Tru64 UNIX

Software License Management

Part Number: AA-RH9EB-TE

July 1999

Product Version: Tru64 UNIX Version 5.0 or higher

This guide describes how to use the License Management Facility (LMF) to manage software licenses on an HP Tru64 UNIX system (formerly DIGITAL UNIX).

Hewlett-Packard Company
Palo Alto, California

None of Compaq, HP, or any of their subsidiaries shall be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for HP or Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Contents

About This Manual

Note: LMF and Legal License Agreements

1 License Management Facility

2 License Management Utility, lmf

A LMF Messages

Glossary

Index

Examples

About This Manual

This guide describes how to use the License Management Facility (LMF) to manage software licenses on an HP Tru64 UNIX system (formerly DIGITAL UNIX) operating system. System managers can use LMF to help them make sure that licenses are used as intended.

Audience

This guide is for system managers responsible for managing software licenses on Tru64 UNIX systems. The guide also provides information for anyone who uses licensed software on Tru64 UNIX systems.

LMF is designed to be used by companies other than HP. Other companies can issue Product Authorization Keys (PAKs) or have HP issue them on their behalf. Additionally, other companies can include license-checking functions in their software. For clarity, however, this manual addresses only software that is supplied and produced by HP.

Organization

This guide is divided into two chapters, one appendix, and a glossary as follows:

Chapter 1 Introduces the tools and components associated with managing licenses. The chapter explains the types of licenses available and how products are affected by LMF. The chapter also describes Product Authorization Keys (PAKs) and the license checking that LMF and the products that support it perform.

Chapter 2 Describes the tasks you can perform using the `lmf` utility.

Appendix A Lists and explains the messages that you might encounter while you are using LMF.

Glossary A glossary of important LMF terms.

Icons on Tru64 UNIX Printed Manuals

The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the manuals to help specific audiences quickly find the manuals that meet their needs. (You can order the printed documentation from HP.) The following list describes this convention:

G	Manuals for general users
S	Manuals for system and network administrators
P	Manuals for programmers
R	Manuals for reference page users

Some manuals in the documentation help meet the needs of several audiences. For example, the information in some system manuals is also used by programmers. Keep this in mind when searching for information on specific topics.

The *Documentation Overview* provides information on all of the manuals in the Tru64 UNIX documentation set.

Reader's Comments

HP welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

The following conventions are used in this manual:

`%`
`$`

A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells.

`#`

A number sign represents the superuser prompt.

`%` **`cat`**

Boldface type in interactive examples indicates typed user input.

`file`

Italic (slanted) type indicates variable values, placeholders, and function argument names.

[|]
{ | }

In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.

. . .

In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.

`cat`(1)

A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat`(1) indicates that you can find information on the `cat` command in Section 1 of the reference pages.

Note: LMF and Legal License Agreements

The terms and conditions of your license agreement determine your legal use of the software.

The License Management Facility (LMF) is a system management tool that can help you comply with your license agreement, but use of LMF does not indemnify you against noncompliance with the terms and conditions of your software license agreement. LMF offers options for many kinds of license agreements, but using some of these options might not be authorized by your specific license agreement.

You must read the terms and conditions of your license carefully to determine which LMF options you can legally use.

This document describes some LMF features that are not currently in use. In the future, some of the features described herein, may be used, but no commitment is made beyond the current software business practices.

1

License Management Facility

When every computer system was a single processor, and software products were dedicated to that processor, managing software licenses was relatively simple and straightforward. In a distributed computing environment, system managers face more complexity.

Distributed computing allows a much wider variation in software use than a single processor. Software might be used systemwide, or by just a few users. Who uses software and where it can be used might change depending upon the computer environment.

Because managing software licenses is becoming more complex, the License Management Facility (LMF) is employed. LMF is a set of software tools that help you manage software licenses. If a software product has LMF support, LMF also verifies that the use of the product complies with its license.

This chapter introduces LMF by answering the following questions:

- What is LMF?
- What is your responsibility in license management?
- What software is affected by LMF?
- What types of software licenses are available?
- What are the contents of a Product Authorization Key (PAK)?
- How does LMF verify product use?

