java Client Support is installed in the directory `APPSERVER_ROOT\Web\LUM\classes\com\ibm\licUseMgmt`, where `APPSERVER_ROOT\Web\` is a WebSphere subdirectory. One file is also stored in `ifor\win\bin`.

## Uninstalling LUM Java Client Support

To remove LUM Java Client Support from your machine, go to Settings → Control Panel → Add/Remove Programs and select License Use Management LUM Java Client Support.







---

## Chapter 4. Getting Started with License Use Runtime

The scenarios in this chapter describe how to set up your License Use Runtime environment and how to manage both nodelocked and network licensed products.

---

### Setting Up Your Servers and Clients

After you install License Use Runtime, you must configure the program on each machine.

You configure using a configuration tool. On platforms other than Windows NT Alpha and Windows Terminal Server Alpha, this tool has a GUI and a command line interface. On the Windows NT Alpha and Windows Terminal Server Alpha platforms it has only a command line interface. The information you provide is saved in a configuration file. When you start License Use Runtime, it uses the information in this file to direct the behavior of the local system in the licensing environment. See Appendix A, "License Use Runtime Configuration File" on page 199 for reference information on the configuration file. The configuration details depend on the role the machine will play in your licensing environment and the types of licenses you need to handle.

### Configuring to Handle Nodelocked Licenses

To handle only products with nodelocked licenses, you need only configure the machine as a nodelocked license server ("Scenario 1: Configuring a Standalone Nodelocked License Server" on page 73) and have the nodelocked license server up and running. Every machine is automatically configured as a nodelocked license server when License Use Runtime is installed. You do not have to do any configuration unless you want to change the default configuration.

If a machine configured as a nodelocked license server is in a network, instances of the Basic License Tool on other machines 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 75), you can run the Basic License Tool on the nodelocked license server machine 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 79) and configure each client as a network license client ("Scenario 4: Configuring a Network License Client" on page 84). If you use products with customer-managed or reservable licenses, you must also configure one server as the central registry ("Scenario 5: Configuring the Central Registry License Server" on page 88).

## Configuration

### Determining the Configuration Required

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

If you know the types of licenses used by the license-enabled products in your environment, Table 10 on page 69 will help you to determine how to configure:

- The license servers for the application

- The machines that will request licenses for the application

Consult the documentation of the license-enabled products for the license types and other information about the enabling that might affect your configuration. In any case, the enrollment certificate file that you receive from the vendor shows the type of license.

## Configuration Scenarios

Table 10. Configuration Required to Support All Types of Licenses

License Type	License Requester	License Server
Simple Nodelocked (Non-Runtime-Based Enabling) <sup>1</sup>	License-Enabled Application	None
Simple Nodelocked (Runtime-Based Enabling) <sup>1</sup>	License-Enabled Application	Nodelocked License Server on the Same Machine as the Application
Use-Once Nodelocked	License-Enabled Application	Nodelocked License Server on the Same Machine as the Application
Concurrent Nodelocked	License-Enabled Application	Nodelocked License Server on the Same Machine as the Application
Per-Server	License-Enabled Application	Nodelocked License Server on the Same Machine as the Application
Use-Once	Network License Client	Network License Server <sup>2</sup> + Central Registry License Server <sup>3</sup>
Concurrent	Network License Client	Network License Server <sup>2</sup> + Central Registry License Server <sup>3</sup>
Per-Seat	Network License Client + Nodelocked License Server	Central Registry License Server <sup>3</sup>
Reservable	Network License Client + Nodelocked License Server <sup>4</sup>	Network License Server <sup>2</sup> + Central Registry License Server <sup>3</sup>

**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 machine (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.

## Configuration Scenarios

### Before You Configure

Before you begin the configuration process, for every machine you are going to configure, you need to decide which roles it will play and how you plan to set up direct binding. You might also decide to override some of the configuration defaults. The actions you can take during configuration are summarized in Table 11. Check the table for all the roles your machine will play, and make all the indicated decisions before you start configuration.

Table 11. 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")				
Change log path or accept default? ("Customizing Log Information")				
Start license servers at system startup (default is no)? ("Automatically Starting License Servers" on page 71)				
Disable remote administration of network license servers (default is no)? ("Disabling Remote Administration" on page 71)				
Disable remote administration of nodelocked license servers (default is no)? ("Disabling Remote Administration" on page 71)				
Set up direct binding ("Configuring Direct Binding" on page 71)				
Prepare a list of nodelocked license servers for remote administration	1			
Have your direct binding servers list ready	1			
Know which machine is the central registry	1			
Change default direct binding ports?	1			

#### Notes:

1. Applicable if you plan to run the Basic License Tool and administer licenses on other machines 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.

## Configuration Scenarios

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 machine. 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. (See "Planning Direct Binding" on page 37.)

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

## Configuration Scenarios

have ready a list of nodelocked license servers whose licenses you want to administer remotely from this machine.

You will enter the hostnames or network addresses of all the servers in the list (other than the nodelocked license server on the local machine, 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 License Use Management Java Client Support on the same machine and want to change the direct binding list:

- 1 Stop License Use Management and the Web server.
- 2 Change the direct binding list.
- 3 Restart License Use Management and the Web server.

### Using the Configuration Tool GUI

On platforms other than Windows NT Alpha, you can use the License Use Runtime GUI to configure your machine. If you use Windows NT Alpha, see “Using the Configuration Tool Command-Line Interface” on page 73.

In the License Use Runtime folder, click on **Configuration Tool**.

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 machine in your licensing environment.



Each of the configuration scenarios in this chapter shows how to configure a machine to play only one role in the licensing environment. If the machine 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.

## Configuring a Standalone Nodelocked License Server

### 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.

See “i4cfg - Configuration Tool” on page 153 for details about the `i4cfg` command.

### 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 License Use Runtime servers. Use “Scenario 2: Configuring a Nodelocked License Server in a Network” on page 75 if you want to run the Basic License Tool and administer licenses on other servers from this machine,

A nodelocked license server can use licenses of the types shown in Table 10 on page 69. By default, every machine 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 machine as a standalone nodelocked license server after installation, perform the following steps:

- 1 Double-click on **Configuration Tool** 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**. The notebook is shown in Figure 22 on page 74.

## Configuring a Standalone Nodelocked License Server

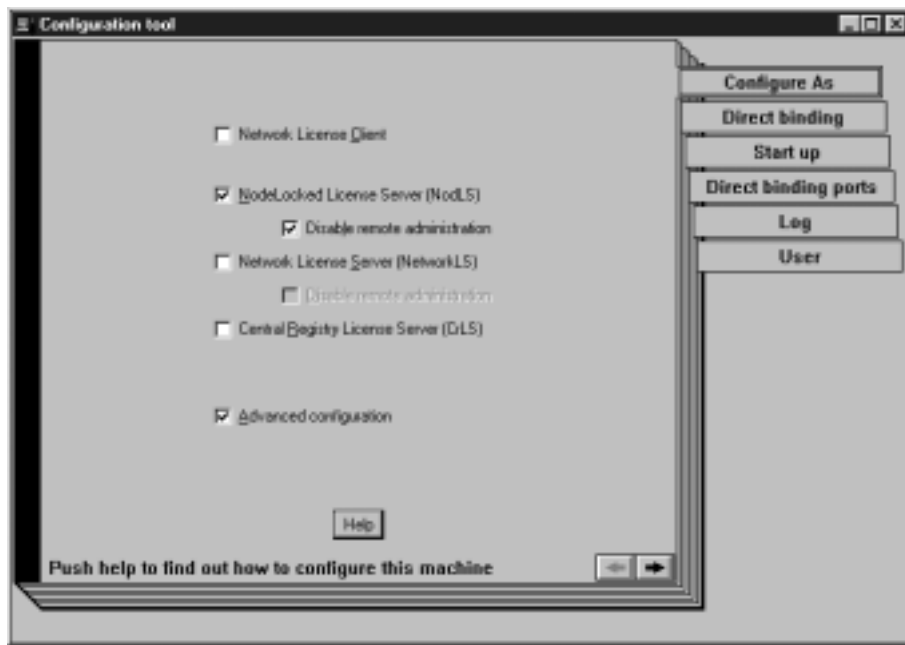


Figure 22. Configuration Tool Notebook - Standalone Nodelocked License Server

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 machine. (This option is available on Windows NT 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 23 on page 75. 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<sub>nn</sub>*, 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.



## Configuring a Nodelocked License Server in a Network

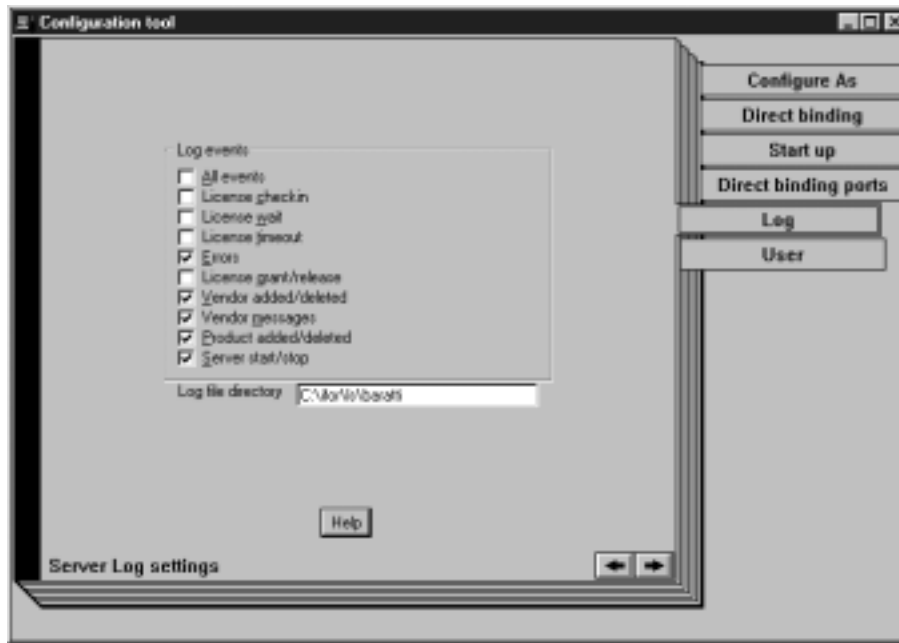


Figure 23. Configuration Tool Notebook - Log Page

- 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 -l 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 machine

Allow instances of the Basic License Tool on other machines to administer nodelocked licenses on this machine

A nodelocked license server can use licenses of the types shown in Table 10 on page 69.

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

- 1 Double-click on **Configuration Tool** in the License Use Runtime folder. The Configuration Tool notebook is displayed.

## Configuring a Nodelocked License Server in a Network

- 2 On the **Configure As** page, select **Nodelocked License Server** and **Advanced Configuration**.
- 3 Leave the **Disable remote administration** box unchecked, to enable instances of the Basic License Tool on other machines to administer nodelocked licenses on this machine.

The completed **Configure As** page is shown in Figure 24.

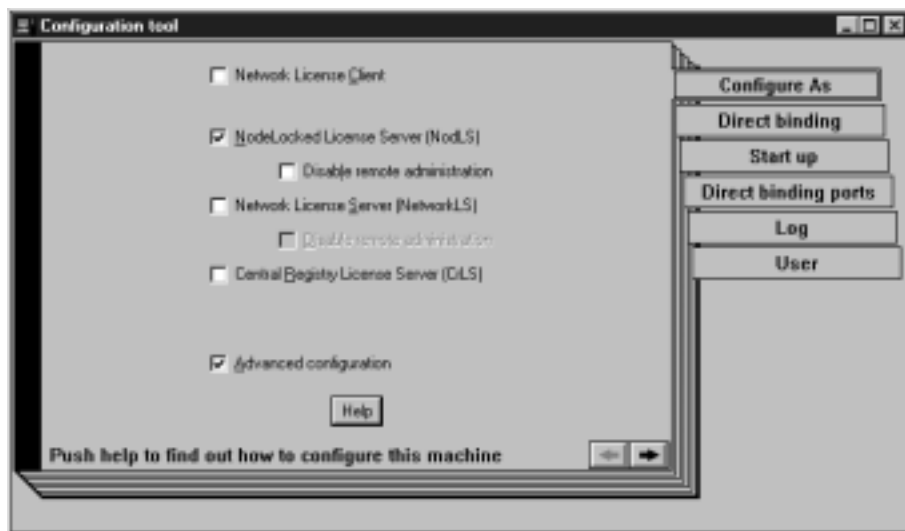


Figure 24. Configuration Tool Notebook - Nodelocked License Server in a Network

- 4 Complete the **Start up** and **Log** pages, as explained in "Scenario 1: Configuring a Standalone Nodelocked License Server" on page 73, if you want to change the defaults.
- 5 Select the **User** tab. The **User** section is displayed, as shown in Figure 25 on page 77.

## Configuring a Nodelocked License Server in a Network

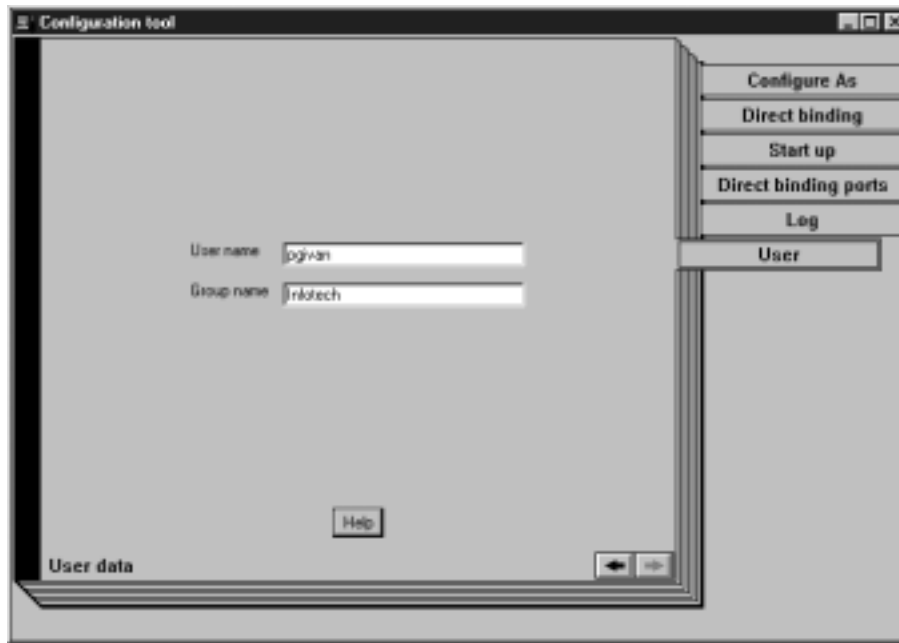


Figure 25. Configuration Tool Notebook - User Section

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 that the License Use Runtime group name is not the same as the Windows NT group name.

License Use Runtime treats the user name as the name of the user who requests and uses licenses on this machine. 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 members of a specific group.

Use the remaining pages of the notebook to establish remote connections between this server and other License Use Runtime license servers. From this machine, 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 26 on page 78.

## Configuring a Nodelocked License Server in a Network

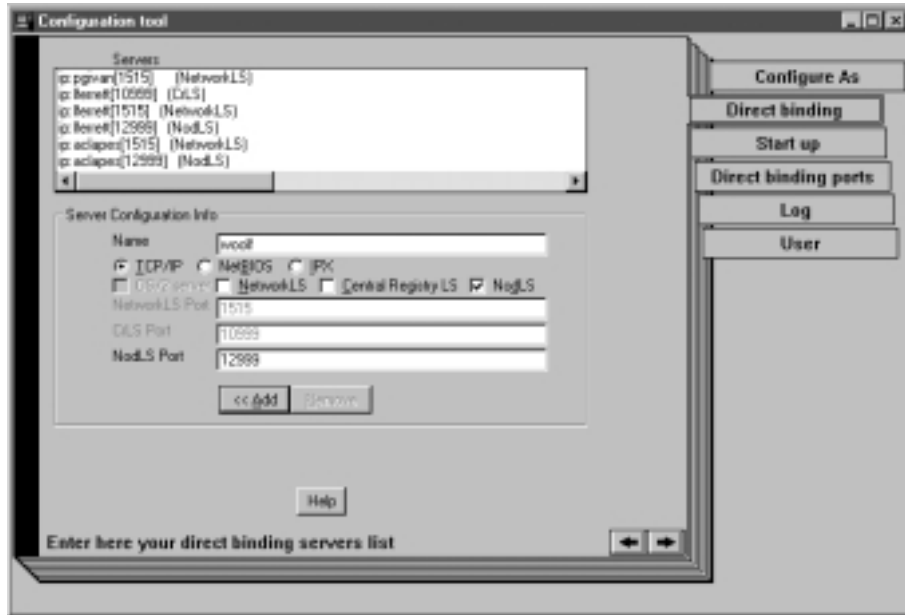


Figure 26. Configuration Tool Notebook - Direct Binding Section

On this page you specify all the network license servers and nodelocked license servers whose licenses you will administer remotely from this machine. 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 machine, 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. On the Windows NT Alpha platform, this is the only choice.
- b In the **Name** field, enter the TCP/IP host name of the machine 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 machine plays in the network.
- d Leave the default values in the **NetworkLS Port**, **CrLS Port**, and **NodLS Port** fields.



For platforms other than Windows NT Alpha, Windows 95, or Windows 98, if NetBIOS is selected, the default values are, respectively, 115, 109, and 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 50.

## Configuring a Network License Server

- 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 To run the configured processes, see “Starting and Listing Your Processes” on page 90.

Now the workstation can be used as a nodelocked license server. The administrator can run the Basic License Tool on the workstation and can administer licenses on remote License Use 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 License Use Runtime as a network license server (in this example, *pgivan*), making it part of a direct binding server list.

When you configure a machine as a network license server, you can use licenses of the types shown in Table 10 on page 69. 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 License Use Runtime:

- 1 Double-click on **Configuration Tool** in the License Use 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 27 on page 80.

## Configuring a Network License Server

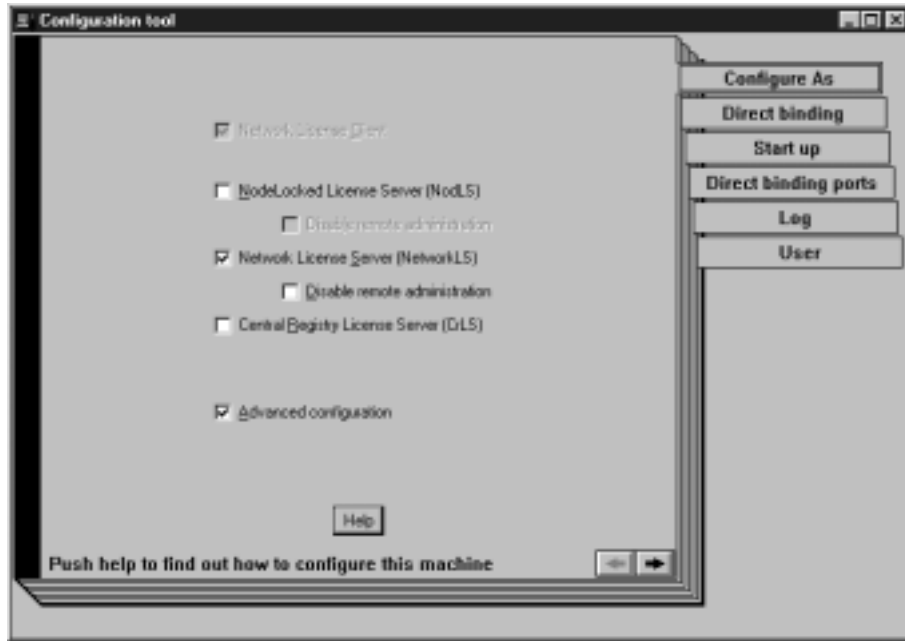


Figure 27. Configuration Tool Notebook - Network License Server

- 3 On the **Start up** page, select **Start services at system startup** to start the network license server when you power on the machine. (This option is available on Windows NT 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 28 on page 81.

The logged events are stored in the files *logdbnn\_*, 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.

## Configuring a Network License Server

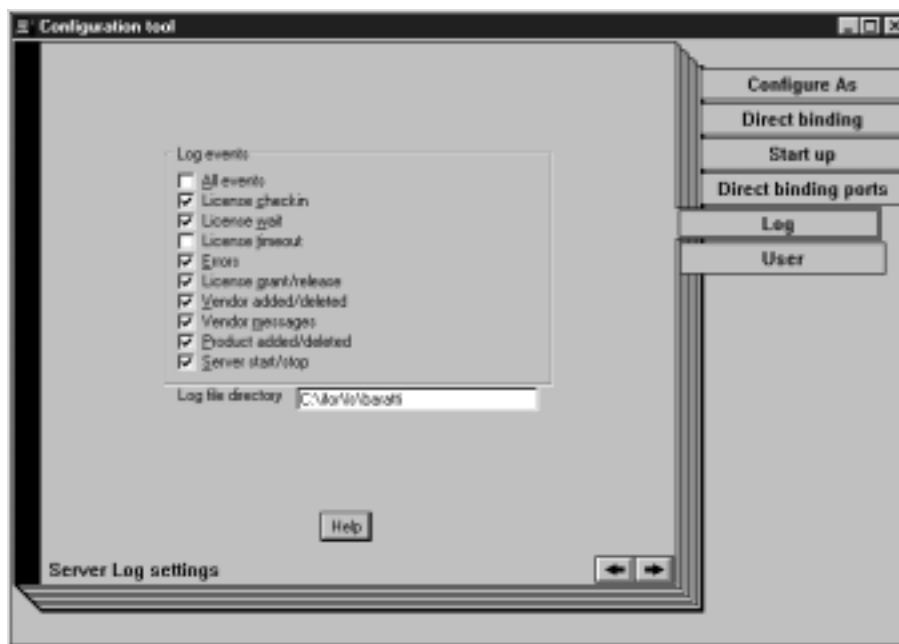


Figure 28. Configuration Tool Notebook - Log Page

**5** Select the **Direct binding** tab.

The Direct binding section is displayed, as shown in Figure 29 on page 82.