1.1 What Is LMF?

The License Management Facility (LMF) is part of the Tru64 UNIX operating system and consists of the following:

- A License Product Authorization Key (PAK)

  A PAK is a set of license information, such as the name and version number of the product, the product release date, the date after which the license expires (if any), and the amount of use allowed by the license. The PAK is your proof of license, and it should be stored in your files for future reference. For more information about PAKs, see Section 1.5.

- The license database

  The license database stores information about all the licenses registered on your system. LMF creates the license database the first time you register a license.

- The license management utility, `lmf`

  The `lmf` utility has commands that allow you to register, load, and maintain licenses on your system. For information about using the `lmf` utility, see Chapter 2, or the `lmf`(8) reference page.

- The license management script, `lmfsetup`

  The `lmfsetup` script includes an alternative to the `lmf` utility for registering licenses. For more information about `lmfsetup`, see the `lmfsetup`(8) reference page.

- License-checking functions

  License-checking functions are included in each product that has LMF support. They verify that potential users of a product have a valid license to use it.

- License Unit Requirement Tables

  License Unit Requirement Tables (LURTs) specify how many license units you need to run a product on a particular model of hardware. Different models of hardware require a different number of license units to allow a product to run. In general, the more powerful your hardware is, the more license units you need to run a product.

  The license unit is a measure of software use.

  The License Unit Requirement Tables are internal to LMF. You cannot display or modify them.

When you buy a software product you are issued a license to use the product. The license is described by the PAK you receive. You must enter the license information on the PAK into the license database by using the `lmf` utility. This process is called *registering a license.*

Once you have registered a license, you use the `lmf` utility to load the license into the kernel cache. In doing so, you make the license information available to the LMF's license-checking functions.

Each time a user attempts to run a product that includes LMF support, the product calls the license-checking functions. The license-checking functions check the kernel cache to be sure that the license allows the user to use the product. The license-checking functions allow the user to run the product when the kernel cache contains a sufficient number of license units. For more information about how LMF checks licenses, see Section 1.6.

1.2 Your Responsibility in License Management

When you buy software, the vendor from which you purchase the software issues an agreement called a *license*. A software license can involve a rental agreement and other complex arrangements. Although the term license can have specific legal meanings, for the purposes of this manual, *license* refers to the authorization you have to use a product.

The License Management Facility (LMF) exists to help you with the task of license management. However, the responsibility remains with your organization for using the tools in a way that fulfills your record-keeping obligations, and for ensuring that your company honors all license terms.

1.3 Products Affected by LMF

Software products are in two general categories:

- Software that does not support LMF

  These software products do not use LMF to authorize software use. License information is issued separate from the software.

- Software that supports LMF

  License information is issued on a PAK that must be registered in the license database. These software products contain the LMF license-checking functions that verify your use of a particular software product.

Not all software products have LMF support. If the license you receive for a product is not in the form of a PAK, that product does not support LMF. If the license is a PAK, the product supports LMF, and you must register it in the license database. Refer to the product documentation for specific information about its support for LMF.

1.4 Types of Software Licenses

When you buy a software product, the software seller issues you a license to use that software. Two main types of licenses are issued for software products that have LMF support. Most products are licensed by either an *availability license* or an *activity license.*

The Availability Table Code field and the Activity Table Code field on your PAK determine whether you have an availability license or an activity license. If your license is an availability license, your PAK contains data in the Availability Table Code field and the Activity Table Code field is blank. If your license is an activity license, the PAK contains data in the Activity Table Code field and the Availability Table Code field is blank.

The sections that follow explain how availability licenses and activity licenses limit a user's access to software.

1.4.1 Availability Licenses

Availability licenses (also called capacity licenses) give users unlimited access to a product once its license has been registered and loaded on a particular processor. This type of license allows you to run a product only on certain, specified processors.

Suppose you have two standalone systems, SMALLSYS and BIGSYS. The SMALLSYS system requires a 400-unit license to run a particular product. BIGSYS is a more powerful processor, so it requires a 700-unit license. Once a product is registered and loaded on SMALLSYS or BIGSYS, any number of users can run the product.

Now suppose that you must shut down the BIGSYS system to upgrade its memory. When you shut down the system, you want to move the products it runs to the SMALLSYS system, so the products are still available to users.

You can move the products that run on the BIGSYS system to SMALLSYS. The licenses for the BIGSYS products must be at least 700-unit licenses. Each product on SMALLSYS requires only 400-unit licenses. Because all the products that run on the BIGSYS system have licenses that are larger than 400-unit licenses, you can run those products on the SMALLSYS system. (You use the `issue` command, described in Section 2.7, to move the license.)

On the other hand, suppose you must shut down the SMALLSYS system for a memory upgrade. You cannot necessarily move the products that run on the SMALLSYS system to the BIGSYS system. The licenses for SMALLSYS products must be at least 400-unit licenses. Each product on BIGSYS requires at least a 700-unit license. You cannot run any product with a 400-unit license on BIGSYS. (Some licenses running on the SMALLSYS system might be at least 700-unit licenses; these products you could move to the BIGSYS system.)

When a product has a license that has too few license units to run on a particular processor, you might be able to increase the number of license units. When the license PAK contains the MOD_UNITS Key Option, you can use the `lmf` utility's `modify` command to increase the number of license units, as described in Section 2.10.1. If the PAK does not have the MOD_UNITS Key Option, contact your support representative, who will probably recommend one of the following:

- A new license that has sufficient license units for the new processor
- Another license for the same product that has additional license units

If the terms of your license allow it, you can register the two licenses for the same product and combine them to form one larger license. For a complete explanation of license combination, see Section 2.11.

1.4.2 Activity Licenses

Activity licenses (also called per-user licenses) limit the number of users who can use a product simultaneously. An activity license might allow as few as one user to run a product or as many as several hundred users to run a product at one time. (In this context, *user* refers to a process, not necessarily a person.)

The number of users who can run a product at the same time is controlled by the number of units for the license. A 400-unit license allows fewer users to run a product than a product with a 700-unit license.

The number of users who can run a product also depends upon the processor on which the product runs. Each processor requires a certain number of license units for each user to run a process. For example, a processor might require 25 license units per user.

Suppose you have two standalone processors. The SMALLSYS requires 25 license units per user to authorize a product, and the BIGSYS requires 50 license units per user to authorize a product.

If you obtain a 1200-unit activity license for BIGSYS, you temporarily can move that license (using the `issue` command) to SMALLSYS when you shut down BIGSYS for maintenance. The 1200-unit license, which allowed 24 users to run the product on BIGSYS, allows 48 users to run the product on SMALLSYS.

You can move a 40-user, 1000-unit activity license originally intended for SMALLSYS to BIGSYS. However, on BIGSYS, the license allows access to only 20 users.

If you need more activity units for a license and your license PAK contains the MOD_UNITS Key Option, you can increase the number of units on the license by using the `lmf` utility’s `modify` command. If the PAK does not have the MOD_UNITS Key Option, contact your support representative, who will probably recommend one of the following:

- A new license with more units
- Another license for the same product that has additional license units

  If the terms of your license allow it, you can register two or more licenses for the same product and combine them to form one larger license. For a complete explanation of license combination, see Section 2.11.
- A different kind of license

Because some products offer both activity and availability licenses, the representative might suggest changing to an availability license.

1.5 License Product Authorization Keys

A License Product Authorization Key (PAK) is a unique set of data used by LMF to confirm that a product is licensed.

The PAK is issued to you when you purchase a software license. It is a valuable proof of purchase, represents your license to use a software product, and should be stored in your files. The license information in a PAK is confidential and should not be publicly posted or widely distributed. To comply with the license terms, you always must register a PAK in the license database.

1.5.1 Obtaining a PAK

You obtain both a PAK and the product from the representative who distributes software. You order a PAK just as you might order another product. Before you order a PAK, define your software and hardware requirements to your sales representative so that you get a license of the correct size. You usually will receive a PAK printed on a piece of paper when you buy the software.