## Configuring a Network License Server

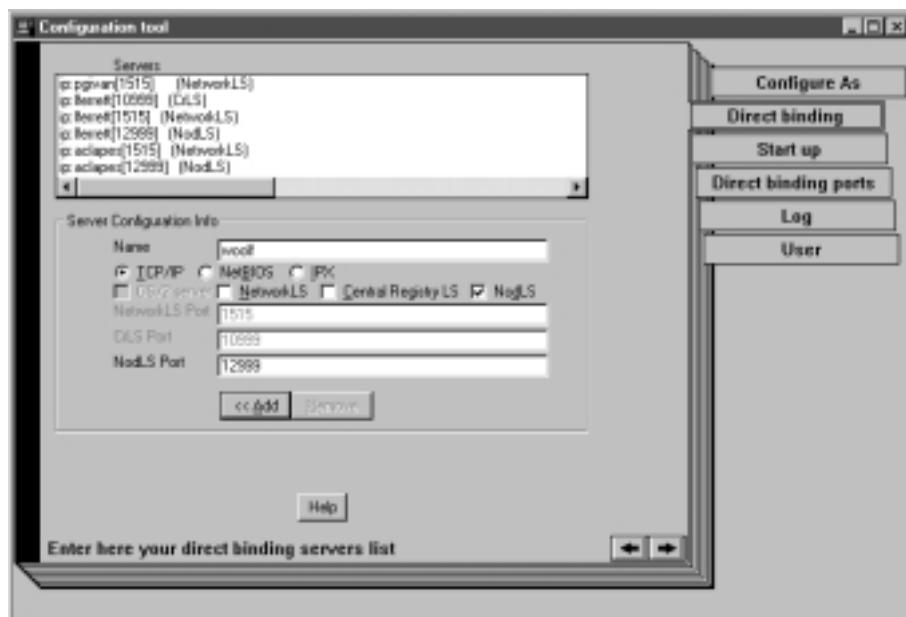


Figure 29. Configuration Tool Notebook - Direct Binding Section

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 machine, be sure you include *both* servers in the list.

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

In addition to specifying the direct binding servers list, use this page to specify any remote nodelocked license servers whose licenses you want to administer from this machine. Do not include the nodelocked license server on this machine. (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. On the Windows NT Alpha platform, this is the only choice.
- b In the **Name** field, enter the TCP/IP host name of the machine 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 machine plays in the network.



## Configuring a Network License Server

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



For platforms other than Windows NT Alpha, Windows 95, or Windows 98, if NetBIOS is selected, the default values are, respectively, 115, 109, and 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 50.

- f Select the <<Add push button 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 30.

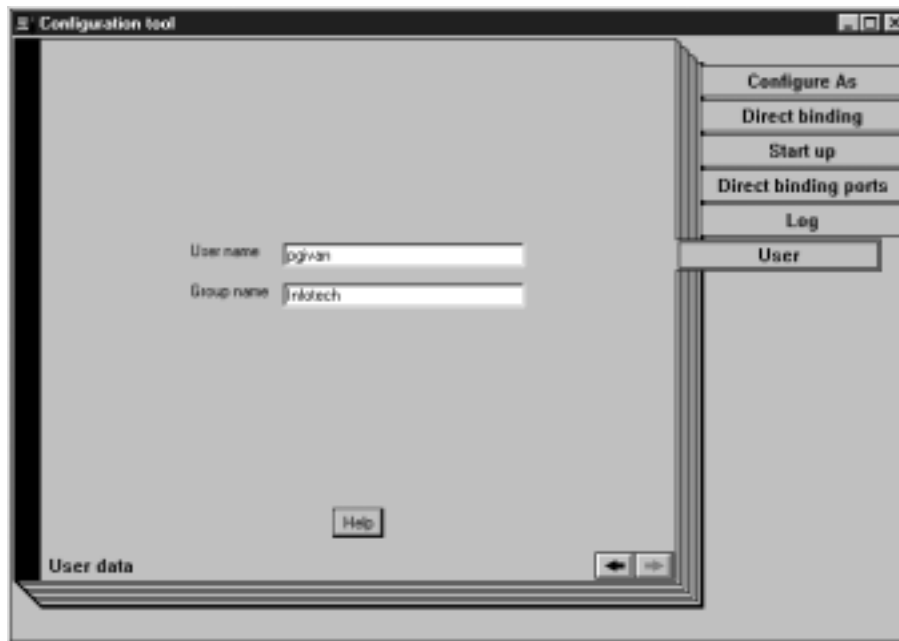


Figure 30. Configuration Tool Notebook - User Section

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 that the License Use Runtime group name is not the same as the Windows NT group name.

## Configuring a Network License Client

License Use Runtime treats the user name as the name of the user who requests and uses licenses on this machine. 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 members of 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 90.

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 a Network License Client

This scenario shows how the administrator or the end user configures License Use Runtime as a client of the network license servers configured in “Scenario 3: Configuring a Network License Server” on page 79 and of the central registry license server configured in “Scenario 5: Configuring the Central Registry License Server” on page 88. Configuring a machine as a network license client makes it possible to use licenses of the types shown in Table 10 on page 69.

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

- 1 Double-click on **Configuration Tool** in the License Use Runtime folder. The Configuration Tool notebook is displayed, as shown in Figure 31 on page 85. Check **Advanced Configuration** to add the **User** tab to the notebook.

## Configuring a Network License Client

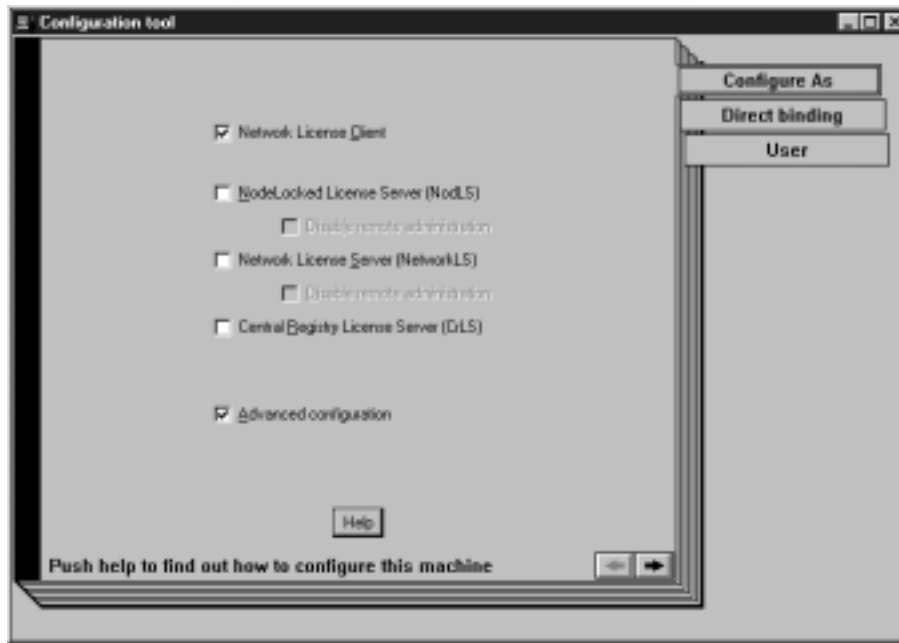


Figure 31. Configuration Tool Notebook - Network License Client

- 2 Select the **Direct binding** tab. The Direct binding section is displayed, as shown in Figure 32 on page 86.

## Configuring a Network License Client

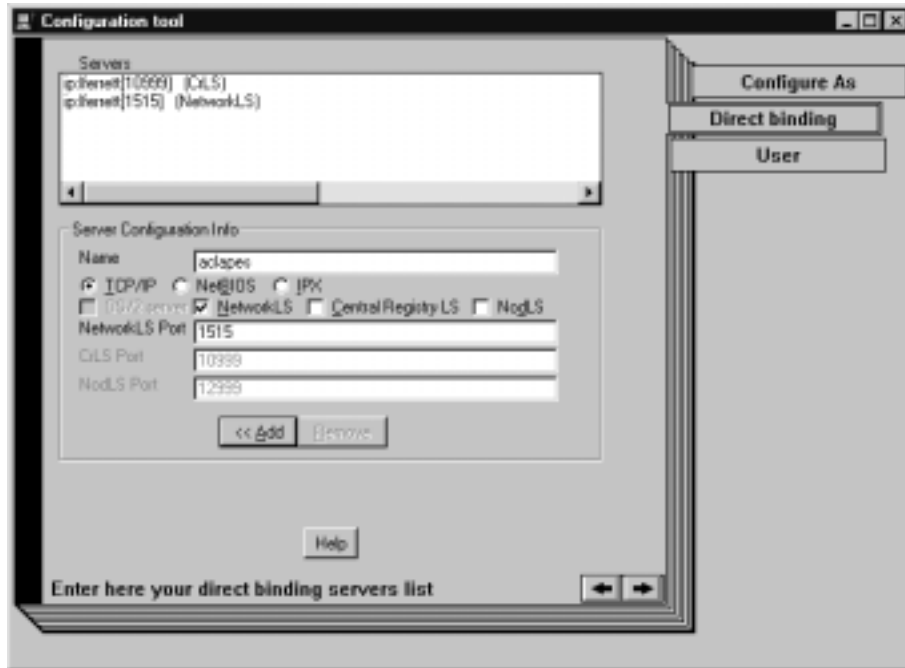


Figure 32. Configuration Tool Notebook - Direct Binding Section

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 37.

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

- a Select the **TCP/IP** radio button. On the Windows NT Alpha platform, this is the only choice.
- 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.



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

## Configuring a Network License Client

If NetBIOS or IPX is selected, the protocol must be configured as explained under "Protocol Configuration Requirements" on page 50.

e Select the <<Add push button to add the server to the **Servers** list.

3 Select the **User** tab. The User section is displayed, as shown in Figure 33.

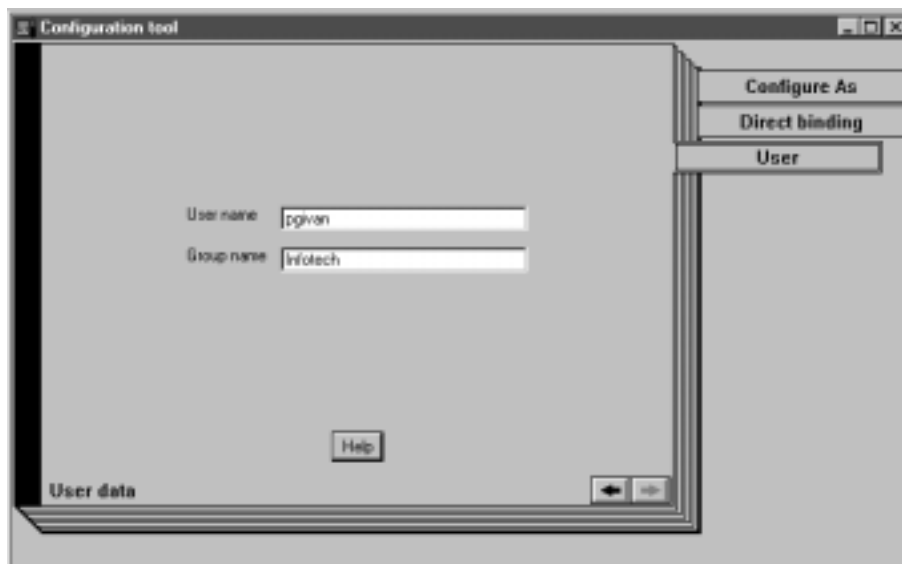


Figure 33. Configuration Tool Notebook - User Section

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 that the License Use Runtime group name is not the same as the Windows NT group name.

License Use Runtime treats the user name as the name of the user who requests and uses licenses on this machine. 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 members of a specific group.

4 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.

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'"
```

## Configuring the Central Registry License Server

### Scenario 5: 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.

Be sure you 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 40.)

Configuring a machine as the central registry license server makes it possible to use licenses of the types shown in Table 10 on page 69. 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 License Use Runtime:

- 1 Double-click on **Configuration Tool** in the License Use Runtime folder. The Configuration Tool notebook is displayed.
- 2 On the **Configure As** page, select **Central Registry License Server**. Note that **Network License Client** is then automatically checked.

Check **Advanced Configuration**, which adds the **User**, **Log**, and **Direct binding ports** pages to the notebook

The **Configure As** page is shown in Figure 34.

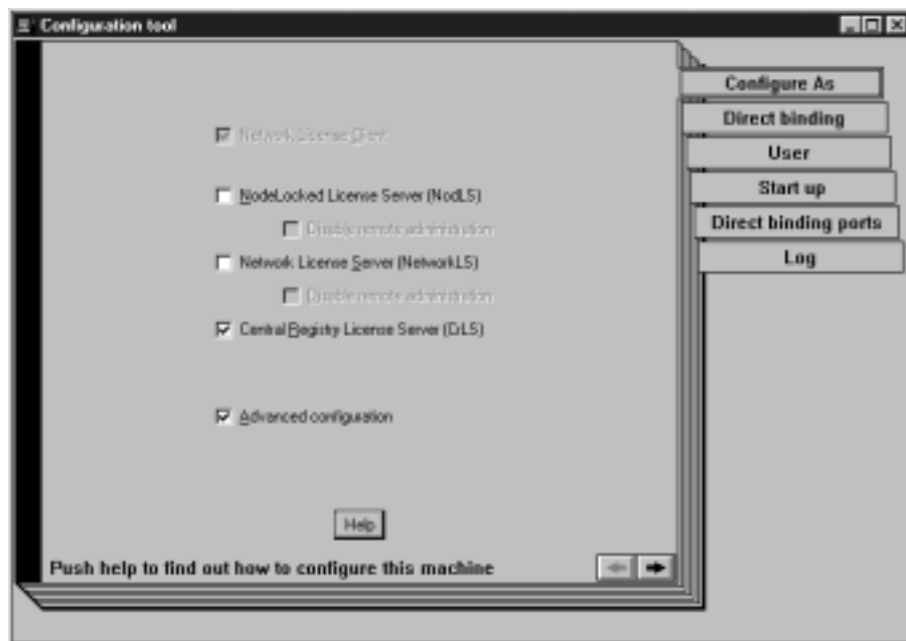


Figure 34. Configuration Tool Notebook - Central Registry License Server

## Configuring the 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 machine. (This option is available on Windows NT only.)
- 4 Select the **Direct binding** tab. The Direct binding section is displayed, as shown in Figure 35.

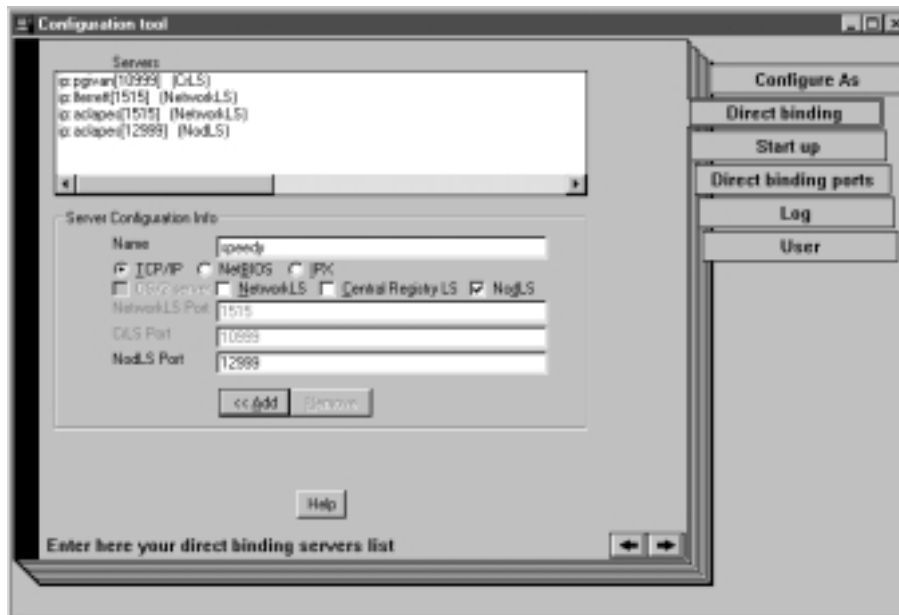


Figure 35. Configuration Tool Notebook - Direct Binding Section

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 machine, include *all* servers in the list.

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

In addition to specifying the direct binding servers list, use this page to specify remote nodelocked license servers whose licenses you want to administer from this machine. Do not include the nodelocked license server on this machine. (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. On the Windows NT Alpha platform, this is the only choice.

## Starting and Listing Processes

- b** Check **NodLS**, **NetworkLS**, or **Central Registry LS**, or any combination, depending on the roles the machine plays in the network.
- c** Leave the default values in the **NetworkLS Port**, **CrLS Port**, and **NodLS Port** fields.



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

- d** Select the <<**Add** push button 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 that the License Use Runtime group name is not the same as the Windows NT group name.
- License Use Runtime treats the user name as the name of the user who requests and uses licenses on this machine. 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 members of 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.”

Now the workstation can 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'"
```

---

## Starting and Listing Your Processes

In Windows NT, you can start and list your processes in two ways: treating them as Windows NT services (“License Use Runtime Processes Defined as Windows NT Services”) or using Service Manager Tool (“Managing License Use Runtime Processes with Service Manager Tool” on page 91).

### License Use Runtime Processes Defined as Windows NT Services

During installation, your License Use Runtime processes are defined as Windows NT services. You can view them in the Settings --> Control Panel --> Services window. If



## Starting and Listing Processes

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 machine. 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 License Use Runtime or one or more of its components, 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 License Use 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 License Use Runtime Processes with Service Manager Tool

When you finish your configuration, double-click on **Service Manager Tool** in the License Use Runtime folder to start the processes. The Service Manager Tool window is displayed, as shown in Figure 36.

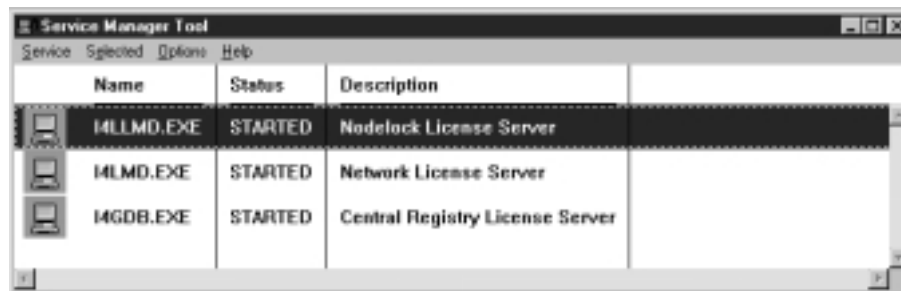


Figure 36. Service Manager Tool Window

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 machine, issue the command:

```
i4cfg -start
```

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

```
i4cfg -list
```

## Administering License Use

---

### 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 5, “License Use Runtime Commands” on page 127.

---

### Administering License Use

The rest of this chapter consists of scenarios that illustrate how the administrator performs the daily activities of managing license-enabled products.

The scenarios assume that the administrator has configured the nodelocked license server, a network license server, and the central registry license server on the server named *PGIVAN*, and that they are all up and running. The Basic License Tool is run from the server named *PGIVAN*.

The scenarios use three sample license-enabled 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*.

Of course, when you perform the activities illustrated in the scenarios you must substitute your own values for variables such as server name, product name, vendor name, enrollment certificate name, product version, and user name.

### Using the Basic License Tool GUI

The scenarios, except for “Scenario 10: Restricting User Access” on page 113, use the Basic License Tool graphical user interface, which features a graphical summary of information about all the products with licenses on the servers you select, a notebook of details about each product, and a graphical summary of information about clusters of network license servers.

#### Starting the Basic License Tool GUI

You must start the Basic License Tool GUI before you can run any of these scenarios. To start the GUI, double-click on the **Basic License Tool** icon in the License Use Runtime folder.

## Administering License Use

Alternatively, start the Basic License Tool GUI by entering the `i4b1t` command with no parameters. The Basic License Tool window is displayed (Figure 37 on page 93).

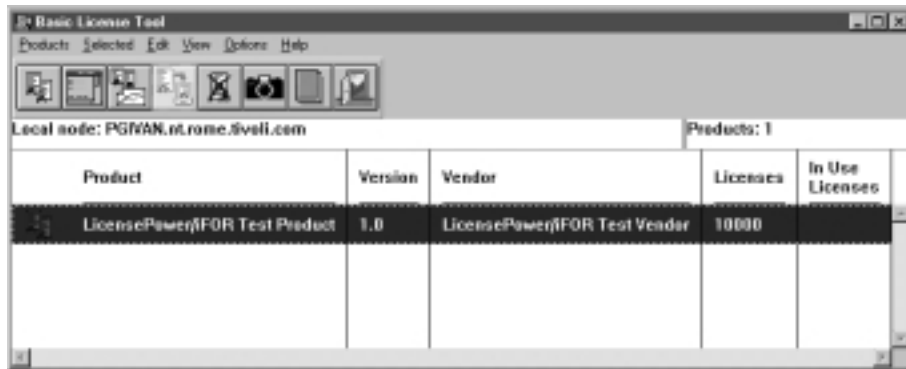


Figure 37. Basic License Tool Window

The products in the list displayed 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

In general, before using the Basic License Tool to view information about concurrent, reservable, per-server, and concurrent nodelocked licenses in use, highlight the product you are interested in and select **Clean up stale licenses** from the **Selected** pull-down. After the cleanup finishes, press F5 to refresh the window.

### Selecting Servers

Use the **Settings...** option of the **Options** pull-down menu to specify types of licenses (nodelocked, network, or both) to be gathered as of the next refresh. You also have the option (exercised from the **Include...** option of the **View** pull-down menu) to select a subset of the active nodelocked and network license servers. Licenses of the types specified in **Settings...**, on the servers specified in **Include...**, are available to the administrator.

Selection of a server is effective as long as the server is active. If a server goes down and is restarted, it is no longer selected.

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 in the **View** pull-down, or deselect or restart the failing server, and try again.

## Managing a Licensed Product

### Using the Basic License Tool Command-Line Interface

You can achieve the same results using the Basic License Tool GUI or using the corresponding command-line interface. At the end of each scenario, a section called “Command-Line Equivalent” shows how to accomplish the same results using the command-line interface.

---

### Performing Basic Administration

The scenarios in this section demonstrate how to:

- Enroll a licensed product (“Scenario 6: Managing a Licensed Product”).
- Distribute licenses from a compound password (“Scenario 6: Managing a Licensed Product”).
- Get a report on the use of licensed products (“Scenario 6: Managing a Licensed Product”).
- Check the current users of licensed products (“Scenario 6: Managing a Licensed Product”).
- Reserve reservable licenses for specific users and monitor the use of reservable licenses (“Scenario 7: Managing Reservable Licenses” on page 102).