1.5.2 Product Authorization Amendments

Although they are not used currently, Product Authorization Amendments (PAAMs) may be issued in the future. A PAAM is similar to a license PAK but includes only the data needed to identify, update, and further authorize product use. For example, you might receive a PAAM if you want to increase the number of units for a license that does not have the MOD_UNITS Key Option. The license for the product should be updated using the `amend` command, as described in Section 2.10.2.

Note

The current business policy is to not issue PAAMs.

1.5.3 Information on a License PAK

This section describes the information contained in the fields on the PAK. Example 1–1 shows a typical PAK. When you register the PAK in the license database, the `lmf` utility displays a Comments field that you can use as needed. For information about how to register a PAK, see Section 2.2.

Example 1–1: Sample License PAK Fields

```
*************************************************************************

                   ISSUER:         DEC[1]

     AUTHORIZATION NUMBER:         AWS-PK-88229-2[2]
             PRODUCT NAME:         ALLSUM[3]
                 PRODUCER:         DEC[4]
          NUMBER OF UNITS:         1200[5]
                  VERSION:        Version 2.0[6]
     PRODUCT RELEASE DATE:         1-JUL-1993[7]
     KEY TERMINATION DATE:[8]
  AVAILABILITY TABLE CODE:         M[9]
      ACTIVITY TABLE CODE:[10]
              KEY OPTIONS:[11]

            PRODUCT TOKEN:[12]
             HARDWARE I.D:[13]
                 CHECKSUM:         1-ODKM-NIIO-JEPJ-FCLB[14]

*************************************************************************
```

The following list describes the fields on the PAK shown in Example 1–1:

[1] The issuer is the LMF name for the entity that supplies the PAK. Most licenses specify DEC. However, PAK issuers might further be identified by region or department within the company. For example, the PAK issuer string could be DEC-USA or DEC-EUROPE. Other software vendors with products that have support for LMF might also issue PAKs.

[2] The authorization number, together with the issuer, uniquely identifies each license. You can use the number for reconciling your records. This number allows everyone to know, with certainty, which license is being referenced.

[3] The product name is the name used by LMF to distinguish among different software products. The product name that appears on the PAK might be slightly different from the name in the Software Product Description (SPD). This difference is due to field length restrictions imposed by LMF.

[4] The producer is the name of the company producing the software. For all software produced by HP, the producer name is DEC. The Producer field is used by LMF to distinguish between products with the same name that are produced by different companies. For example, you might have two FORTRAN compilers, one produced by HP and one produced by another company.

[5] The Number of Units field shows how many license units are supplied with the PAK. If this field contains a zero (0), the license has unlimited size. The license allows unlimited use of the product on any type of processor.

6 The Version field restricts the use of the PAK to particular versions of a product. This field is not used on all PAKs. For example, if the version number on the PAK is 2.0, the PAK can be used with all versions of the product up to and including Version 2.0.

PAKs are not issued that specify both a version number and a product release date. PAKs might be issued that are not restricted by version number or product release date.

7 The Product Release Date field restricts the use of the PAK to versions of the product released before a certain date. This field is not used on all PAKs.

PAKs are not issued that specify both a version number and a product release date. PAKs might be issued that are not restricted by version number or product release date.

8 The key termination date controls when the PAK expires. After this date, the PAK no longer represents a valid license for the product.

9 The availability table code represents the number of units required to give unlimited use of a product on a particular hardware system model. This field contains either a letter or `CONSTANT=`*`integer`*.

A letter represents the License Unit Requirement Table (LURT). The LURT defines the number of units required for the product to run on each hardware system model. The LURT is internal to LMF; you cannot display or modify it.

If the field contains `CONSTANT=`*`integer`*, the *integer* defines how many units the product requires to run, regardless of which hardware system model you have. If your PAK has an availability table code with, for example, `CONSTANT=100`, it means that the product needs 100 units to run on any hardware system model, regardless of size.