### 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 Basic License Tool window (Figure 37 on page 93), select **Products** from the menu bar.
- 2** Select **Enroll product...** from the pull-down menu.  
The Enroll Product window is displayed.
- 3** Select **Import...** The Import window is displayed. From the **File name** list, select the enrollment certificate **smrtjava.lic**, as shown in Figure 38 on page 95.

## Managing a Licensed Product

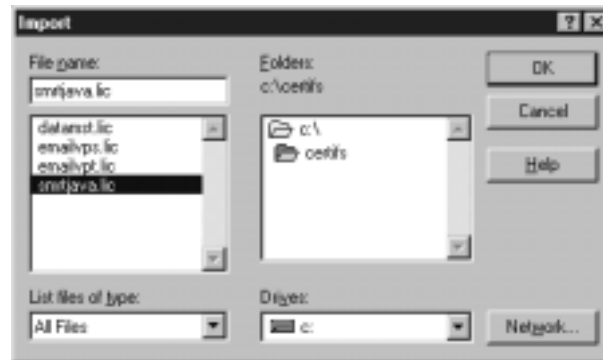


Figure 38. Import Window

Click on **OK**.

The Enroll Product window is redisplayed, filled in with the information from the enrollment certificate file, as shown in Figure 39.



Figure 39. 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*).

## Managing a Licensed Product

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 machine.

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.

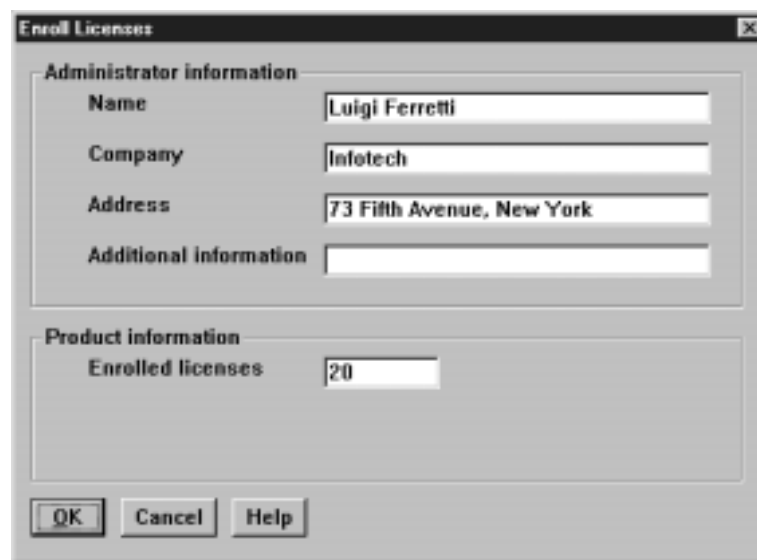
If the product is customer-managed and has nodelocked licenses, select a nodelocked license server.

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.

Click on **OK**.

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

- 5 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 40.



Administrator information	
Name	Luigi Ferretti
Company	Infotech
Address	73 Fifth Avenue, New York
Additional information	

Product information	
Enrolled licenses	20

Buttons: OK, Cancel, Help

Figure 40. Enroll Licenses Window

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

The Basic License Tool window is displayed again, as shown in Figure 41 on page 97. Note that there is a new line corresponding to SMARTJava, and that the icon for the product (a wrapped box) indicates that the product has a compound password.

## Managing a Licensed Product

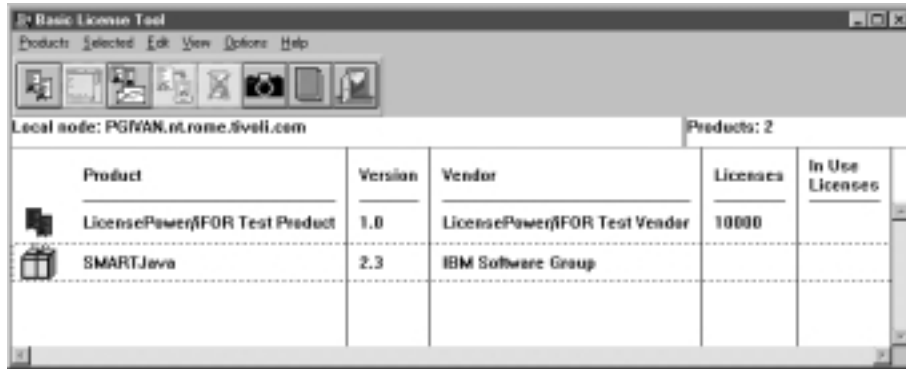


Figure 41. Basic License Tool Window with SMARTJava Enrolled

### 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.

Distribution of licenses is required in the case of network licenses delivered with a compound password. 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*, the administrator performs the following steps:

- 1 Select the line corresponding to the SMARTJava product in the Basic License Tool window (Figure 41).
- 2 Select **Selected** from the menu bar and **Distribute licenses...** from the pull-down menu. The Distribute Licenses window is displayed, as shown in Figure 42.

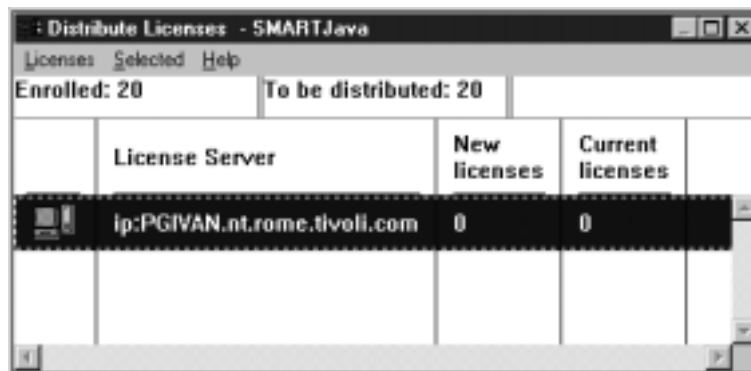


Figure 42. Distribute Licenses Window

## Managing a Licensed Product

- 3 In the Distribute Licenses window, select the network license server. Click on the selected server with the right mouse button. A pop-up menu is displayed. Select **Set number of licenses....** The Set number of licenses window is displayed.
- 4 Enter **5** in the **Number of licenses** field, as shown in Figure 43, and click on **OK**.



Figure 43. Set Number of Licenses Window

The Distribute Licenses window is redisplayed, as shown in Figure 44.

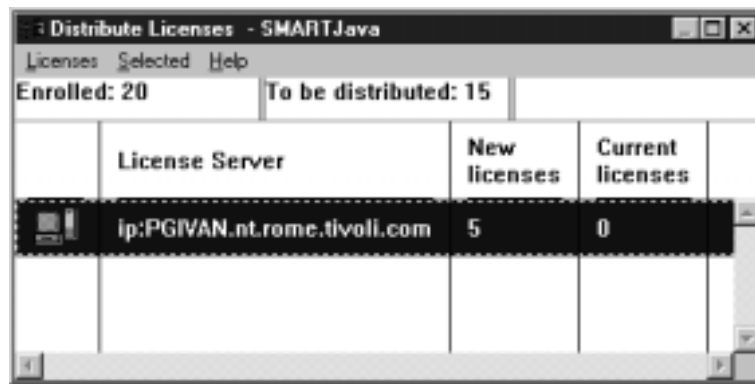


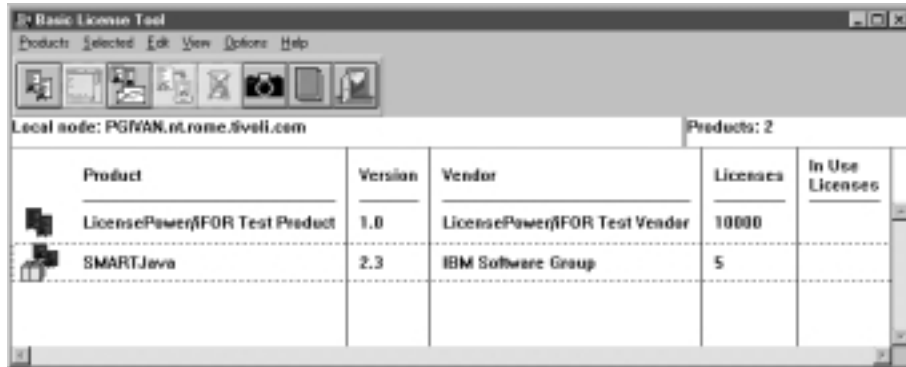
Figure 44. Distribute Licenses Window with Number of Licenses Set

Note that the window now shows 15 as the number of licenses available to be distributed. Select **Distribute** from the **Licenses** pull-down menu to confirm data and distribute the licenses.

- 5 In the Basic License Tool window (Figure 45 on page 99), notice that the number of available licenses has changed to 5 and the icon for SMARTJava has changed to show a compound password with distributed concurrent licenses.



## Managing a Licensed Product



The screenshot shows the 'Basic License Tool' window. At the top, there is a menu bar with 'Products', 'Selected', 'Edit', 'View', 'Options', and 'Help'. Below the menu bar is a toolbar with several icons. The window title is 'Basic License Tool'. Below the toolbar, it says 'Local node: PGMAN.nt.rome.svall.com' and 'Products: 2'. The main area contains a table with the following data:

Product	Version	Vendor	Licenses	In Use Licenses
LicensePower(FOR Test Product)	1.0	LicensePower(FOR 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 the administrator gets a report of usage of SMARTJava for one month: from July 2 to August 2, 1998.

- 1 Select the line corresponding to SMARTJava in the Basic License Tool window (Figure 45).
- 2 Select **Reports...** from the **Selected** pull-down menu.

The Reports window is displayed.

Fill in the Reports window as shown in Figure 46.



The screenshot shows the 'Reports' dialog box. It has several sections:

- Report type:** Radio buttons for 'Standard', 'License requests by product', 'License requests by user', 'License use by product', 'License use by user', and 'Customer-managed use audit'. 'License requests by product' is selected.
- Data range:** A checkbox 'Set data range' is checked. Below it are fields for 'From' and 'To' with columns for 'Year', 'Month', and 'Day'. 'From' is 1998-07-02 and 'To' is 1998-08-02.
- Event filter:** A checkbox 'All events' is checked.
- Product dependent:** Checkboxes for 'LDM modifications', 'Errors', and 'License-related'.
- Product independent:** Checkboxes for 'Vendor messages', 'Server start/stop', and 'File errors'.

At the bottom are 'OK', 'Cancel', and 'Help' buttons.

Figure 46. Reports Window

## Managing a Licensed Product

- a** Select **License requests by product** report type.
  - b** Check **Set date range** 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 on **OK**.
- 3** View the Report window, containing 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
=====

Vendor/Product                Licenses   Licenses   Percent
                               Vrsn  Requested  Granted  Rejections
-----
IBM Software Group            2.3     12         12
SMARTJava

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

- 4** This report shows that 120 requests were made to use SMARTJava Version 2.3, and all of them were granted.
  - 5** Click on **Cancel** in the Reports window.

## Monitoring the Number of Product Users

To check the number of concurrent users of SMARTJava, perform these steps:

- 1** Select the line corresponding to the SMARTJava product on the Basic License Tool window (Figure 45 on page 99).
- 2** Select **Open as Details** from the **Selected** pull-down menu. The Details notebook is displayed.
- 3** Select the **Concurrent Users** tab. The Concurrent Users page is displayed, as shown in Figure 47 on page 101.

## Managing a Licensed Product

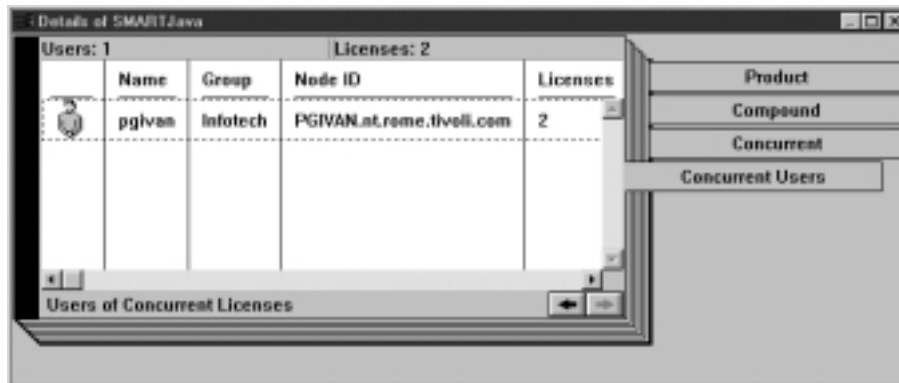


Figure 47. Concurrent Users Page

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 used the graphical user interface 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.

To accomplish the same results using the command line interface, use the following commands:

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

```
i4b1t -a -n PGIVAN -f c:\certifs\smrtjava.lic -T 2  
-R "'Luigi Ferretti' Infotech '73 Fifth Avenue New York'"  
-I "'First installed by Luigi'"
```



If you choose to enroll a product using the command line interface, check the top of the enrollment certificate file; the vendor, while generating the password, may have generated the command to be used.

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

```
i4b1t -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:

```
i4b1t -r2 -p "SMARTJava" -b 7/ 2/1998 -g 8/ 2/1998
```

## Managing Reservable Licenses

To display information about concurrent users of SMARTJava:

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

### Scenario 7: Managing Reservable Licenses

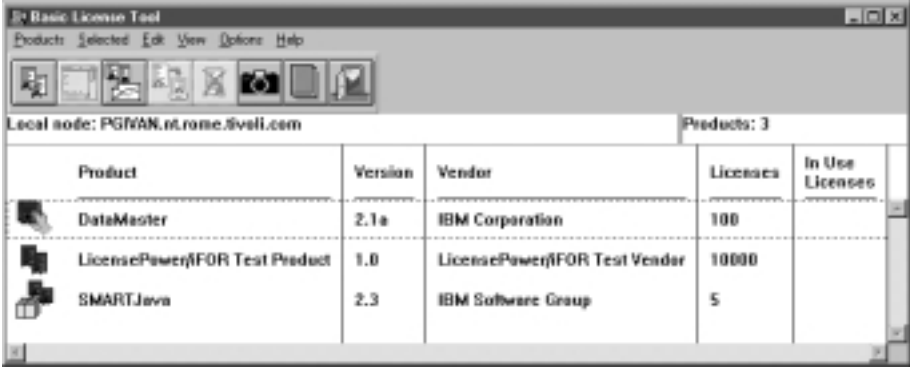
In this scenario, the administrator manages licenses of the DataMaster product. This scenario shows you how to:

- 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 94, except that the Enroll Licenses window is not used.

The enrollment certificate file for DataMaster is shown as an example in “Checking License Details” on page 185.

The product appears in the Basic License Tool window, as shown in Figure 48.



The screenshot shows the Basic License Tool window with a menu bar (Products, Selected, Edit, View, Options, Help) and a toolbar. Below the toolbar, it displays 'Local node: PGMAN.nt.rame.svali.com' and 'Products: 3'. A table lists the following products:




Product	Version	Vendor	Licenses	In Use Licenses
 DataMaster	2.1a	IBM Corporation	100	
 LicensePower(FOR Test Product)	1.0	LicensePower(FOR Test Vendor)	10000	
 SMARTJava	2.3	IBM Software Group	5	

Figure 48. Basic License Tool Window with Reservable Licenses

Note that the icon (a hand holding some licenses) indicates that the licenses are reservable.

To reserve some of the licenses for a specific user and monitor use of the reservable licenses, the administrator performs the following steps:

- 1 Double-click on the product in the Basic License Tool window and go to the Reservable tab of the Details notebook, as shown in Figure 49 on page 103.

## Managing Reservable Licenses

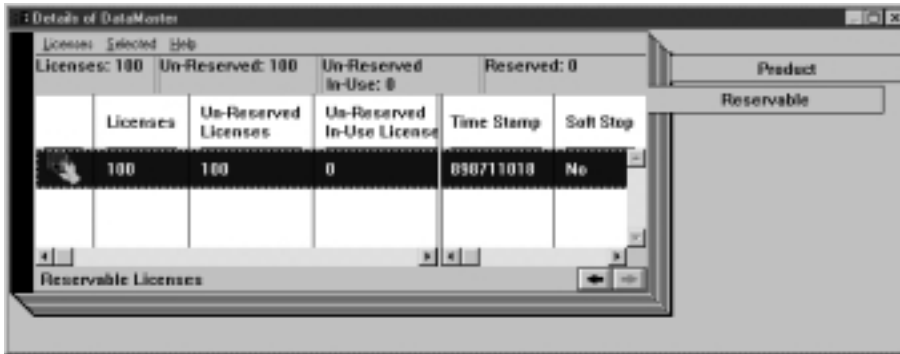


Figure 49. Details Notebook with Reservable Tab. Custom configuration serial number column not shown.

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

- 2 Click on the product with the right mouse button, 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/or node ID for which you want to reserve licenses; and the date and time that the reservation is to expire, as shown in Figure 50. Note that the latest allowed expiration date of a reservation is 12/31/2037. 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.



Figure 50. Reserving Reservable Licenses

Click on **OK**.

- 4 Returning to the Details notebook, note that the Reserved page has been added, as shown in Figure 51 on page 104.

## Managing Reservable Licenses

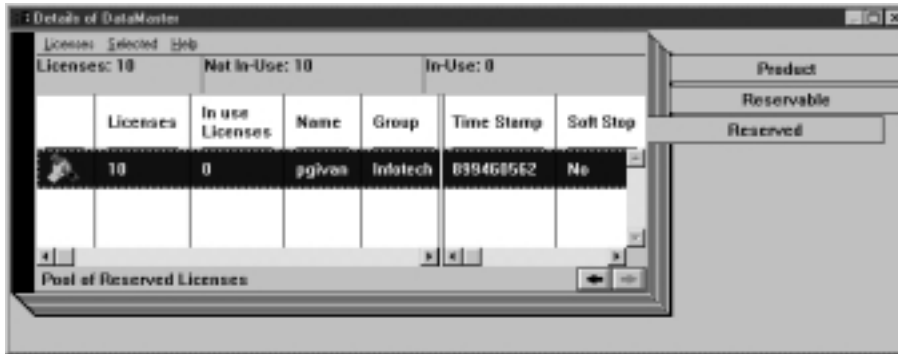


Figure 51. Details Notebook with Reserved Licenses

Note that this page shows 10 licenses reserved for the user *pgivan* in the group *Infotech*.

- Now assume that the user *pgivan* requests a license. Return to the Basic License Tool window and press F5 or **Refresh now**. Returning to the Details notebook, note that the Reserved Users page has been added, as shown in Figure 52.

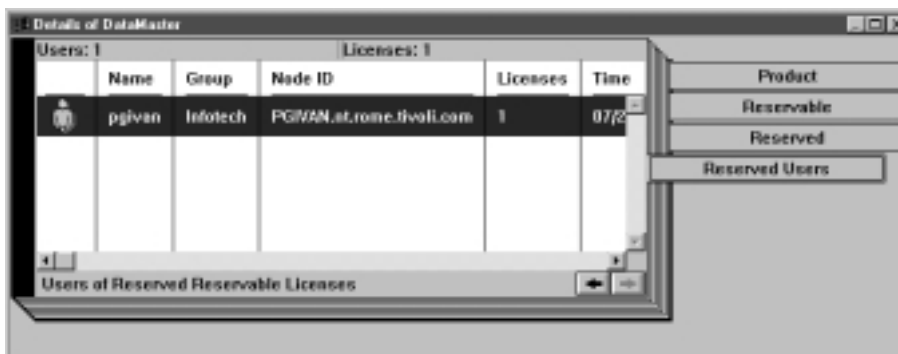


Figure 52. Reserved Users Page with Reservable Licenses in Use

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.

Note that the Un-Reserved Users page is added to the Details notebook for DataMaster, as shown in Figure 53 on page 105.

## Managing Reservable Licenses

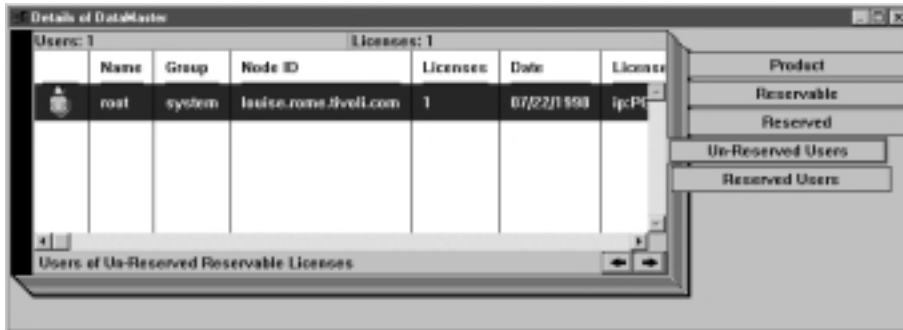


Figure 53. Un-Reserved Users Page

- 7 Checking the Reservable page at this point, note that it shows 10 reserved licenses, 1 unreserved license in use, and 90 unreserved licenses, as shown in Figure 54.

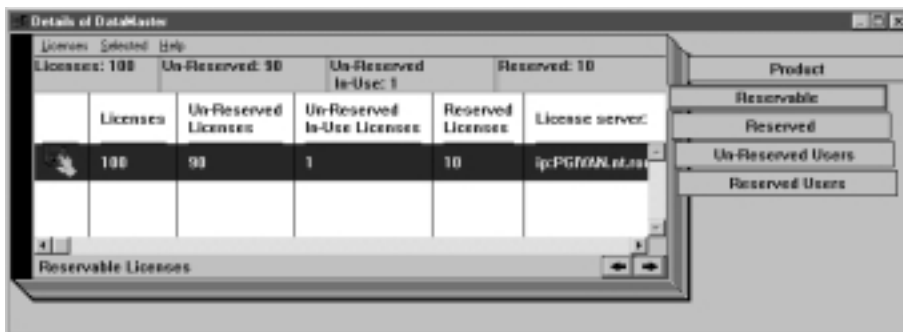


Figure 54. Reservable Page with Unreserved Licenses in Use

### Command-Line Equivalent

This scenario used the graphical user interface 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.

To accomplish the same results using the command line interface, use the following commands:

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

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

## Switching from Per-Server to Per-Seat Licenses

**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 89946 562 -A 1 -g 11/ 2/1998 -H 18: -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-Controlled Policies” on page 13. 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 113).

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

In this scenario, the administrator switches the per-server/per-seat policy for the product e-MailVision from per-server to per-seat. The per-server license has already been enrolled, as shown in the Basic License Tool window (Figure 55 on page 107). The enrollment process is the same as for customer-managed concurrent licenses, as shown in “Enrolling the Product” on page 94.



## Switching from Per-Server to Per-Seat Licenses



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

Figure 55. Basic License Tool Window with Per-Server Licenses Enrolled

Notice that, because per-seat licensing has not yet been enabled, the icon for e-MailVision shows per-server licenses.

To do the switch, the administrator follows these steps:

- 1 Install the per-seat license through the Basic License Tool, 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 shown in “Enrolling the Product” on page 94.
- 2 Select the line corresponding to e-MailVision in the Basic License Tool window (Figure 55). Select **Selected** from the menu bar, and then **Update licenses...** from the pull-down menu. The Details notebook is displayed. In the **Per-Seat** page, click with the right mouse button on the entry for the product, and select **Update licenses...** from the pop-up menu.

The Update Licenses window is displayed. Check the **Enable per-seat licensing** box, as shown in Figure 56 on page 108.

## Switching from Per-Server to Per-Seat Licenses

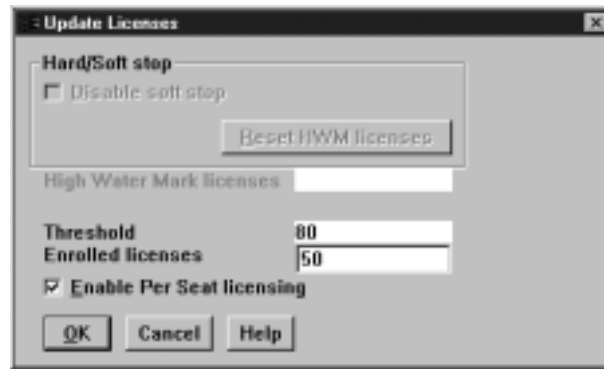
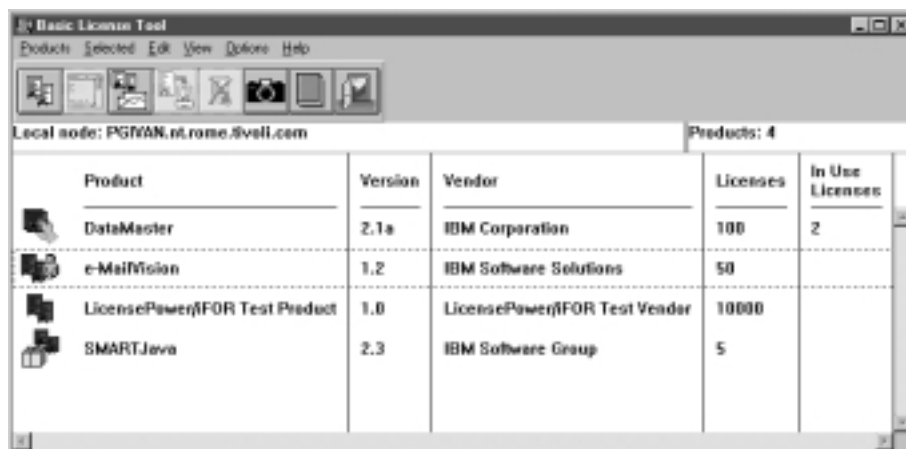


Figure 56. Enabling Per-Seat Licensing

- 3 Click on **OK**.
- 4 In the Basic License Tool window (Figure 57), note that the icon has changed to show per-seat licenses. From now on, requests from application clients will result in the granting of per-seat licenses. The license server will remove the old per-server licenses.



Product	Version	Vendor	Licenses	In Use Licenses
DataMaster	2.1a	IBM Corporation	100	2
e-MailVision	1.2	IBM Software Solutions	50	
LicensePower(FOR Test Product)	1.0	LicensePower(FOR Test Vendor	10000	
SMARTJava	2.3	IBM Software Group	5	

Figure 57. Basic License Tool Window with Per-Seat Licenses Enrolled and Enabled

## Using the Hard Stop/Soft Stop Policy

### Command-Line Equivalent

This scenario used the graphical user interface to switch the e-MailVision product from per-server licensing to per-seat licensing.

To accomplish the same results using the command line interface, use the following commands:

#### 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 94, 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:

- 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)

- Decides to acquire ten more licenses

- Updates the number of enrolled licenses to 30

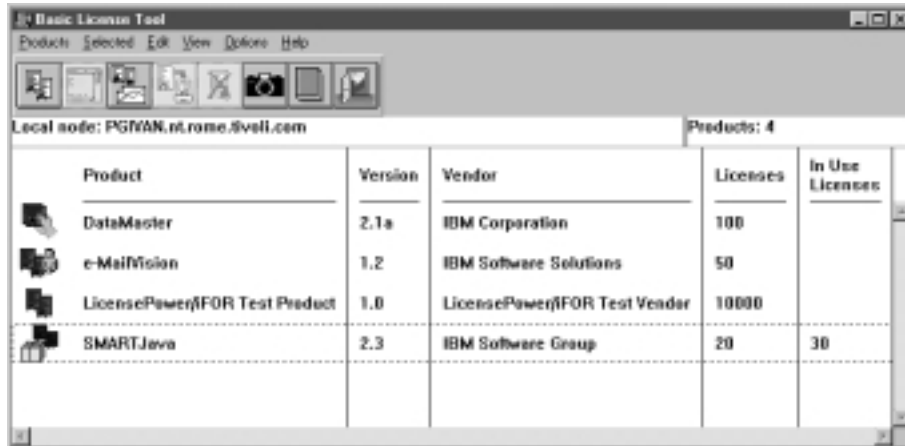
- Resets the high-water mark to 0

- Distributes ten more licenses from the compound password

Follow these steps:

- 1 Check the current license usage in the Basic License Tool window (Figure 58 on page 110).

## Using the Hard Stop/Soft Stop Policy



The screenshot shows the 'Basic License Tool' window with a menu bar (Products, Selected, Edit, View, Options, Help) and a toolbar. Below the toolbar, the local node is 'PGNYAN.nt.rame.svall.com' and there are 4 products. The main area contains a table with the following data:

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

Figure 58. Basic License Tool 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:  
Select SMARTJava in the Basic License Tool window (Figure 58).
- 3 Select **Update licenses...** from the **Selected** pull-down menu. The Details notebook is displayed. 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 59).

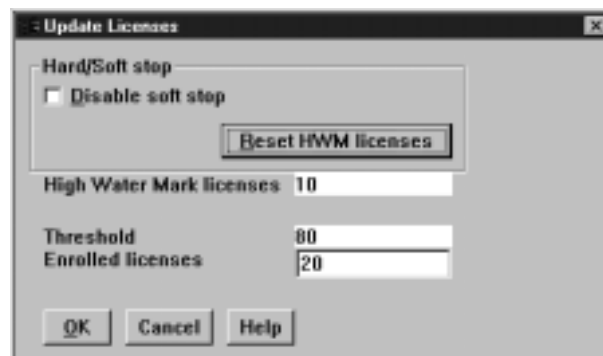


Figure 59. Update Licenses Window with High-Water Mark

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

- 4 Decide to acquire ten more licenses, and pay the vendor for them.

## Using the Hard Stop/Soft Stop Policy

- 5 Returning to the Update Licenses window, enter 30 in the **Enrolled licenses** field, as shown in Figure 60 on page 111.

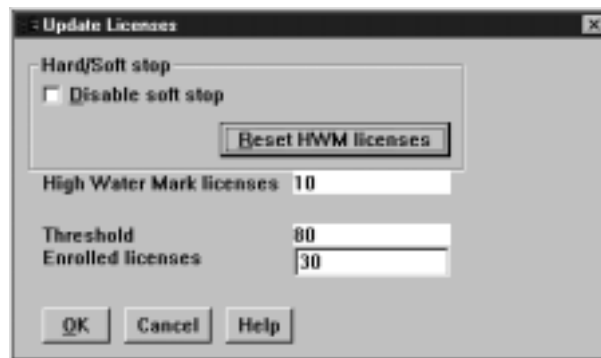


Figure 60. Update Licenses Window - Enrolling More Licenses

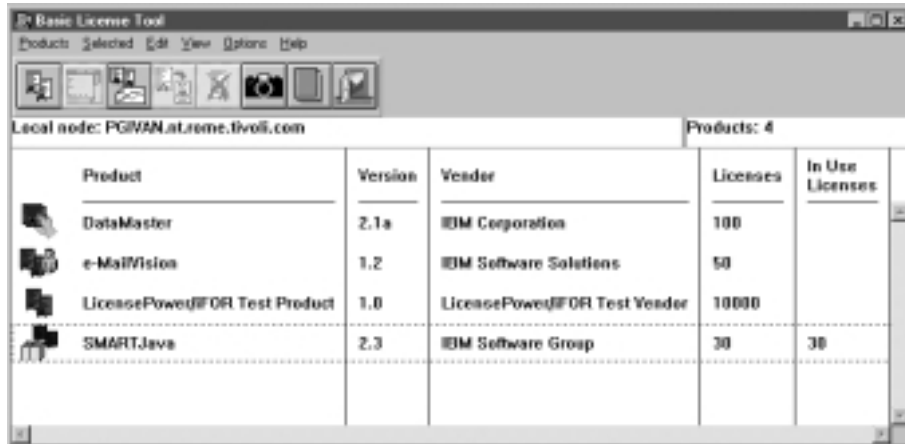
At the same time, reset the high-water mark to 0 by clicking on **Reset HWM Licenses** (Figure 61).



Figure 61. Resetting the High-Water Mark

- 6 Click on **OK**. A confirmation message is displayed.
- 7 Distribute the ten newly enrolled licenses (see "Distributing the Licenses" on page 97). In the Basic License Tool window, note that the number of available licenses for SMARTJava has been updated, as shown in Figure 62 on page 112.

## Using the Hard Stop/Soft Stop Policy



The screenshot shows the 'Basic License Tool' window. At the top, it says 'Local node: PGIVAN.at.reme.tivoli.com' and 'Products: 4'. Below this is a table with the following data:

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

Figure 62. Basic License Tool Window with Licenses Updated

### Command-Line Equivalent

This scenario used the graphical user interface to check usage of soft stop licenses for SMARTJava, enrolled and distributed an additional ten licenses, and reset the high-water mark to 0.

To accomplish the same results using the command line interface, use the following commands:

#### 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 3 -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 1  
-w PGIVAN -I "'Luigi Ferretti'"
```

## Restricting User Access

### 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:

#### **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.

**allow** and **disallow** are mutually exclusive.

#### **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.

**allow** and **disallow** are mutually exclusive.

## Creating and Administering a Cluster

- 4 Store the file in the `\ifor\ls\conf` directory of the machine 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.

---

## Administering High-Availability Licensing

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.

Note that when you create a cluster, License Use 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 Administering 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, be sure all the servers you intend to put in the cluster are configured to communicate with each other through direct binding.
- 2 In the Basic License Tool window (Figure 37 on page 93), select **View** from the menu bar and **Clusters...** from the pull-down menu. The Clusters window is displayed, as shown in Figure 63 on page 115.



## Creating and Administering a Cluster

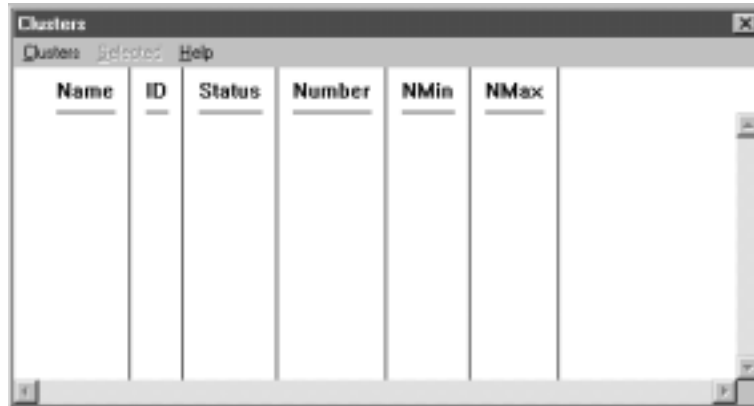


Figure 63. Clusters Window

- 3 Select **Clusters** from the menu bar and **Create...** from the pull-down menu. The Create Cluster window is displayed.
- 4 Use the **Initial Number** entry field 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 64, the administrator specifies that the initial number of members is 4.

Use the **Name** entry 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*.

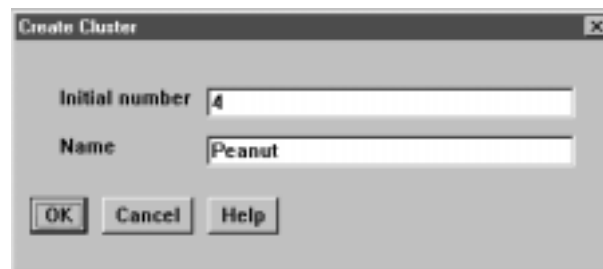


Figure 64. Create Cluster Window

- 5 Click on **OK**. A pop-up message gives you a chance to change the initial number of members.
- 6 After you confirm your choices, the **Define Cluster Members** window is displayed.

## Creating and Administering a Cluster

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 OS/2, Windows 95, and Windows 98 servers may be listed, do not select them. Only AIX, HP-UX, IRIX, Solaris, Windows NT (x86), Windows NT Alpha, Windows Terminal Server (x86), and Windows Terminal Server Alpha network license servers can 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 this example, as shown in Figure 65, the administrator selects servers *moon*, *hydra*, *speedy*, and *louise*.



Figure 65. Define Cluster Members Window

- 7 Click on **OK**. A pop-up window is displayed to give you the chance to change your selections. The Clusters window is redisplayed with an entry for the newly-defined cluster, as shown in Figure 66.

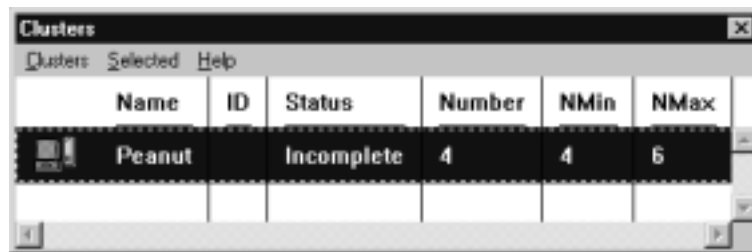


Figure 66. Clusters Window with New Cluster Added

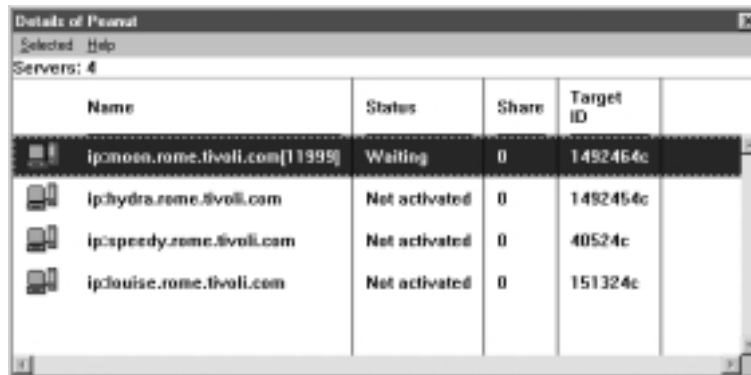
Note that the status of the cluster is **Incomplete**, because not enough servers have been activated yet.

## Creating and Administering a Cluster

### 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 pull-down menu. The Details of Peanut window is displayed, as shown in Figure 67.



The screenshot shows a window titled 'Details of Peanut' with a menu bar containing 'Selected' and 'Help'. Below the menu bar, it says 'Servers: 4'. The main content is a table with the following data:

Name	Status	Share	Target ID
ip:moon.rome.tivoli.com[1199]	Waiting	0	1492464c
ip:hydra.rome.tivoli.com	Not activated	0	1492464c
ip:speedy.rome.tivoli.com	Not activated	0	40524c
ip:louise.rome.tivoli.com	Not activated	0	151324c

Figure 67. Details of New Cluster

Note that 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. Note also that 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 the cluster. Return to the Basic License Tool window and use F5 to refresh the display, and then view the Clusters window again until the cluster has exited from **Change Pending** status.

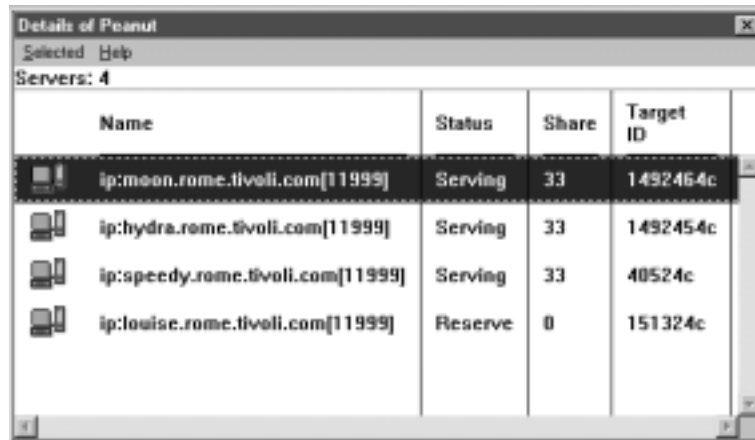
- 3 Return to the Details of Peanut window. Note that:

The status of servers *moon*, *hydra*, and *speedy* has changed to **Serving**.

Each server is serving 33% of the licenses, as shown in Figure 68 on page 118.

Server *louise* is in **Reserve** status, ready to take over if *moon*, *hydra*, or *speedy* goes down.

## Creating and Administering a Cluster



The screenshot shows a window titled "Details of Peanut" with a menu bar containing "Selected" and "Help". Below the menu bar, it says "Servers: 4". The main content is a table with four columns: "Name", "Status", "Share", and "Target ID".

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 68. Details of Cluster after Three Activations

- 4 Return to the Clusters window (Figure 69). Note that the status of the cluster has changed to **Active**. The cluster ID has been generated and is displayed. The administrator can give the cluster ID to software vendors who will create passwords bound to the cluster.



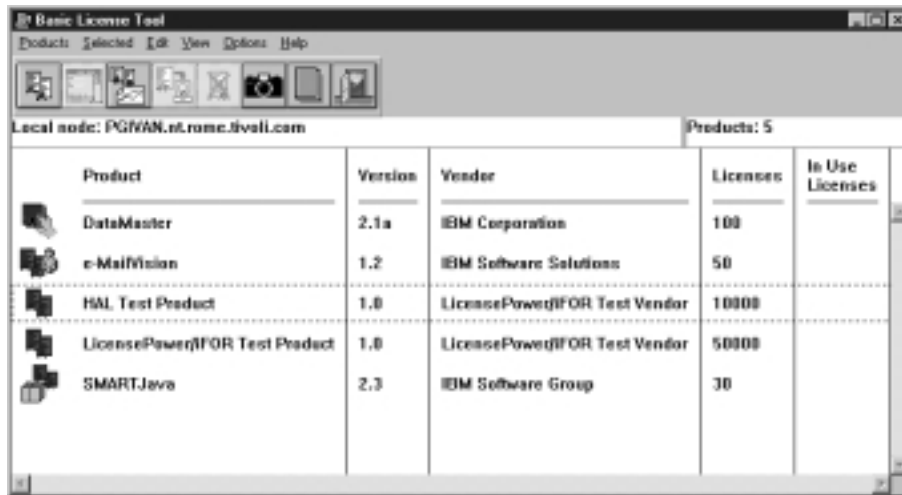
The screenshot shows a window titled "Clusters" with a menu bar containing "Clusters", "Selected", and "Help". Below the menu bar, there is a table with six columns: "Name", "ID", "Status", "Number", "NMin", and "NMax".

Name	ID	Status	Number	NMin	NMax
Peanut	8c8dd27acc99.8d.14.92.46.4c.00.00.00	Active	4	4	6

Figure 69. Clusters Window with Cluster ID

- 5 Return to the Basic License Tool window (Figure 70 on page 119). Note that the HAL test product has been enrolled on the cluster. You can use it to test operation of the cluster.

## Creating and Administering a Cluster



The screenshot shows the 'Basic License Tool' window. At the top, it says 'Local node: PGWAW.r1.rome.it.vall.com' and 'Products: 5'. Below this is a table with the following data:

Product	Version	Vendor	Licenses	In Use Licenses
DataMaster	2.1a	IBM Corporation	100	
e-MailVision	1.2	IBM Software Solutions	50	
HAL Test Product	1.0	LicensePower(FOR Test Vendor	10000	
LicensePower(FOR Test Product	1.0	LicensePower(FOR Test Vendor	50000	
SMARTJava	2.3	IBM Software Group	30	

Figure 70. Basic License Tool Window with HAL Test Product

### 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** pull-down, select **Add cluster member...** The Add Cluster Members window is displayed.

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 OS/2, Windows 95, and Windows 98 servers may be listed, do not select them. Only AIX, HP-UX, IRIX, Solaris, Windows NT (x86), Windows NT Alpha, Windows Terminal Server (x86), and Windows Terminal Server Alpha network license servers can be members of a cluster.

Select a server (in this example, *thelma*) from the **Available NetworkLS** list and use the **Add>>** push button 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. Figure 71 on page 120 shows the Add Cluster Members window after a fifth member has been moved to the **Cluster Members** list.

## Creating and Administering a Cluster

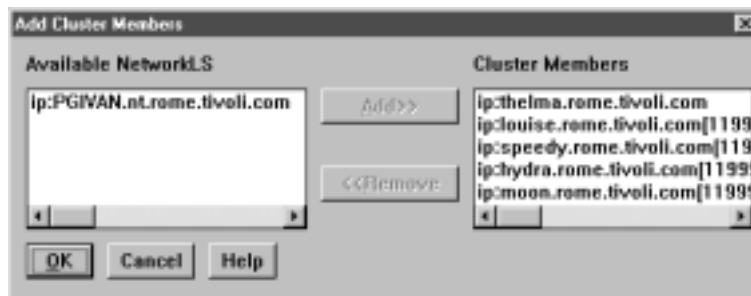
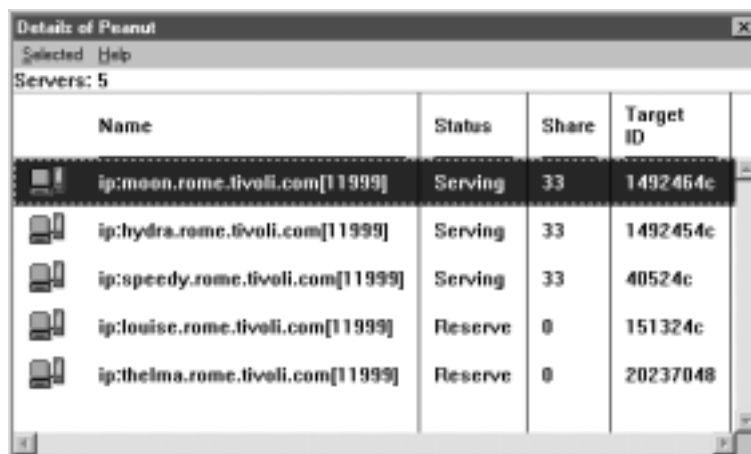


Figure 71. Add Cluster Members Window

- 2 Click on **OK**. A pop-up window is displayed to give you the chance to change your selection. The Clusters window is redisplayed; note that the number of servers in *Peanut* is 5, and the status of the cluster is **Active**. The newly added server is automatically activated.
- 3 Open the Details of Peanut window. Note that servers *moon*, *hydra*, and *speedy* are all serving, that each is serving one-third of the licenses, and that *louise* and *thelma* are in reserve, as shown in Figure 72.



Name	Status	Share	Target ID
ip:moon.rome.tivoli.com[1199]	Serving	33	1492464c
ip:hydra.rome.tivoli.com[1199]	Serving	33	1492454c
ip:speedy.rome.tivoli.com[1199]	Serving	33	40524c
ip:louise.rome.tivoli.com[1199]	Reserve	0	151324c
ip:thelma.rome.tivoli.com[1199]	Reserve	0	20237048

Figure 72. Details of Cluster after Adding a Server

### 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.

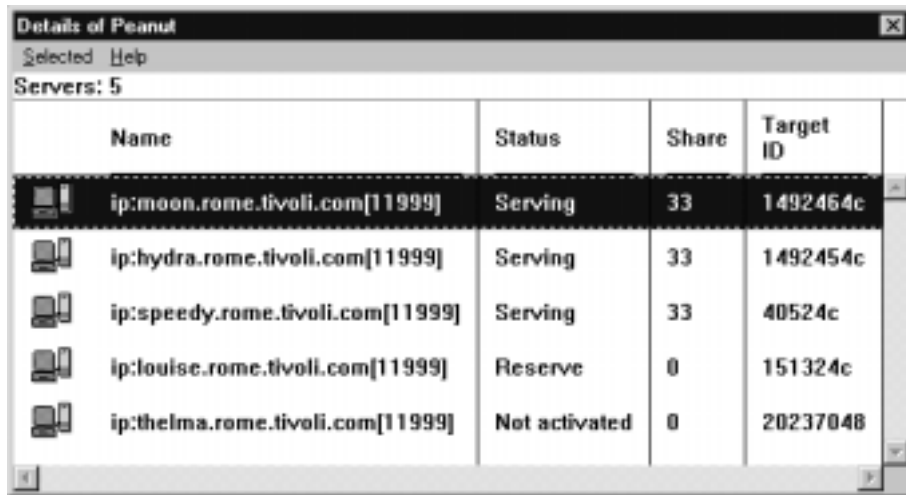
To deactivate a server in a cluster:

- 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.

## Creating and Administering a Cluster

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 the cluster. Return to the Basic License Tool 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**, as shown in Figure 73.



The screenshot shows a window titled "Details of Peanut" with a menu bar containing "Selected" and "Help". Below the menu bar, it says "Servers: 5". The main content is a table with the following columns: Name, Status, Share, and Target ID. The table contains five rows of server information.

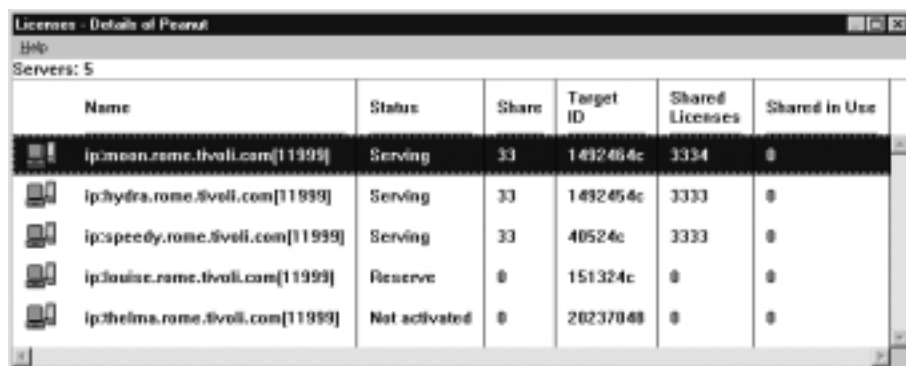
Name	Status	Share	Target ID
ip:moon.rome.tivoli.com[11999]	Serving	33	1492464c
ip:hydra.rome.tivoli.com[11999]	Serving	33	1492454c
ip:speedy.rome.tivoli.com[11999]	Serving	33	40524c
ip:louisc.rome.tivoli.com[11999]	Reserve	0	151324c
ip:thelma.rome.tivoli.com[11999]	Not activated	0	20237048

Figure 73. Details of Cluster after Deactivation

### 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 Basic License Tool window and open the product's Details notebook. On the Concurrent page, click on the product with the right mouse button and select **Show Servers**. The Details of Cluster window is displayed with additional information about the number of licenses being served by each server, as shown in Figure 74 on page 122.

## Creating and Administering a Cluster



The screenshot shows a window titled "Licenses - Details of Peanut" with a "Help" button and "Servers: 5" indicator. Below is a table with the following data:

Name	Status	Share	Target ID	Shared Licenses	Shared in Use
ip:moon.rome.itvol.com[11999]	Serving	33	1492464c	3334	0
ip:hydra.rome.itvol.com[11999]	Serving	33	1492464c	3333	0
ip:speedy.rome.itvol.com[11999]	Serving	33	40524c	3333	0
ip:louise.rome.itvol.com[11999]	Reserve	0	151324c	0	0
ip:thelma.rome.itvol.com[11999]	Not activated	0	20237040	0	0

Figure 74. Details of Servers Serving HAL Test Product

### Enrolling and Removing Licenses on a Cluster

Checking the Clusters window, you can see the cluster ID of any cluster.

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
```



## Upgrading a Custom Configuration

To get a report of cluster status from the perspective of one of the activated servers in the cluster:

```
i4blt -H s -n moon
```

---

### 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 components, 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 on the product in the i4blt window, then select the **Concurrent** tab of the Details notebook. The Details page looks similar to that shown in Figure 75.

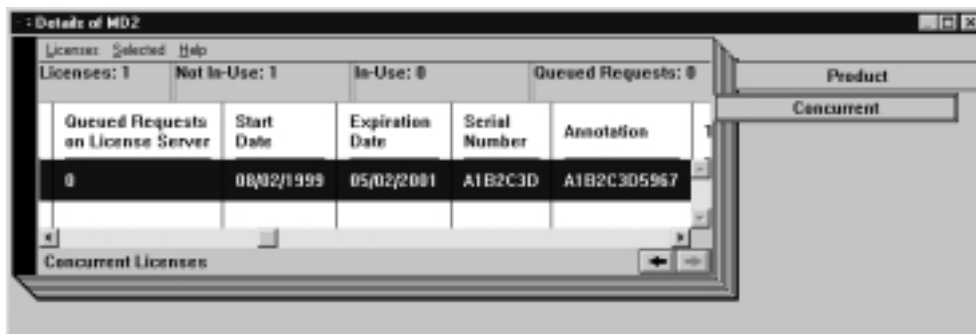


Figure 75. Initial State of the Concurrent Page of the Details Notebook for a Custom Configuration License

## Upgrading a Custom Configuration

To upgrade your current license:

- 1 In the i4blt window, select **Products** from the menu bar.

The Enroll Product window is displayed, as shown in Figure 76.



Figure 76. Enroll Product Window for Custom Configuration

- 2 Select **Import**.

The Import window is displayed.

In the **Filter** field, enter the path to the directory in which the enrollment certificate is stored (in this example, `home\ferretti\certif`). Then, from **Files**, select the custom configuration enrollment certificate (in this example, **m2update.lic**), as shown in Figure 77 on page 125.

## Upgrading a Custom Configuration

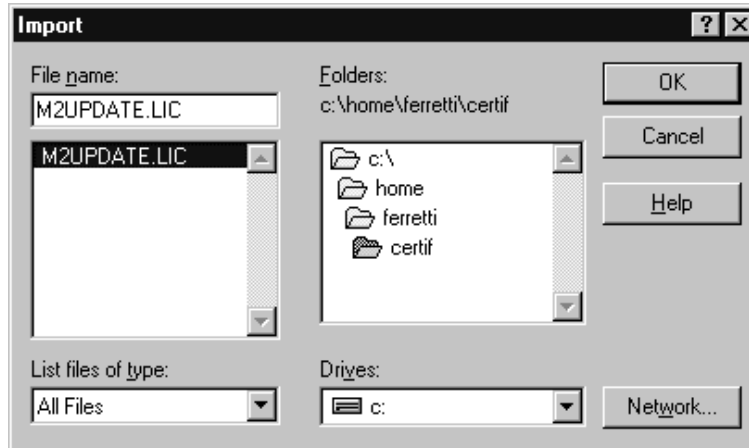


Figure 77. Import Window for Custom Configuration

Click **OK**.

The Enroll Product window is displayed, filled in with information from the enrollment certificate file, as shown in Figure 78.

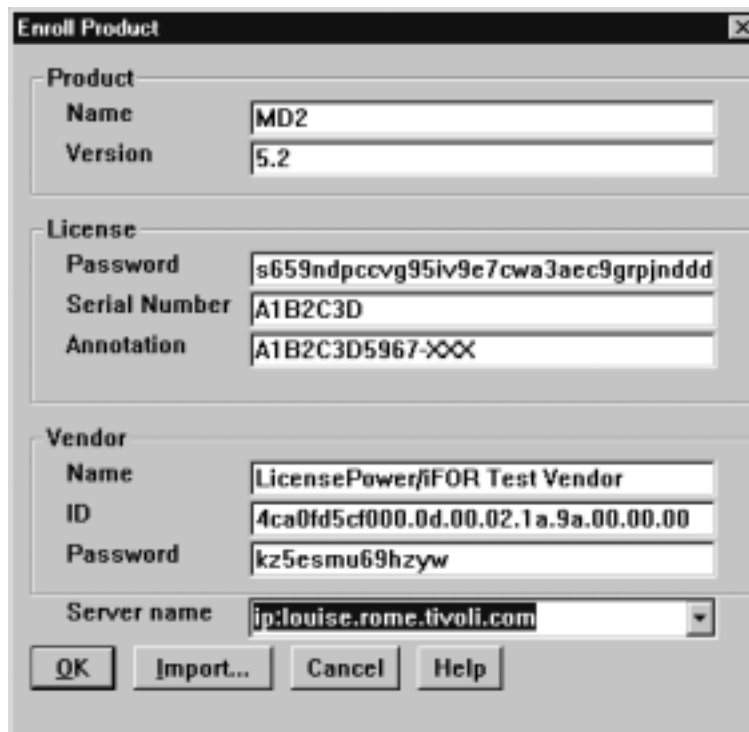


Figure 78. Enroll Product Window for Custom Configuration

## Upgrading a Custom Configuration

- 3 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.

If a specific target ID is set in the enrollment certificate, the licenses must be enrolled on the network license server of that target machine.

If the target ID in the enrollment certificate file is set to ANY, select a network license server.

Note that the license serial number is the same as it was for the previously enrolled license.

Click **OK**.

The Basic License Tool window is displayed.

The Concurrent page of the Details notebook now contains the updated license information, as shown in Figure 79. 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.

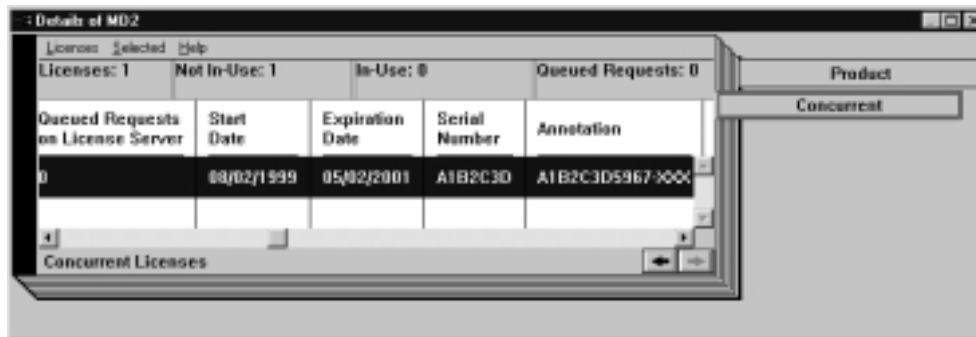


Figure 79. Upgraded State of the Concurrent Page of the Details Notebook for a Custom Configuration License

### 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` Is the name of the file that contains the upgraded license certificate.

`louise` Is the name of the server on which the initial key is installed.

---

## Chapter 5. License Use Runtime Commands

This chapter describes how to use the License Use Runtime command line interface.

In the .HTM version of this Command Reference, changes made since Version 4.0 are shown in purple.

In the descriptions of command syntax, the following conventions are used:

- Code items shown in **bold** type exactly as shown.
- Replace items shown in *italic* type with your own values.
- Parameters shown in brackets ( [ ] ) are optional.
- Choose one from a list of parameters shown in braces ( { } ).

The following commands are available:

<b>i4blt</b>	Basic License Tool
<b>i4cfg</b>	Configuration Tool
<b>i4tv</b>	Test Verification Tool
<b>i4target</b>	Target View Tool
<b>i4ls</b>	Network License Server Process
<b>i4ls -b</b>	Nodelocked License Server Process
<b>i4ls -a</b>	Central Registry Process
<b>i4nbssvr</b>	NetBIOS Socket Server Process (Windows NT only)
<b>i4lct</b>	License Creation Tool
<b>i4nat</b>	Nodelocked Administration Tool, for backward compatibility only

## i4blt - Basic License Tool

---

### i4blt - Basic License Tool

If issued with no options, the **i4blt** command starts the Basic License Tool graphical user interface.

#### General Rules for the i4blt Command

1. 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)*

2. 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*

3. 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"
```

4. 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"
```

5. The parameters you specify in any of the command options (for example, server names, vendor names, and product names) are case-sensitive.

6. 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**

## i4blt - Basic License Tool

### Examples

Display the i4blt -E syntax:

```
i4blt -E
```

Display the i4blt -r syntax:

```
i4blt -rh
```

### Primary Command Options

The i4blt command has the following primary command options:

**-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 Product License or an Application Client Identifier)**

Delete products from a license database, or 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, products, and licenses.

**-s (Display Product License Status)**

Gather status information about product 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.

## i4blt - Basic License Tool

### **-h (Display Help)**

Display command syntax and usage information about the Basic License Tool command-line interface.

### **-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.

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  
[ -R administrator_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
```

### **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 machine.



## i4blt - Basic License Tool

If the product is vendor-managed, and the enrollment certificate file specifies a target ID, the licenses are enrolled on that machine.

If the product is vendor-managed, and the enrollment certificate file does not specify a target ID, the licenses are enrolled on the local machine.

**-f *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.

**-v *vendor\_name* (If you do not have the enrollment certificate)**

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***.

***vendor\_password***

The unique vendor password string for the vendor that you specify in the ***vendor\_name*** argument.

**-p *product\_info* (If you do not have the enrollment certificate)**

The information on the licensed product that you intend to install.

***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\_info* (for customer-managed use products only)**

The information on the administrator who enrolls the product.

***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.

## i4blt - Basic License Tool

**-T *enrolled\_licenses* (for customer-managed use products only)**

The number of licenses you have acquired from the software supplier. This parameter is required.

**-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***

The serial number of a custom configuration license. The serial number is a string of up to 31 alphanumeric characters that uniquely identifies a custom configuration.

### Examples

Add a customer-managed use product:

```
i4blt -a
-v "Venus 4ca fd5cf . d. . 2.1a.9a. . . kz5esmu69hzyw"
-p "timer 1.1 wzx3ewdfrvu4v64d53bbrkzhheaaaaa"
-R "Alex IBM Rome" -T 1
-I "'Alex Marshall'"
```

Add a vendor-managed use product:

```
i4blt -a -n thelma
-p "scena 1. suf fpeixfi5v78a22xxrkzhheaaaaa"
-v "Operatix 7gp4ac8jj . d. . 2.1a.9a. . . 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** ] ]

## i4blt - Basic License Tool

```
[ -H hard_soft_mode [ yes | no ] ]  
[ -M hwm_reset ]  
[ -t threshold [ 1...100 ] ]  
[ -l signature_information ]
```

### Parameters

#### **-v** *vendor\_name*

The name of the vendor that manufactured the product that you intend to update.

#### **-p** *product\_info*

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*

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 ones.

#### **-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.

## i4blt - Basic License Tool

### **-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.

### **-l *signature\_information***

Information about the user issuing the command, to be stored with the signature stamp. Use this parameter along with the **-T** parameter.

## Examples

Update the number of licenses for the **Test Compiler** product, Version **1.1** of vendor **Psychosync** to **50**. The product has network licenses.

```
i4blt -U -v "Psychosync" -p "'Test Compiler' 1.1"  
-T 5 -I "'Paula'"
```

Set the soft stop policy and reset the high-water mark of the **Test Compiler** product, Version **1.1** of vendor **Psychosync**. The product has network licenses.

```
i4blt -U -v "Psychosync" -p "'Test Compiler' 1.1" -H no -M
```

Update to **5** the number of nodelocked licenses for the **ScreenPic** product, Version **2** of vendor **ArtTools** on nodelocked license 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  
[-l 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\_info***

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.

## i4blt - Basic License Tool

### **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

This option deletes a product license 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* }

[ **-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 command:

```
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.

## i4blt - Basic License Tool

**-v *vendor\_name***

Name of the vendor whose product license you intend to delete.

**-p *product\_info***

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.

**-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 an expired license to use 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. " -t 7814 1788
```

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. " -A thelma
```

## i4blt - Basic License Tool

### **-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_info  
[ -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\_info***

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**.

## i4blt - Basic License Tool

### **-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 **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 3/ 2/1998 -H 11: -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, License Use 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 { I | 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 **I** 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.



## i4blt - Basic License Tool

### **-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, 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 ] | l ]  
[ -F server_type { l | w | a } ]  
[ -n server_names ]  
[ -v vendor_names ]  
[ -p { product_name[?product_version] } ... ]  
[ -u user_names ]  
[ -t timestamp ]
```

### **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 **-n**, **-v**, **-p**, or **-u** filter option together 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.

## i4blt - Basic License Tool

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 **-v**, **-p**, or **-u** filter option together with this parameter.

### **-lp [ -i ]**

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.

Specify the **-i** option to display detailed information about each product in a product list.

### **-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.

## i4blt - Basic License Tool

To list information on a specific license, use the **-t** filter option, followed by the timestamp of the license.

### High-Availability Output:

Cluster name

#### **For each server in the cluster:**

- Server name
- Server status:

<b>Serving</b>	Running, serving licenses
<b>Waiting</b>	Server is ready, but the cluster is in incomplete or inactive state
<b>Unavailable</b>	Not started
<b>Reserve</b>	In reserve in case a serving server becomes unavailable
<b>Not activated</b>	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.

## i4blt - Basic License Tool

- 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
- Server on which the license is installed
- License annotation (if any)
- Serial number (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

## i4blt - Basic License Tool

**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
```

## i4blt - Basic License Tool

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

This option displays information about current product usage on the license servers that you specify.

#### 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**, its default value is **c**.

##### -n *server\_names*

The name of each of the license servers for which you want to display product usage information. This parameter is not used 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.

## i4blt - Basic License Tool

### **-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**

This option lists server, event, vendor, product, and user information on the license servers that you specify.

### **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 ]
```

## i4blt - Basic License Tool

### 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. 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. 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. Be sure to 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.



## i4blt - Basic License Tool

**-g *end\_date***

Specifies the end date of a report. Be sure to 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 5/21/1998 -e 2,6
```

## i4blt - Basic License Tool

### *License Use by Product Report.*

Report current license usage information on server **saturn** for the products **NetLS Test Product**, **Compiler**, **PsychoSynch**, **Megamail/2**, **EZ-Vectors**, and **DataVision**:

```
i4blt -r4 -n "saturn" -p "'NetLS Test Product' 'Compiler'
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 5/ 1/1999 -g 7/31/1999
```

## **-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***, 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.

## i4blt - Basic License Tool

### Examples

Delete all the log file entries recorded on server **neptune** before August 25, 1998:

```
i4blt -x 8/25/1998 -n neptune
```

Delete all the log file entries recorded on all nodelocked license servers before August 25, 1998:

```
i4blt -x 8/25/1998 -F 1
```

### -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 2 % 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, this is the default.

**yes** To check the threshold conditions on the products periodically, with the frequency specified with the **-X** parameter.

## i4blt - Basic License Tool

### **-X** *frequency*

Specifies the number of minutes between one license usage check and the next. Enter a value between 1 and 1440. It is mandatory if you set the **A** parameter to **yes**.

### **-l** *log*

Specifies whether or not 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 5 -A yes -X 24
```

## **-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 | a | d | s }  
[ -N cluster_name ]  
[ -T initial_number_of_servers ]  
[ -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** parameter is recommended to give you 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)

## i4blt - Basic License Tool

Cluster status:

<b>Active</b>	Running, serving licenses
<b>Change Pending</b>	Waiting for a change in the status of a server, or of the cluster, to be propagated to all the servers
<b>Inactive</b>	Not enough servers up and running
<b>Incomplete</b>	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:

<b>Serving</b>	Running, serving licenses
<b>Waiting</b>	Server is ready, but cluster is in incomplete or inactive state
<b>Unavailable</b>	Not started
<b>Reserve</b>	In reserve in case a serving server becomes unavailable
<b>Not activated</b>	Defined as a member of the cluster but 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 *action\_type* is **c**, **a**, or **d**, and it is recommended when *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 *action\_type* is **c**, and is not valid if *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 an OS/2, Windows 95, or Windows 98 machine.

If *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.

## i4cfg - Configuration Tool

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 *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 *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` (recommended only for troubleshooting purposes):

```
i4blt -H s -n pirandello
```

### -h Display Help

This option displays general syntax information for the Basic License Tool command line interface.

#### Syntax

```
i4blt -h
```

#### Example

Display the `i4blt` syntax:

```
i4blt -h
```

## i4cfg - Configuration Tool

---

### i4cfg - Configuration Tool

Use the `i4cfg` command, together with its options, as the preferred alternative to editing the `i4ls.ini` file on a Windows NT Alpha machine, or to using the Configuration Tool GUI on a non-Windows NT Alpha machine to configure your machine to perform various roles in the licensing environment. Before coding the `i4cfg` command, see “Before You Configure” on page 70 to plan your configuration requirements.

If issued with no options, the `i4cfg` command starts the Configuration Tool graphical user interface.

#### 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 machine is to play in your licensing environment. Code any combination of these values, optionally separated by commas:

- c** Reset the current role of the machine to network license client.
- n** Reset the current role of the machine to nodelocked license server.
- s** Reset the current role of the machine to network license server.
- r** Reset the current role of the machine to central registry license server.
- C** Update the current role of the machine to include network license client.
- N** Update the current role of the machine to include nodelocked license server.
- S** Update the current role of the machine to include network license server.
- R** Update the current role of the machine 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 machine will communicate in a direct binding environment. Enclose the complete list in double quotes.

## i4cfg - Configuration Tool

Specify the network license servers, nodelocked license servers, and central registry license server as follows:

```
' network protocol1:network_address1 [ port_number1 ]
protocol2:network_address2 [ port_number2 ] ...'
' nodelocked protocol1:network_address1 [ port_number1 ]
protocol2:network_address2 [ port_number2 ] ...'
' registry protocol:network_address [ port_number ]'
```

Where *protocol* is **ip**, **ipx**, **netbios**, or **os2@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 e1s` 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 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.

#### **g - License grant/release**

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

#### **v - Vendor added/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/deleted**

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



## i4cfg - Configuration Tool

### **s - Server start/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 machine belongs.

### **-h**

Displays command syntax and usage information about the Configuration Tool command-line interface.

### **-l *logfile\_path***

The path in which you want log files to be stored.

### **-list**

Displays a list of active processes.

### **-R**

Startup options that you want to disable. Code any combination of these values, optionally separated by commas:

- a** Automatic startup of processes at system startup (disabled by default)  
Automatic startup is not valid on Windows 95 or Windows 98 machines
- 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 processes at system startup (disabled by default)  
Automatic startup is not valid for Windows 95 or Windows 98 machines
- 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 machine.

### **-stop**

Stops all the processes that are running on your machine.

### **-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.

## i4cfg - Configuration Tool

The three subparameters for port numbers are positional; if you omit one, its value is reset to the default. For example:

```
"ip "  
"ip '1516,1 999,1215'"  
"ipx ',,11999'"
```

### 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, customize the path to the log files and the selection of events logged, and specify the user group:

```
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 ecegv -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 machine 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 machine named *thelma* as the central registry license server and a network license server. Configure to communicate with a network license server named *hydra*. Specify automatic startup of the servers, and specify the user 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
```

## License Use Runtime Tools

7 Display the command syntax and usage:

```
i4cfg -h
```

---

### License Use Runtime Tools

This section contains information on the following License Use Runtime and NCS tools:

#### Test Verification Tool (i4tv)

Verifies that license servers are running properly.

#### Target View Tool (i4target)

Displays the target ID of your machine. The vendor of a licensed product may ask you to provide the target ID of the machine on which the license is to be installed.

### i4tv - Test Verification Tool

Use the i4tv tool after the license servers are started to verify that they are running properly. The tool resides in the ifor\win\bin directory. A message describing a completed license transaction and a list of all license servers will be displayed.

#### Syntax

```
i4tv { [ -n hostname ] [ -v ] } | [ { -h | -usage | -version }  
[-p number_of_transactions ] }
```

#### Parameters

##### -n *hostname*

Checks that the specified machine 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.

## License Use Runtime Tools

### Example

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 i4target tool to display the target ID of your machine. The tool resides in the ifor\win\bin directory.

### Syntax

```
i4target [ -O | -V | -o | -l | -h | -d devicename | -s ] | -z ] [ -v ]
```

### Parameters

- O Displays the most secure target identifier of the machine 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 machine.
- l Displays all target IDs of the machine, starting with the most secure (the one based on the network adapter, if it is available).

For Windows 95 and Windows 98, note that if the machine is not attached to the network when the i4target command is issued, the command returns only the software-based target ID. If the machine has two or more different network cards (for example, token ring and Ethernet), the command returns the target ID based on the first card detected.

In Windows 95 and Windows 98, the NetBEUI protocol must be installed and configured properly.

For Windows NT Version 4.0 only, the hardware-based (win32mac) target ID is retrieved from the device driver that controls the network adapter, 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 machine is not connected to the network.

For this mechanism to work, the following must be installed on the machine where the licenses will be installed:

One of the following operating systems:

- Windows NT 4.0 (x86) with Service Pack 3, 4, or 5 (Service Pack 4 is recommended)
- Windows NT 4.0 Alpha with Service Pack 3, 4, or 5 (Service Pack 4 is recommended)

## License Use Runtime Tools

- Windows NT 4.0 Terminal Server Edition (x86) with Service Pack 3 or 4 (Service Pack 4 is recommended)
- Windows NT 4.0 Terminal Server Edition for Alpha with Service Pack 3 or 4 (Service Pack 4 is recommended)

A device driver 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 or NetBEUI

The i4target executable. If License Use Runtime is not yet installed on the machine, you can download i4target from the License Use Management Web site (<http://www.software.ibm.com/is/lum>), or the vendor of 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.
- s** Displays the software-based target ID. (Windows NT 4.0 only.)
- z** Displays all the devices on the machine; use it to check the available network devices before using the **-d** parameter to select a device driver manually. (Windows NT 4.0 only.)
- d *devicename***  
Specifies that the device driver *devicename* is to be used to retrieve the target ID. (Windows NT 4.0 only.)
- v** Displays information in verbose mode. On Windows NT 4.0, the information that this parameter displays includes the NDIS level of the device driver.

On Windows NT 4.0, 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:

Your machine has multiple network adapters and you prefer to use an adapter other than the one automatically selected as the basis for the target ID, or

The i4target -OI 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 these ways:

Enter the command:

```
ipconfig /all
```

For example, for IBM token ring adapter IBMTRP1, the information returned includes the line:

```
Token ring adapter IBMTRP1:
```

## License Use Runtime Processes

Go to Programs --> Administrative Tools (Common) --> Windows NT Diagnostics --> Network --> Transports, and read the adapter name following the underscore. For example, if NetBT\_IBMTRP1 is listed, the adapter name is IBMTRP1.

If NetBIOS is installed, go to Settings --> Control Panel --> Network --> Services NetBIOS Interface --> Properties, and use the last adapter name on the right.

- 2 Enter the command:

```
i4target -z
```

which displays all the devices on the machine, and be sure the network device you want to use is listed.

- 3 Enter the command:

```
i4target -d devicename
```

(For example, `i4target -d IBMTRP1`).

The entry I4DRIVER is stored in the NT Registry, in the path HKEY\_LOCALMACHINE\SOFTWARE\IBM\License Use Runtime to record the name of the network device driver used to retrieve the target ID.

---

## License Use Runtime Processes

Read this section for reference information on License Use Runtime processes.

### i4ls - Network License Server Process

The i4ls process starts the network license server on the local node. If the machine 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.)

## License Use Runtime Processes

- 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 timeout 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 timeout interval, which is how long a product can run after it has lost contact with the license server.)
- v**  
License Use 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 License Use Runtime 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 checkin, vendor, product, timeout, or message events:

```
i4ls -no cvptm
```

### i4ls -b - Nodelocked License Server Process

The `i4ls -b` process starts the nodelocked license server on the local node. If the machine 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.

## License Use Runtime Processes

### 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 *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.
- 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

License Use 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 License Use Runtime 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 nodelocked license server and do not log checkin, 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 machines:

```
i4ls -b -s
```



## License Use Runtime Processes

### i4ls -a - Central Registry License Server Process

The Central Registry is a License Use 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 License Use Runtime server:

```
i4cfg -list
```

If the machine 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 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.)
- 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

License Use 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.

## i4lct - License Creation Tool

- 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 only)

You need to run the NetBIOS socket server on any Windows NT machine configured to communicate with an OS/2 machine through the NetBIOS protocol. The OS/2 machine 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
- 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.

The i4lct command is used to create passwords. Run this command on a machine where License Use Runtime is installed.



1. The passwords you generate with the license creation tool of License Use Runtime Version 4.5.5 also work on License Use Runtime servers and clients of previous releases. High-availability licenses, introduced in Version 4.5.1, can be installed only on machines running Version 4.5.x. Licenses of types introduced in Version 4 (such as reservable and per-seat), cannot be installed on machines running earlier releases of License Use Runtime. Custom configuration licenses, introduced in Version 4.5.5, can be installed only on machines running Version 4.5.5.

In the enrollment certificate file, the *PasswordVersion* parameter is set as follows:

- 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 machines running Version 4.5.x)

## i4lct - License Creation Tool

- 5 If the password is for a license type, or exercises a policy, introduced in Version 4 (and therefore is not installable on machines running earlier versions)
- 4 Otherwise

2. To create test passwords, use **test** as the value of the **-i**, **-k**, and **-v** parameters.
3. To extract licenses for individual customers from a compound password assigned to a sales representative, use **supplier** as the value of the **-k** parameter.

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 12 on page 166 summarizes the valid combinations of license type, password use control level, password type, and enabled policies the vendor can specify with i4lct.

## i4lct - License Creation Tool

Table 12. Valid Uses of i4lct

License Type	Password Use Control Level	Password Type	Policies
Concurrent (-I c)	Customer-Managed (-R c)	Compound (-w c)	Hard Stop/Soft Stop (-A s) Multiuse Rules (-m) License Annotation (-a)
Concurrent (-I c)	Vendor-Managed (-R v)	Simple (-w l)	Multiuse Rules (-m) License Annotation (-a) Custom Configuration (-C)
Concurrent (-I c)	Vendor-Managed (-R v)	Compound (-w c)	Multiuse Rules (-m) License Annotation (-a)
Reservable (-I r)	Customer-Managed (-R c)	Compound (-w c)	Hard Stop/Soft Stop (-A s) License Annotation (-a)
Reservable (-I r)	Vendor-Managed (-R v)	Simple or Compound (-w l or -w c)	License Annotation (-a)
Use-Once (-I u)	Customer-Managed (-R c)	Compound (-w c)	License Annotation (-a)
Use-Once (-I u)	Vendor-Managed (-R v)	Simple or Compound (-w l or -w c)	License Annotation (-a)
Per-Seat (-I pt)	Customer-Managed (-R c)	Simple (-w l)	Hard Stop/Soft Stop (-A s) License Annotation (-a)
Per-Server (-I ps)	Customer-Managed (-R c)	Simple (-w l)	Hard Stop/Soft Stop (-A s) Multiuse Rules (-m) License Annotation (-a)
Simple Nodelocked (-I n)	Vendor-Managed (-R v)	Simple (-w l)	License Annotation (-a) Custom Configuration (-C)
Simple Nodelocked (-I n)	Vendor-Managed (-R v)	Compound (-w c)	License Annotation (-a)
Simple Nodelocked (-I n)	Vendor-Managed (-R v)	Compound Nodelocked (-w cn)	Try-and-Buy (-A t)* License Annotation (-a)
Concurrent Nodelocked (-I cn)	Customer-Managed (-R c)	Simple (-w l)	Hard Stop/Soft Stop (-A s) Multiuse Rules (-m) License Annotation (-a)
Concurrent Nodelocked (-I cn)	Vendor-Managed (-R v)	Simple (-w l)	Multiuse Rules (-m) License Annotation (-a)
Use-Once Nodelocked (-I un)	Customer-Managed (-R c)	Simple (-w l)	License Annotation (-a)
Use-Once Nodelocked (-I un)	Vendor-Managed (-R v)	Simple (-w l)	License Annotation (-a)

**Note:** \* When -w is set to cn, the try-and-buy attribute is required.

## i4lct - License Creation Tool

### Syntax

i4lct

#### Parameters required to generate a license:

**-i** { *vendor\_id* | **create** | **test** }  
**-k** { *vendor\_key* | **test** | **supplier** }  
**-v** { *vendor\_name* | **test** }  
**-l** *license\_type*  
**-p** *product\_id*  
**-N** *product\_name*  
**-w** *password\_type*  
{ **-d** *duration* | **-e** *expiration\_date* }  
**-r** *revision*  
**-R** *password\_registration\_level*  
{ **-T** *target\_id* **-t** *target\_type* | **-X** *extended\_target\_id* **-x** *extended\_target\_type* }

#### 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* ]

#### 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* ]  
[ **-c** *customer\_information* ]  
[ **-C** *serial\_number* ]  
[ **-L** *log\_file* ]  
[ **-O** ]  
[ **-P** *16\_bit\_flag* ]  
[ **-s** *start\_date* ]

## i4lct - License Creation Tool

### 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 80 characters long.

#### **-A** *attributes*

Possible values are:

- s** To enable the end user to modify the product policy from soft stop to hard stop and vice versa. 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**).

#### **-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*

Specifies the serial number of a custom configuration license. The serial number is a string of up to 31 alphanumeric characters that uniquely identifies a custom configuration.

#### **-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 32767.

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.

## i4lct - License Creation Tool

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 36500 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.

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*

Specify 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.

- h** Displays help for the i4lct command.

## i4lct - License Creation Tool

### **-i** *vendor\_id*

Specifies the vendor ID. It can also assume the following values:

- create** Specify it to generate a new vendor ID while generating a production password.
- test** Specify it if you are creating test passwords.

### **-k** *vendor\_key*

Specifies the vendor key. This must be an integer between 1 and 2 147 483 647, or one of the following values:

- test** Specify it if you are creating 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. 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 6pw4cilxw . n. . 3.4g.5y. . . -k supplier  
-n 497 -l c -d 1 -N "cmpLev3" -p 317 -r 1. -t any -T any  
-v "Operatix" -w 1
```

The use of this parameter is suggested when you have sales representatives in other locations. You can generate a compound password with a big number of licenses, and provide them with it. They enroll the password and then generate the licenses for customers extracting simple passwords from your compound. This will prevent you from generating the enrollment certificate files for all the customers, or from having 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 **Windows** directory.



## i4lct - License Creation Tool

### **-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), **j** (same job 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), **j** (same job ID).

For details about the **g** (same group or same display) parameter, see “Defining Rules for Multiple-Use Concurrent Licenses” on page 174.



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*

For a compound password, this is the maximum number of licenses that can be derived from the password. It is valid only for vendor-managed products (**-R** set to **v**). Its maximum allowed value is 65534. For customer-managed products, you cannot specify this parameter, and the value is set to 65535.

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 70000, 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 characters long. If it is omitted, a product name with value NULL is created by i4lct. All product name specifications must be enclosed within double quotation marks (“**product\_name**”). A product name that contains character spaces must additionally be enclosed within single 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 License Use Runtime earlier than Version 4.0, two commands are generated: the i4blt command for use with License Use 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

## i4lct - License Creation Tool

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. If this parameter is omitted, a revision with value NULL is created by i4lct.

### **-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 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.



If this option 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.



If this option is omitted, the derived start date of the password defaults to the current date.

## i4lct - License Creation Tool

### **-t** *target\_type*

The target type of the license server on which the licenses are to be installed.

Valid values are **any**, **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.

### **-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**, 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 License Use Runtime Version 4.5.x, 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 License Use Runtime Version 4.5.x, **-X** is the ID of a cluster.

## Defining Rules for Multiple-Use Concurrent Licenses

### 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. -R v  
-w 1 -l c -t any -a "Example Product" -s 1/ 1/1998 -d 365 -n 1 -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 1 -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. 2.c . 9.c8.93. . . -k 53989 -l n -p 5  
-N "Mechanical Design" -d 73 -t aix -T 152c234 -v "Mechanical Systems"  
-w 1 -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. 2.c . 9.c8.93. . . -k 53989 -l n -p 5  
-N "Mechanical Design" -d 5 -t aix -T 152c234 -v "Mechanical Systems"  
-w 1 -r 1.2 -C 85AB2215691 -a "MD2" -u
```

## Defining Rules for Multiple-Use Concurrent Licenses

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

See "Multiuse Rules" on page 12 for general information about multiuse rules.

Multiple use rules are specified for individual passwords when the software vendor runs `i4lct`, rather than in calls from the product to the license server. This means that rules are applied to individual licenses, rather than to the product itself.

The vendor can therefore specify multiple use rules for each customer, without making any changes to the product itself, and without affecting other customers' licenses for the product.

## Nodelocked Administration Tool

The following scenarios describe how the multiuse rules work when:

A license with same group rule is installed on the server (10 licenses are available on the server).

Two clients are in the same group.

### Scenario 1

1. Client1 requests 1 license; License Use Runtime shows 1 license in use
2. Client2 requests 1 license; License Use Runtime still shows 1 license in use

### Scenario 2

1. Client1 requests 5 licenses; License Use Runtime shows 5 licenses in use
2. Client2 requests 2 licenses; License Use Runtime still shows 5 licenses in use

### Scenario 3

1. Client1 requests 2 licenses; License Use Runtime shows 2 licenses in use
2. Client2 requests 5 licenses; License Use Runtime shows 7 licenses in use

When the second request in a scenario is higher than the first, License Use Runtime adds the requests, ignoring the multiuse rule.

---

## i4nat - Nodelocked Administration Tool (for Backward Compatibility Only)

In Version 4.0, the Nodelocked Administration Tool was merged with the Basic License Tool. The i4nat command is supported only for compatibility with versions of License Use Runtime earlier than Version 4.0.



If the Nodelocked Administration Tool is not available to you, edit the nodelock file manually, using an ASCII text editor. The default location of the file is:

```
\ifor\ls\conf\nodelock
```

If this file is not in the default directory, check your product documentation or contact the product vendor.

If issued with no options, the **i4nat** command starts the Nodelocked Administration Tool interactive interface. You can enter the following primary command options:

**-a (Add a Nodelocked License)**

Add a nodelocked license for a given product to the nodelock file.

**-d (Delete a Nodelocked License)**

Delete a nodelocked license for a given product from the nodelock file.

**-l (Display License Information)**

List either one or all nodelocked licenses enrolled in the nodelock file.

## Nodelocked Administration Tool

### **-u (Update a Concurrent Nodelocked License)**

Update the number of concurrent users of a concurrent nodelocked license.

### **-h (Display Interface Usage)**

Display command syntax and usage information about the Nodelocked Administration Tool command line interface.

To get help with the Nodelocked Administration Tool command line interface, you can also enter the following command:

```
man i4nat
```

### **-a Add a Nodelocked License**

Use this command to add a new nodelocked license to the nodelock file.

If the vendor provides you with the nodelocked license in the form of an enrollment certificate file, you can add the license automatically. Otherwise, you have to add it manually.

### **Syntax**

If you have the enrollment certificate file:

```
i4nat -a -f filename -c
```

If you do not have the enrollment certificate file:

```
i4nat -a
```

```
-v vendor_name vendor_id
```

```
-p product_name product_version license_password [license_annotation comment]
```

```
-c count
```

### **Parameters**

#### **-f filename (For automatic entry only)**

The complete path and file name of the enrollment certificate file.

#### **-v vendor\_name (For manual entry only)**

The name of the vendor who manufactured the product whose license is being added. Vendor name specifications are case-sensitive.

**vendor\_id**

The unique vendor ID string for the vendor specified in the **vendor\_name**.

#### **-p product\_info (For manual entry only)**

The information on the licensed product that you intend to install.

#### **product\_name (For manual entry only)**

The name of the licensed product that you have to add. Product name specifications are case-sensitive.

**product\_version**

The version of the product that is specified in the **product\_name** argument

## Nodelocked Administration Tool

### ***license\_password***

The unique license password string associated with the product.

### ***license\_annotation***

The license annotation information (if any) the vendor provided.

### ***comment***

Any comment that you want to add to the nodelocked license record within the nodelock file.

### **-c *count* (For concurrent nodelocked licenses only)**

The number of users you want to use the license simultaneously.

## Examples

Add a new product:

```
i4nat -a
-v "vendor_name vendor_id"
-p "product_name product_version license_password [ license_annotation ]"
-c 2
```

## **-d Delete a Nodelocked License**

Deletes a nodelocked license from the nodelock file. The license is identified by its unique password.

### **Syntax**

```
i4nat -d -p license_password
```

### **Parameters**

#### **-p *license\_password***

The license password that uniquely identifies the license that you want to delete.

## Examples

The following command deletes the license to use a product with password:

```
2ap6tesiawwrs7qkd4y9wthzx6mj22i2
```

```
i4nat -d -p 2ap6tesiawwrs7qkd4y9wthzx6mj22i2
```

## Nodelocked Administration Tool

### -l Display License Information

Displays information regarding a license if the -p option is specified; otherwise, it displays a list of all the licenses installed on the nodelock file.

#### Syntax

```
i4nat -l [ -p license_password ]
```

#### Parameters

**-p *license\_password***

The license password that uniquely identifies the license that you want to display.

#### Examples

The following command displays the license to use a product with password:

```
2ap6tesiawwrs7qkd4y9wthzx6mj22i2
```

```
i4nat -l -p 2ap6tesiawwrs7qkd4y9wthzx6mj22i2
```

### -u Update Concurrent Nodelocked License Information

Updates the number of concurrent users of a concurrent nodelocked license.

#### Syntax

```
i4nat -u -p license_password -c count
```

#### Parameters

**-p *license\_password***

The license password that uniquely identifies the concurrent nodelocked license you want to update.

**-c *count***

The number of users you want to simultaneously use the licensed product. If you set this parameter to 0 the license becomes a simple nodelocked license, that is it is no more a concurrent nodelocked license, and cannot be used by concurrent users.

#### Examples

The following command updates the number of concurrent users of the license for the product with password: *2ap6tesiawwrs7qkd4y9wthzx6mj22i2*, and sets that number to 10.

```
i4nat -u -p 2ap6tesiawwrs7qkd4y9wthzx6mj22i2 -c 1
```



## Nodelocked Administration Tool

### **-h Display Command Line Interface Usage**

Displays syntax and usage information for the command line interface.

#### **Syntax**

`i4nat -h`

#### **Examples**

The following command displays the command line syntax and usage:

```
i4nat -h
```



---

## Chapter 6. Hints and Tips

Read this chapter to better manage your licensing environment.

---

### Managing Time Zone

To make it possible for License Use Runtime to display the correct local time in administration tool panels, you must set the environmental variable TZ on Windows.

In Windows 95 and Windows 98, add the following line in the autoexec.bat file:

```
SET TZ='value'
```

where 'value' is the same as that in step 3 in the Windows NT procedure that follows.

In Windows NT:

- 1** Go to **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** are characters you choose to remind you of the setting; **s** is + or -; and **nn** is the difference in hours from Greenwich mean time. A positive difference means west, and a negative difference means east. For example, you might specify Italian standard time (one hour east of Greenwich mean time) as:

```
IST-1GMT
```

---

### 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.

License Use Runtime implements a backup procedure of all databases on license server machines.

### Causes for Corrupted Definition or Database Files

Since License Use 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)

## Recovery Procedure

When a License Use Runtime database is corrupted, after the database has been recovered, try to find out the real cause of the problem. Consider that the contents of the definition and database files used by License Use Runtime are changed only by defined administrative commands and tools.

### Automatic Backup Procedure

License Use 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), or 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, "License Use Runtime Configuration File" on page 199.



Be 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":

- 1 Stop the License Use 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:** In case of corruption, 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.

## Managing the Reports Log Files

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

You can run the backup procedure manually by running:

```
\ifor\ls\conf\db_back.cmd
```

On a machine configured only as a network license client, only manual backup is available.

The command copies the file and databases on a backup disk (the a: drive).

### Manual Recovery

To start the recovery procedure in case of corruption, use:

```
\ifor\ls\conf\db_recov.cmd
```

This command restores the files and databases that were saved by the db\_back command, from the a: drive.

---

## Managing the Reports Log Files

When you ask for a report, the Basic License Tool reads the current log files:

```
\ifor\ls\conf\logdbnn_  
\ifor\ls\conf\crlognn_  
\ifor\ls\conf\llmlgnn_
```

The names of the current log files end with an underscore.

The files logdbnn, crlognn, and llmlgnn contain all the collected License Use 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 73 and “Scenario 3: Configuring a Network License Server” on page 79. *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, License Use 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, “License Use Runtime Configuration File” on page 199.

The numbering of log files starts from 00. Suppose you have the following files on the machine:

```
logdb  
logdb 1_  
logdb 2
```

## Managing a Custom Configuration

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 logdbnn\_, crlognn\_, and llmlognn\_ if they exist on the server.

---

## 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 7. Troubleshooting

This chapter provides suggestions for improving performance, problem determination, and debugging when using products managed with License Use Runtime. This chapter assumes you have read the preceding chapters in this book. It suggests steps you can take should certain problems occur:

- At a local machine running products with nodelocked licenses
- Using various types of network licenses
- Running License Use Runtime processes
- With performance
- With the binding between servers and clients
- With network protocols and hardware

---

### 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 ID (TargetID 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: 7=new in Version 4.5.5, 6=new in Version 4.5.1, 5=new in Version 4.0, 4=not new in Version 4)

## Troubleshooting

For example, this is the enrollment certificate file for the DataMaster product that was used as an example in “Scenario 7: Managing Reservable Licenses” on page 102:

```
i4blt -a -v "'IBM Corporation' 8499f53d66dd.8d. 1.51.32.4c. . . gm898vcvtpiq8"  
-p "'DataMaster' '2.1a' qj4y2zjivvr9ryffuw8x9se48vvaaaa "
```

```
[LicenseCertificate]  
Checksum=7B33C C 71 128534 916679A859 54  
TimeStamp=898711 18  
PasswordVersion=5  
VendorName=IBM Corporation  
VendorPassword=gm898vcvtpiq8  
VendorID=8499f53d66dd.8d. 1.51.32.4c. . .  
ProductName=DataMaster  
ProductID=2222  
ProductVersion=2.1a  
ProductPassword=qj4y2zjivvr9ryffuw8x9se48vvaa  
ProductAnnotation=  
LicenseStyle=reservable  
LicenseStartDate= 6/24/1998  
LicenseDuration=14436  
LicenseEndDate=12/31/2 37  
LicenseCount=1  
MultiUseRules=  
RegistrationLevel=3  
TryAndBuy=No  
SoftStop=No  
TargetType=ANY  
TargetTypeName=Open Target  
TargetID=ANY  
ExtendedTargetType=  
ExtendedTargetID=  
SerialNumber=  
Upgrade=No  
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 two lines because of space constraints.



---

### 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 94 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 machine 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 machine 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 machine 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:

```
\ifor\ls\conf\nodelock
```

If the file is not in the default directory, check your product documentation or contact the product vendor.

#### **If the product uses runtime-based enabling:**

Check that the nodelocked license server (i4llmd) is up and running (see “Starting Required Processes” on page 190).

## Troubleshooting

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=2
```

---

### 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 machine 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 machines where you have the licenses installed.
- If `i4tv` does display active servers, check that they include a machine 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.

License Use 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. License Use 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 machine. 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 190).

Check that the client machine can reach the central registry.

Check that the nodelocked license server is up and running at the client (see “Starting Required Processes” on page 190).

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, be sure the nodelocked license server is running on the machine where the application server runs (see “Starting Required Processes” on page 190).

If an application with per-seat licenses fails to start:

- Be sure the central registry license server is up and running, and that the nodelocked license server is running on the machine where the application server runs (see “Starting Required Processes” on page 190).
- Be 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 106).
- Be sure the machine 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:

Be 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 190).

Be sure the machine where you are working can reach the central registry.

## Troubleshooting

If you received the licenses in a compound password, make sure you have distributed the licenses. See “Distributing the Licenses” on page 97 for information on how to distribute licenses.

If soft stop does not work, be 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 machine where the license is installed. To get the target ID of the machine, run the i4target tool (“i4target - Target View Tool” on page 158) on that machine.

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 machine.
- The target ID of your target machine has changed.

---

### 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 37.

---

### 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 License Use Runtime Processes

This section covers problems that could arise if License Use 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 License Use Runtime process is not running.

## Troubleshooting

To get a list of the License Use Runtime processes that are running on a machine, use the `i4cfg -list` command or double-click on the **Service Manager Tool** icon in the License Use Runtime folder. The names of the processes are shown in Table 13 on page 191. For an overview of which license servers are required for each license type, see Table 10 on page 69.

Table 13. License Use Runtime Processes

Process	Name
Nodelocked License Server	i4llmd
Network License Server	i4lmd
Central Registry License Server	i4gdb
NetBIOS socket server*	i4nbssvr

**Note:** \*NetBIOS is not supported under Windows NT Alpha.

To start the processes, select **Start** from the **Service** pull-down menu.

If your process still does not start, check the error messages in the `i4ls.log`, `i4lmd.log`, `i4llmd.log`, and `i4gdb.log` files in the `\ifor\ls\conf` 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 5, "License Use Runtime Commands" on page 127). 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 machine 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

---

### Troubleshooting Custom Configuration Licenses

#### Cannot Install a Custom Configuration License

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 `i4tv` (test verification) tool 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 5, “License Use Runtime Commands” on page 127.

#### 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 you 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 machine, including *louise* itself, must contain the entries:

```
ip:louise[1 999]
ip:louise[1515]
```

Note that the port numbers must match the values of the `ipPort` and `ipGDBPort` parameters in the configuration file of *louise*.

---

### Troubleshooting Communication Protocols

If you have communication problems, first verify that the protocol you use is installed on the machine and configured properly, as described in “Protocol Configuration Requirements” on page 50.

## Troubleshooting

### NetBIOS

The NetBIOS protocol is not supported on Windows 95, Windows 98, Windows NT Alpha, or Windows Terminal Server Alpha machines. If you use NetBIOS under another operating system, the following brief checklist should help you make sure your NetBIOS system is OK:

- 1 Make sure you entered the machine names correctly when you configured the direct binding servers list at license servers and clients of your environment. Note that the machine name and the server name fields of the Configuration Tool are case-sensitive:
  - On Windows NT (x86), the machine names are the computer names. They must be entered in uppercase.
  - On OS/2, the machine names must be entered in lowercase.
- 2 Check that the NetBIOS socket server is up and running on every machine that uses NetBIOS to communicate with an OS/2 machine.
- 3 Check that the NetBIOS socket server is up and running on every Windows NT (x86) machine that your system needs to communicate with. For instructions, refer to “Running the NetBIOS Socket Server” on page 45.
- 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 50).

### TCP/IP

If you use TCP/IP, the following is a brief checklist to help you make sure your TCP/IP system is OK:

- 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 Be 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.

## Troubleshooting

- 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 License Use Runtime are based on universal datagram protocol (UDP). In a very highly loaded network, UDP connections may receive timeouts before data is delivered. This is normal behavior; you need to reduce the total network load.
- 9 If your License Use Runtime server is on a machine configured with the loopback facility of TCP/IP, when you use the `i4blt -a` command to enroll a product on the local server, you must specify:  

```
-n ip:localhost
```

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 Coexistence with Other Products

This section explains how to handle problems relating to the coexistence of License Use Runtime with other products.

### Coexistence with Digital DCE for Windows NT

If you install Digital's DCE for Windows NT while the License Use Management client is running, one or both of the following messages may be displayed:

```
Unable to delete file  
c:\PROGRA~1\DCE\dcelocal\locale\enus1252.dll  
Access denied.
```

```
Unable to delete file  
c:\PROGRA~1\DCE\dcelocal\locale\enus437.dll  
Access denied.
```

These messages can be safely ignored. License Use Management client also accesses these shared libraries, which will not be removed from the system. This is the correct behavior.

---

## Troubleshooting the Hardware

Following is a brief checklist to help you make sure your hardware is OK.

- 1 If you get the error message:  

```
Invalid target ID
```

check that your target ID has not changed.



## Troubleshooting

- 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 other 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

In order to help IBM help you in problem determination, you should gather additional information to send to your IBM representative when you request support. License Use Runtime processes and tools can be run in traced mode as explained in the following sections.

### Running Processes in Traced Mode

To run License Use 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
```

- 2 Start the trace 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
```

and from the same window, restart the processes by issuing the following command:

```
\ifor\win\bin\i4cfg -start
```

#### To collect the trace data:

- 4 Produce the error condition again.
- 5 Stop all active processes again. The output trace information is collected in the files with extension `trc`.

## Troubleshooting

### To view the trace data using the GUI:

- 6 In the Service Manager Tool window select **Stop** from the **Service** pull-down menu.
- 7 Select the line corresponding to a specific process in the list.
- 8 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           (for only the x86 platform)
```

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           (for only the x86 platform)
```

Then start the tools from the same window.

### Collecting Other Data

Other information data concerning License Use Runtime servers is automatically collected by the license server (i4ls) process. This data is stored in the 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 configuration file, i4ls.ini. The file is located in the directory where the operating system is installed (by default, WINNT on Windows NT, WINSRV on Windows Terminal Server, and windows on Windows 95 and Windows 98.)

## Troubleshooting

The user file:

```
\ifor\ls\conf\userfile
```

License Use 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 (x86):

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 machine logging in the `WEBSPHERE_root\logs\ncf.log` file, change both `ncf.jvm.stdoutlog.enabled` and `ncf.jvm.stdoutlog.file` from `false` to `true`.

### 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

---

### Troubleshooting Java Client Support Uninstallation

If you uninstall License Use 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 License Use Runtime Communication component without encountering problems, make sure that no License Use Management environment variables are set. To remove environment variables, do one of the following:

- After uninstalling License Use Runtime, shut down and restart the machine.

- Use regedit to remove the License Use Management environment variable from the registry.

- Remove the License Use Management environment variable from the Environment tab in the System Properties window.

---

## Appendix A. License Use Runtime Configuration File

This appendix describes the License Use Runtime i4ls.ini configuration file. The file is located in the directory in which the operating system is installed (by default, WINNT on Windows NT (x86 or Alpha), WTSRV on Windows Terminal Server (x86 or Alpha), and Windows on Windows 95 and Windows 98). You should normally use the configuration tool to configure License Use 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**).

### **[/FOR/LS Machine-Configuration]**

#### **ConfigureAs=client**

*Obsolete; supported only for backward compatibility with earlier versions of License Use Runtime.*

Specifies whether the machine 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 machine.

#### **GroupName=**

The name of the group to which the user of this machine belongs.

#### **DebugProc=no**

Specifies whether or not the License Use 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 License Use 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 License Use Runtime processes are started in debug traced mode and the output is redirected to a file. Possible values

## Configuration File

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 License Use Runtime.*

Specifies whether or not the License Use 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 License Use 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 License Use Runtime.*

Maximum number of messages logged by the concurrent nodelock process

### **LogFile=**

*Obsolete; supported only for backward compatibility with earlier versions of License Use Runtime.*

Log file path and name of the concurrent nodelock process log.

### **CommunVersion=V4R5**

Version of License Use Runtime communication subcomponent.

### **RuntimeVersion=V4R5**

Version of License Use Runtime runtime subcomponent.

### **NCSSupportVersion=**

*Not used on the Windows platform.*

### **Communication=yes**

The machine is configured to communicate in a network.

### **NamespaceBindingSupport=**

*Not used on the Windows platform.*

### **AdvancedConfiguration=no**

The user selected **Advanced Configuration** when configuring the machine.

## Configuration File

### **[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 License Use 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.

This parameter value must be the same on all servers within your licensing environment. This parameter 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. This parameter 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.

## Configuration File

### **NumberOfLogFile=2**

The number of log files License Use Runtime writes. For example, if logdb is the log file name, and **NumberOfLogFile** is set to 2, License Use 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, License Use 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 11999 is already used for some other purpose. If you change this value, change it on cluster members.

#### **netbiosPort=115**

The NetBIOS port number the license server listens to when supporting its clients.

*Not used on the Windows NT Alpha platform.*



## Configuration File

### **netbiosGDBPort=109**

The NetBIOS port number the central registry license server listens to when supporting its clients.

*Not used on the Windows NT Alpha platform.*

### **netbiosNDLPort=12999**

The NetBIOS port number the nodelocked license server listens to for remote administration.

*Not used on the Windows NT Alpha platform.*

### **ipxPort=1515**

The IPX port number the license server listens to when supporting its clients.

*Not used on the Windows NT Alpha platform.*

### **ipxGDBPort=10999**

The IPX port number the central registry license server listens to when supporting its clients.

*Not used on the Windows NT Alpha platform.*

### **ipxNDLPort=12999**

The IPX port number the nodelocked license server listens to for remote administration.

*Not used on the Windows NT Alpha platform.*

### **RunGLBD=**

*Not used on the Windows platform.*

### **RunGDB=no**

Whether the central registry license server is to be started on this machine. Possible values are **yes** and **no**.

### **DisableRemoteAdmin=no**

Whether or not 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 or not 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 or not 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 License Use Runtime.*

Log file path and name of the license server process log.

## Configuration File

**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 machine. Possible values are **yes** and **no**.

**RunLMD=no**

Whether the network license server is to be started on this machine. Possible values are **yes** and **no**.

**UseHostTable=no**

*Not used on the Windows platform.*

**PassiveTime=300**

*Not used on the Windows x86 or Windows Alpha platform.*

**MaxActivities=512**

*Not used on the Windows x86 or Windows Alpha platform.*

**MaxActivitiesThreshold=100**

*Not used on the Windows x86 or Windows Alpha platform.*

**TraceActivities=no**

*Not used on the Windows x86 or Windows Alpha 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**.

## Configuration File

### **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 machine uses to communicate with the NetBIOS protocol. (Windows NT x86 only)

#### **NCBS=**

*Not used on the Windows NT x86 or Windows NT Alpha platforms.*

#### **HasOS2Clients=no**

Change it to **yes** on a Windows NT machine that must support OS/2 clients using NetBIOS.

*Not used on the Windows NT Alpha platform.*

### **[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.

*Not used on the Windows NT Alpha platform.*

#### **Threshold\_Automatic=0**

Whether or not the threshold is automatically checked periodically. Possible values are **0** (no) or **1** (yes).

*Not used on the Windows NT Alpha platform.*

#### **Refresh\_Automatic=0**

Whether or not the Basic License Tool GUI is to do a periodic automatic refresh. Possible values are **0** (no) or **1** (yes).

*Not used on the Windows NT Alpha platform.*

#### **Refresh\_Frequency=60**

The frequency, in minutes, of the automatic refresh.

*Not used on the Windows NT Alpha platform.*

## Configuration File

### **ReadTimeout=20**

*Not used on the Windows NT Alpha platform.*

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 machine is poor. This parameter can be changed only by editing the configuration file.

### **[iFOR/LS NCS-Client]**

#### **UseDirectBindingOnly=yes**

Whether or not the client licensed applications running on this machine are to locate the license servers using direct binding only. Always **yes** on the Windows platform.

#### **FilterNDL=no**

Whether or not nodelocked licenses are to be excluded from the set of licenses administered by the Basic License Tool.

#### **FilterNet=no**

Whether or not 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:

<b>ip</b>	Any platform
<b>netbios</b>	Windows NT x86 or Windows Terminal Server x86
<b>ipx</b>	Windows NT x86, Windows Terminal Server x86, or Windows 95

#### **DirectBindServer2=ip:louise.rnsl.ibm.com[1515]**

The format is *family:servername[port]*.

where *family* can be:

<b>ip</b>	Any platform
<b>netbios</b>	Windows NT x86 or Windows Terminal Server x86
<b>ipx</b>	Windows NT x86, Windows Terminal Server x86, or Windows 95

## Configuration File

### **DirectBindServer3=ip:louise.rnsl.ibm.com[10999]**

The format is *family:servername[port]*.

where *family* can be:

<b>ip</b>	Any platform
<b>netbios</b>	Windows NT x86 or Windows Terminal Server x86
<b>ipx</b>	Windows NT x86, Windows Terminal Server x86, or Windows 95

### **OS2NumServers=0**

The number of OS/2 NetBIOS license 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:

<b>ip</b>	Any platform
<b>netbios</b>	Windows NT x86 or Windows Terminal Server x86
<b>ipx</b>	Windows NT x86, Windows Terminal Server x86, or Windows 95

### **NumDirectBindNDLServers=2**

The number of nodelocked license servers whose licenses can be administered remotely from this machine.

### **DirectBindNDLServer1=ip:lab68082.rome.lab.tivolicom[12999]**

The format is *family:servername[port]*.

where *family* can be:

<b>ip</b>	Any platform
<b>netbios</b>	Windows NT x86 or Windows Terminal Server x86
<b>ipx</b>	Windows NT x86, Windows Terminal Server x86, or Windows 95

### **DirectBindNDLServer2=ip:lab68084.rome.lab.tivolicom[12999]**

The format is *family:servername[port]*.

where *family* can be:

<b>ip</b>	Any platform
<b>netbios</b>	Windows NT x86 or Windows Terminal Server x86
<b>ipx</b>	Windows NT x86, Windows Terminal Server x86, 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 file `\ifor\ls\conf\nodelock`

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 80 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 12/25/2 3
543b f87c 93. 2.81.87.92.34. . . gganccupqb5dauxabdw "" "2. " "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. Features and Functions Added in Version 4

Table 14 lists the features and functions that have been added to License Use Management in Version 4.

**Attention:** Do not use obsolete commands or APIs, which are supported for backward compatibility, with these newer features and functions.

*Table 14. Features and Functions Added in Version 4*

	<b>Feature or Function</b>	<b>Page</b>
License types	Concurrent nodelock	8
	Use-once nodelock	8
	Per server	9
	Reservable	10
	Per seat	11
License policies	Try and buy	12
	Custom configuration	13
Server enhancements	Hard-stop/soft-stop selection	13
	Customer-managed use control	7
	High-water mark	18
	Threshold	18
	High-availability licensing	20
NCS enhancement	Direct binding	37



---

## Glossary

### A

**ACID.** See *application client identifier*.

**application client identifier.** 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 License Use 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 License Use 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 License Use 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*.

### C

**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 global location broker

**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 a 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 License Use Runtime with the help of the administrator.

Configuring License Use 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.

## hard stop license information

### 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 Runtime supports the following protocols on OS/2:

- NetBIOS
- TCP/IP
- IPX

License Use Runtime supports the following protocols on Windows NT:

- NetBIOS
- TCP/IP
- IPX

On Windows 95 and Windows 98, NetBIOS is not supported. On Windows 98 and Windows NT Alpha, IPX is not supported.

On AIX, HP-UX, IRIX, Solaris, and Windows NT Alpha License Use Runtime 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

## license key non-runtime-based enablement

licenses, license type, start and end dates for the licenses, and a time stamp.

**license key.** See *password*.

**license password.** 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).

**NetBIOS.** A transport protocol for PC local area networks, which was introduced with the IBM PC Network and implemented in Microsoft MS-Net and LAN Manager. Application programs use NetBIOS for client/server or peer-to-peer communications.

**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 License Use 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 License Use Runtime client.

**network license server.** In License Use Management, a node in the network on which network licenses are stored for use by License Use 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 License Use 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.

## object simple nodelocked 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.

**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 License Use 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.

**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 vendor-managed use control

**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 License Use Management 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 License Use 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 License Use 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

### A

- access restriction, user
  - purpose 14
  - scenario 113
- administer high-availability licensing (i4blt -H) 150
- administration tool (Basic License Tool)
  - command 128
  - enabling remote administration, direct binding 37
  - interfaces, overview 18
  - overview 18
  - performance 190
  - purpose 3
  - remote administration 18
  - scenarios 94
  - selecting servers 93
  - starting 92
- administrator
  - definition 213
- AIX releases supported 5
- annotation, license
  - definition 215
  - purpose 13
- applets, Java
  - concurrent licenses 26
  - license-enabling supported 1
  - per-seat licenses 33
  - reservable licenses 29
  - restricted to network licenses 3
  - use-once licenses 24
- application client
  - definition 213
  - per-server/per-seat licensing 9
- application client identifier (ACID)
  - definition 213
- Application Developer's Toolkit overview 1
- application server
  - definition 213
  - per-server/per-seat licensing 9
- archiving log files 183
- automatic backup 182
- automatic startup of processes
  - configuring with GUI 74
  - configuring with i4cfg command 155
  - planning 71
- available licenses 93

### B

- backup procedure
  - automatic 182
  - manual 183
  - overview 21
  - using 181
- BackupMode parameter of configuration file 182
- BackupPath parameter of configuration file 182
- Basic License Tool
  - command 128
  - definition 213
  - enabling remote administration, direct binding 37
  - interfaces, overview 18
  - overview 18
  - performance 190
  - purpose 3
  - remote administration 18
  - scenarios 94
  - selecting servers 93
  - starting 92
- binding
  - See direct binding
- books, online xvi

### C

- CD-ROM
  - getting License Use Runtime from 50
  - images for code server installation 62
  - LUM Java Client Support 64
- central registry license server
  - automatic start 71
  - cannot be moved 40
  - components required 48
  - configuration 88
  - configuration options 70
  - configuring with i4cfg command 153
  - definition 213
  - direct binding on 89
  - enrolling custom configuration products on 126
  - enrolling customer-managed use products on 95
  - log file 183
  - must be unique 40
  - overview 21
  - planning 40

- central registry license server (*continued*)
  - process 163
  - required for customer-managed use products 21
  - required for reservable licenses 21
  - selecting machines 35
  - selecting with Basic License Tool 93
- certificate file, enrollment
  - checking for password version 165
  - command to enroll a password 101
  - example 186
  - for per-server/per-seat 14
  - importing 94
  - purpose 5
- changes in third edition, summary xix
- check period 191
  - definition 213
- clean up stale licenses (i4blt -C) 138
- client, network license
  - components required 48
  - configuration 84
  - configuration options 70
  - configuring with i4cfg command 153
  - direct binding on 85
- client, setting up 67
- cluster
  - activating members 116
  - adding members 117
  - creating 114
  - creating, i4blt command 150
  - deactivating members 117
  - definition 213
  - deleting 42
  - direct binding considerations 45
  - enrolling licenses on 116
  - ID 116
  - membership 41
  - number of servers in 42
  - overview 20
  - password 41
  - planning 41
  - removing licenses from 122
  - replacing a server in 42
  - scenario 114
  - size 41
- code server installation 62
- cold start 191
- command line interface
  - configuration tool 73
  - custom configuration, upgrading 126
  - hard stop/soft stop 112
- command line interface (*continued*)
  - i4blt 128
  - i4cfg 152
  - i4lct 164
  - i4ls 160
  - i4ls -a 163
  - i4ls -b 161
  - i4nat 175
  - i4target 158
  - i4tv 157
  - installation 59
  - managing a licensed product 101
  - managing reservable licenses 105
  - reference documents xvii
  - switching from per-server to per-seat 109
- Communications component
  - disk requirements 49
  - required for LUM Java Client Support 64
  - required, by machine role 48
- components
  - required, by machine role 48
  - selecting 48
- compound password
  - definition 213
  - distributing licenses from 97
  - purpose 5
  - use by sales representatives 164
  - use for vendor-managed use products 6
- concurrent license
  - configuration required 69
  - definition 213
  - Java applications and applets 26
  - operation 26
  - purpose 10
  - troubleshooting 188
- concurrent nodelocked license
  - configuration required 69
  - definition 214
  - purpose 8
- configuration
  - central registry license server 88
  - command 152
  - command line interface 73
  - direct binding 71
  - GUI 72
  - interfaces, overview 17
  - network license client 84
  - network license server 79
  - nodelocked license server in a network 75
  - nodelocked license server, standalone 73

- configuration (*continued*)
  - options 70
  - required for each license type 68
  - scalability 16
  - starting 72
- configuration file (i4ls.ini)
  - BackupMode parameter 182
  - BackupPath parameter 182
  - ColdStart parameter 191
  - DebugNCS parameter 195
  - DebugProc parameter 195
  - DebugToFile parameter 195
  - directory 199
  - ipGDBPort parameter 192
  - ipPort parameter 192
  - layout 199
  - MaxLogFileSize parameter 183
  - NumberOfLogFile parameter 183
  - ReadTimeout parameter 188
  - TraceActivities parameter 195
- configuration tool
  - interface types 67
- CPU requirements 49
- crlognn file
  - managing 183
- custom configuration
  - deleting keys 184
  - deleting products or reducing numbers 184
  - enrolling products in central registry 126
  - overview 13
  - requesting a license upgrade 184
  - tips on managing 184
  - upgrading 123
- custom configuration license
  - troubleshooting 192
- customer-managed use control
  - central registry required 21
  - definition 214
  - enrolling products on central registry 95
  - overview 7
  - RegistrationLevel tag, enrollment certificate file 185
  - troubleshooting 189
- customization log 70

## D

- date, synchronizing 188
- DebugNCS parameter of configuration file 195
- DebugProc parameter of configuration file 195

- DebugToFile parameter of configuration file 195
- deinstallation
  - License Use Runtime 63
  - LUM Java Client Support 64
- delete a product license (i4blt -d) 135
- delete server log entries (i4blt -x) 148
- direct binding
  - configuration on nodelocked license server 77
  - configuration options 71
  - default ports 72
  - definition 214
  - enabling remote administration 37
  - high-availability considerations 45
  - Java Client Support example 39
  - network example 38
  - on central registry license server 89
  - on network license client 85
  - on network license server 79
  - performance 39
  - performance considerations 39
  - planning 39
  - ports 37
  - servers list 37
  - specifying with i4cfg command 153
  - troubleshooting 192
- direct binding server list
  - definition 214
- directory, installation 54
- disk requirements
  - License Use Runtime 49
  - LUM Java Client Support 63
- display a list (i4blt -l) 139
- display command line interface usage (i4blt -h) 152
- display product license status (i4blt -s) 144
- distribute licenses (i4blt -E) 134
- distributing licenses from a compound password 97
- Documentation component
  - contents 48
  - disk requirements 49
- download site
  - overview 5
  - using 51
- drive for installation 54
- dynamic nodelocking
  - definition 214

## E

- enablement, license
  - command to enroll a password 101

- enablement, license (*continued*)
  - importing 94
  - models 15
  - nodelocked licenses 7
  - publication xvii
  - purpose 1
- end user
  - definition 214
- enrolling a product
  - on a cluster 122
  - scenario 94
  - using `i4blt -a` command 130
- enrollment certificate file
  - checking for password version 165
  - definition 214
  - example 186
  - for per-server/per-seat 14
  - importing, scenario 94
  - purpose 5
- error log data, collecting
  - running enabled applications in traced mode 196
  - running processes in traced mode 195
  - running tools in traced mode 196

## F

- FTP site
  - overview 5
  - using 51

## G

- generate a report (`i4blt -r`) 145
- group, user
  - specifying with `i4cfg` command 155
- GUI (graphical user interface)
  - Basic License Tool 92
  - configuration tool 72
  - installation 51

## H

- hard stop policy
  - definition 215
  - in enrollment certificate file 185
  - purpose 13
  - scenario 109
- hardware
  - requirements 49
  - troubleshooting 194

- high-availability licensing
  - cluster membership 41
  - cluster size 41
  - deactivating a server 120
  - definition 215
  - deleting a cluster 42
  - direct binding considerations 45
  - enrolling licenses 122
  - `i4blt` command 150
  - overview 20
  - password version 164
  - planning 41
  - removing licenses 122
  - scenario 114
  - viewing licenses 121
- high-water mark
  - definition 215
  - purpose 14
  - resetting 111
- HP-UX support 5
- HTM files
  - command reference xvii
  - installation 48
  - message reference xvii
  - Using Application Developer's Toolkit xvii
  - Using License Use Runtime xvii
  - viewing xvi

## I

- `i4blt`
  - administer high-availability licensing (`i4blt -H`) 150
  - clean up stale licenses (`i4blt -C`) 138
  - delete a product license (`i4blt -d`) 135
  - delete server log entries (`i4blt -x`) 148
  - display a list (`i4blt -l`) 139
  - display command line interface usage (`i4blt -h`) 152
  - display product license status (`i4blt -s`) 144
  - distribute licenses (`i4blt -E`) 134
  - enroll a product (`i4blt -a`) 130
  - examples 92
  - generate a report (`i4blt -r`) 145
  - log threshold events 149
  - monitor threshold events 149
  - reserve licenses (`i4blt -R`) 137
  - starting the Basic License Tool 92
  - update enrolled licenses (`i4blt -U`) 132
- `i4cfg` 152
  - See *also* configuration tool

- i4lct 167
  - defining rules for multiple-use concurrent licenses 174
  - examples 174
  - format 167
  - license for production passwords 17
  - options 168
  - overview 17
  - usage 164
- i4ls 160
- i4ls -a 163
- i4ls -b 161
- i4ls.ini configuration file
  - BackupMode parameter 182
  - BackupPath parameter 182
  - ColdStart parameter 191
  - DebugNCS parameter 195
  - DebugProc parameter 195
  - DebugToFile parameter 195
  - directory 199
  - ipGDBPort parameter 192
  - ipPort parameter 192
  - layout 199
  - MaxLogFileSize parameter 183
  - NumberOfLogFile parameter 183
  - ReadTimeout parameter 188
  - TraceActivities parameter 195
- i4nat 175
- i4target 158
- i4tv 157
- ID, cluster 116
- ifor directory 54
- installation
  - command 59
  - interactive 51
  - interactive, after uninstallation 57
  - LUM Java Client Support 63
  - noninteractive 58
  - scalability 16
  - selecting components 48
  - silent 59
- interactive installation 51
- interprocess communication failure 188
- ipGDBPort parameter of configuration file 192
- ipPort parameter of configuration file 192
- IPX
  - configuration 50
  - definition 215
  - from a code server 62
  - requirements 49

IRIX support 5

## J

- Java applications and applets
  - concurrent licenses 26
  - license-enabling supported 1
  - per-seat licenses 33
  - planning for 40
  - reservable licenses 29
  - restricted to network licenses 3
  - troubleshooting 197
  - use-once licenses 24
- Java Client Support
  - direct binding example 39
  - installation 63
  - obtaining code 64
  - troubleshooting 197
  - uninstalling 64, 198
- Java Development Kit 64
- Java Runtime Kit 64

## K

- key
  - See password

## L

- language support xvii
- license
  - available 93
  - definition 215
  - publication xvii
  - purpose 2
  - types 7
- license administration tool
  - command 128
  - enabling remote administration, direct binding 37
  - interfaces, overview 18
  - overview 18
  - performance 190
  - remote administration 18
  - scenarios 94
  - starting 92
- license annotation
  - definition 215
  - purpose 13
- license creation tool
  - defining rules for multiple-use concurrent licenses 174

- license creation tool (*continued*)
  - example, custom-configuration licenses
  - examples 174
  - format 167
  - license for production passwords 17
  - options 168
  - overview 17
  - usage 164
- license enablement
  - models 15
  - nodelocked licenses 7
  - purpose 1
- license password
  - checking enrollment certificate for version 165
  - compound 5
  - concurrent vendor-managed 41
  - overview 2
  - production 17
  - purpose 2
  - simple 5
  - test 17
  - types 5
- license policy
  - vendor-controlled
    - custom configuration 13
    - multiuse rules 12
    - try-and-buy 12
- License Requests by Product report, example 100
- license server, network
  - automatic start 71
  - components required 48
  - configuration 79
  - configuration options 70
  - configuring with i4cfg command 153
  - definition 216
  - direct binding on 79
  - enabling remote administration, direct binding 37
  - in a cluster 20
  - log file 183
  - process 160
  - purpose 3
  - selecting machines 35
  - selecting with Basic License Tool 93
- license types
  - concurrent 10
  - concurrent nodelocked 8
  - configuration required 68
  - in enrollment certificate file 185
  - network 9
  - nodelocked 7

- license types (*continued*)
  - per-seat 11
  - per-server 9
  - reservable 10
  - simple nodelocked 8
  - use-once 11
  - use-once nodelocked 8
- License Use Management
  - basic concepts 2
  - features and functions added in Version 4 211
- License Use Management Web site xvii
- License Use Runtime
  - introduction 1
  - overview 1
- LicenseEndDate tag, enrollment certificate file 185
- LicenseStartDate tag, enrollment certificate file 185
- llm1gnn file
  - configuration 74
  - managing 183
- log data, collecting
  - running enabled applications in traced mode 196
  - running processes in traced mode 195
  - running tools in traced mode 196
- log files
  - configuration on network license server 80
  - configuration on nodelocked license server 74
  - configuration options 70
  - configuring with i4cfg command 154
  - managing 183
- log threshold events 149
- logdbnn file
  - configuration 80
  - managing 183
- logging, WebSphere 197

## M

- manual backup 183
- manual recovery 183
- manuals, online xvi
- MaxLogFileSize parameter of configuration file 183
- memory requirements 49
- monitor threshold events 149
- monitoring the number of product users 100
- multiuse rules
  - defining 174
  - definition 216
  - purpose 12

## N

- NCS (network computing system)
    - definition 216
    - overview 36
  - NetBEUI 50
  - NetBIOS
    - configuration 50
    - default values 78
    - definition 216
    - requirements 49
    - troubleshooting 193
  - NetBIOS socket server
    - process 164
    - running 45
  - network computing system (NCS)
    - definition 216
    - overview 36
  - network connections, troubleshooting 192
  - network license
    - concurrent 10
    - configuring for 67
    - definition 216
    - overview 3
    - per-seat 11
    - purpose 3
    - reservable 10
    - supported for Java applications and applets 3
    - troubleshooting 188
    - types 9
    - use-once 11
  - network license client
    - components required 48
    - configuration 84
    - configuration options 70
    - configuring with i4cfg command 153
    - direct binding on 85
  - network license server
    - automatic start 71
    - components required 48
    - configuration 79
    - configuration options 70
    - configuring with i4cfg command 153
    - definition 216
    - direct binding on 79
    - enabling remote administration, direct binding 37
    - in a cluster 20
    - log file 183
    - process 160
    - purpose 3
  - network license server (*continued*)
    - remote administration, overview 18
    - selecting machines 35
    - selecting with Basic License Tool 93
    - workload balancing 36
  - network planning 35
  - new features and functions in Version 4 211
  - nodelock file
    - prepare manually 209
  - nodelocked license
    - concurrent 8
    - configuring for 67
    - definition 216
    - not supported for Java applications and applets 3
    - operation 22
    - overview 3
    - per-server 9
    - purpose 3
    - simple 8
    - troubleshooting 187
    - types 7
    - use-once 8
  - nodelocked license server
    - automatic start 71
    - components required 48
    - configuration options 70
    - configuration, in a network 75
    - configuration, standalone 73
    - configuring with i4cfg command 153
    - definition 216
    - direct binding example 38
    - direct binding on 77
    - log file 183
    - process 161
    - purpose 7
  - non-runtime-based enabling
    - definition 216
    - operation 22
    - purpose 7
    - troubleshooting 187
  - noninteractive installation 58
  - notational conventions xviii
  - NumberOfLogFile parameter of configuration file 183
- ## O
- online books xvi
  - OS/2
    - releases supported 5

## P

- password
  - for a cluster 41
- password use control level
  - definition 217
  - in enrollment certificate file 185
  - overview 6
- password, license
  - checking enrollment certificate for version 165
  - compound 5
  - concurrent vendor-managed 41
  - creation command 164
  - creation tool 17
  - definition 217
  - production 17
  - purpose 2
  - simple 5
  - test 17
  - types 5
  - use control levels 6
- PasswordVersion tag, enrollment certificate file 185
- PDF file xvii
- per-seat license
  - configuration required 69
  - deciding between per-server and per-seat 14
  - definition 217
  - Java applications and applets 33
  - operation 32
  - purpose 11
  - switching from per-server 106
  - troubleshooting 189
- per-server license
  - configuration required 69
  - deciding between per-server and per-seat 14
  - definition 217
  - operation 31
  - purpose 9
  - switching to per-seat 106
  - troubleshooting 189
- performance
  - direct binding 39
  - troubleshooting 190
- platforms, License Use Runtime 4
- policies, license
  - hard stop/soft stop 13
  - license annotation 13
  - multiuse rules 12
  - per-server/per-seat 14
  - product wait queues 12

- policies, license (*continued*)
  - try-and-buy 12
  - user access restriction 14
- port numbers
  - central registry license server 37
  - changing with i4cfg command 155
  - changing with the GUI 78
  - network license server 37
  - nodelocked license server 37
- PostScript file xvii
- processes
  - Central Registry 163
  - listing 90
  - network license server 160
  - nodelocked license server 161
  - starting 90
  - troubleshooting 190
- processor requirements 49
- ProductName tag, enrollment certificate file 185
- ProductVersion tag, enrollment certificate file 185
- protocols
  - configuring with i4cfg command 155
  - supported 49

## Q

- queues
  - definition 217
  - overview 12

## R

- RAM requirements 49
- README file, License Use Runtime xvii
- ReadTimeout parameter of configuration file 188
- recovery
  - after automatic backup 182
  - manual 183
  - of processes 191
- RegistrationLevel tag, enrollment certificate file 185
- Registry, editing 50
- remote administration
  - configuring direct binding for 77
  - enabling through direct binding 37
  - overview 18
- remote administration, disabling 71
- remote procedure call
  - purpose 36
- reports
  - definition 217



- reports (*continued*)
  - log files 183
  - requesting 99
  - types 18
- requirements
  - hardware 49
  - software 49
- reservable license
  - central registry required 21
  - configuration required 69
  - definition 217
  - Java applications and applets 29
  - managing 102
  - operation 28
  - purpose 10
  - troubleshooting 189
  - unreserved 104
- reserve licenses (i4blt -R) 137
- resetting the high-water mark 111
- response file
  - creating 60
  - noninteractive installation 58
  - purpose 58
- restart of processes 191
- Runtime component
  - disk requirements 49
  - required, by machine role 48
- runtime-based enabling
  - configuration required 69
  - definition 217
  - operation 23
  - purpose 7
  - troubleshooting 187

## S

- scenarios
  - configuring a central registry license server 88
  - configuring a network license client 84
  - configuring a network license server 79
  - configuring a nodelocked license server in a network 75
  - configuring a standalone nodelocked license server 73
  - creating and administering a cluster 114
  - high-availability licensing 114
  - managing a licensed product 94
  - managing reserved licenses 102
  - restricting user access 113
  - switching from per-server to per-seat 106

- scenarios (*continued*)
  - using hard stop/soft stop 109
- server, network license
  - automatic start 71
  - components required 48
  - configuration 79
  - configuration options 70
  - configuring with i4cfg command 153
  - direct binding on 79
  - enabling remote administration, direct binding 37
  - in a cluster 20
  - log file 183
  - process 160
  - purpose 3
  - selecting 35
- server, setting up 67
- server, troubleshooting heavy workload 190
- Service Manager Tool 91
- setup command
  - code server installation 62
  - command 59
  - interactive installation 51
  - noninteractive installation 58
- signature stamp 19
- silent installation 59
- Silicon Graphics support 5
- simple nodelocked license
  - configuration required 69
  - definition 217
  - purpose 8
- simple password
  - definition 218
  - purpose 5
  - use for vendor-managed use products 6
- socket server, NetBIOS
  - process 164
  - running 45
- soft stop policy
  - definition 218
  - in enrollment certificate file 185
  - in use when licenses are available 188
  - purpose 13
  - scenario 109
  - troubleshooting 189
- SoftStop tag, enrollment certificate file 185
- software
  - LUM Java Client Support 63
  - requirements 49
- Solaris support 5

- stale licenses
  - i4blt command for cleanup 138
  - troubleshooting 191
- Start up page, configuration notebook 74
- starting processes 90
- startup of processes, automatic
  - configuring with GUI 74
  - configuring with i4cfg command 155
  - planning 71
- summary of changes xix
- Sun Solaris support 5
- switching from per-server to per-seat 14

**T**

- target ID
  - in enrollment certificate file 185
  - troubleshooting 190
- target view tool 158
- TargetType tag, enrollment certificate file 185
- TCP/IP
  - definition 218
  - required for LUM Java Client Support 64
  - requirements 49
  - troubleshooting 193
- test verification tool 157
- threshold
  - definition 218
  - purpose 20
  - setting 134
- time stamp
  - definition 218
- time, synchronizing 188
- tools
  - target view 158
  - test verification 157
- TraceActivities parameter of configuration file 195
- traced mode 196
- troubleshooting
  - collecting error log data 195
  - collecting other data 196
  - custom configuration 192
  - custom configuration licenses 192
  - customer-managed use products 189
  - deleting keys 184
  - deleting products or reducing numbers 184
  - direct binding 192
  - hardware 194
  - heavy server workload 190
  - interprocess communication failure 188

- troubleshooting (*continued*)
  - NetBIOS 193
  - network connections 192
  - network licenses 188
  - nodelocked licenses 187
  - per-server/per-seat licenses 189
  - performance 190
  - processes 190
  - requesting a license upgrade 184
  - reservable licenses 189
  - restart and recovery of processes 191
  - running enabled applications in traced mode 196
  - running processes in traced mode 195
  - running tools in traced mode 196
  - servlet support 197
  - TCP/IP 193
  - uninstalling Java Client Support 198
  - vendor-managed use products 190
- try-and-buy policy
  - definition 218
  - purpose 12
- types, license
  - concurrent 10
  - concurrent nodelocked 8
  - configuration required 68
  - in enrollment certificate file 185
  - network 9
  - nodelocked 7
  - per-seat 11
  - per-server 9
  - reservable 10
  - simple nodelocked 8
  - use-once 11
  - use-once nodelocked 8
- typographic conventions xviii

**U**

- UDP (universal datagram protocol) 194
- uninstallation
  - License Use Runtime 63
  - LUM Java Client Support 64
- universal datagram protocol (UDP) 194
- update enrolled licenses (i4blt -U) 132
- use control levels
  - definition 217
  - in enrollment certificate file 185
  - overview 6
- use-once license
  - configuration required 69

- use-once license (*continued*)
  - definition 218
  - Java applications and applets 24
  - operation 24
  - purpose 11
  - troubleshooting 188
- use-once nodelocked license
  - configuration required 69
  - definition 218
  - purpose 8
- user access restriction
  - purpose 14
  - scenario 113
- user file
  - definition 218
  - purpose 14
  - scenario 113
- users, monitoring 100

- Windows (*continued*)
  - software requirements 49
  - workload, troubleshooting 190

## V

- vendor-managed use control
  - concurrent license password 41
  - definition 218
  - overview 6
  - RegistrationLevel tag, enrollment certificate file 185
  - troubleshooting 190
- VendorName tag, enrollment certificate file 185
- verifying network connections 46
- version installed, determining 48

## W

- wait queues
  - definition 217
  - overview 12
- Web server
  - required for Java applications and applets 3
  - supported 63
- Web site, LUM
  - downloading License Use Runtime code 4
  - LUM Java Client Support 64
  - purpose and URL xvii
- WebSphere
  - logging 197
  - required 64
- what's new xix
- Windows
  - protocol configuration 50
  - releases supported 5

# IBM

Program Number: 5697-B23  
5697-E77

Printed in Denmark by IBM Danmark A/S

SH19-4347- 2