10 The activity table code represents the number of units required for each simultaneous user of the product. This field contains either a letter or `CONSTANT=`*`integer`*.

A letter represents the LURT. The LURT defines the number of units required for each simultaneous user to run the product on a particular hardware system model. The LURT is internal to LMF; you cannot display or modify it.

If the field contains `CONSTANT=`*`integer`*, the *integer* defines how many units the product requires for each user, regardless of the hardware system model you have. If your PAK has an activity table code with, for example, `CONSTANT=25`, it means that each simultaneous user of the product needs 25 units to run the product on any type of processor, regardless of size.

11 The Key Options field might contain one of the following options:

- ALPHA

  The ALPHA Key Option indicates that this license is valid only on the Alpha AXP family of systems.

- MOD_UNITS

  The MOD_UNITS Key Option indicates that you are allowed to modify the Number of Units field (using the `modify` command). For a complete description of how to modify the Number of Units field, see Section 2.10.1.

- NO_SHARE

  The NO_SHARE Key Option indicates that you cannot combine two or more licenses for the product on the same processor.

- P_FAMILY

  Some licenses might have the P_FAMILY Key Option in this field. However, LMF allocates license units to each executing process for the product, even if several processes belong to the same user.

Note

The P_FAMILY Key Option is not valid for DEC OSF/1 licenses.

12 The Product Token field is not currently used by LMF. However, you must enter any data that is included in this field when you register the PAK. Otherwise, you receive an error message.

13 The Hardware I.D. field is not currently used by LMF. However, you must enter any data that is included in this field when you register the PAK. Otherwise, you receive an error message.

14 The Checksum field shows a checksum that has been generated from the individual data elements on the PAK. The checksum is unique for each PAK. After you have entered the PAK data into the license database, the checksum is calculated to determine that you have entered the information correctly.

1.6 License Checking Performed by LMF

LMF performs license checking for availability licenses and activity licenses. The type of checking performed depends upon the type of license you have.

If you have an availability license, LMF checks the license when it loads the license into the kernel cache from the license database. The license-checking functions check the license each time a user attempts to run the product.

If you have an activity license, LMF does not check the license when it loads the product into the kernel cache. However, the license-checking functions verify that each user is allowed to use the product when the user attempts to run the product. The license-checking functions might also check the license at other times while the user is using the product. For example, if the user creates a subprocess, the license-checking functions verify that enough license units are available for the subprocess to run the product.

This section explains the steps LMF performs to check licenses for availability-licensed products and activity-licensed products.

1.6.1 Checking Availability-Licensed Products

A valid availability license makes a product available to all the users on a system. LMF allows a user to run a product if the number of units on the license matches or exceeds the license unit requirement for the current processor. Availability licenses are checked as follows:

- By LMF when the license details are loaded into the kernel cache from the license database
- By the license-checking functions in the product when a user attempts to run the licensed software

These two activities are described in the following paragraphs.

1.6.1.1 Loading a License into the Kernel Cache

When LMF loads an availability license from the license database into the kernel cache, it verifies that the license supplies enough license units to run on the current hardware system model. LMF uses the Availability Table Code field on the PAK to determine how many license units a product needs to run on the current hardware system model. Depending upon the contents of the field, LMF performs the following license checking:

- The field contains `CONSTANT=0`.

  LMF performs no further checking. This license allows the product to run on any processor. LMF loads the license into the kernel cache.

- The field contains `CONSTANT=`*`integer`*.

  LMF defines the number of license units required to be *`integer`*. To run on any processor, the license must have at least *`integer`* license units.

  LMF compares the value specified in the Number of Units field on the PAK to *`integer`*. If the value is equal to or greater than *`integer`*, LMF loads the license into the kernel cache. If the value is less than *`integer`*, LMF issues an error message and does not load the license.

- The fields contains a letter, such as M.

The letter corresponds to a LURT that LMF uses internally. Each LURT contains a list of hardware system models and their corresponding license unit requirements. For example, a LURT might specify that one hardware model requires 1000 license units, while another, more powerful system requires 1500 license units.

To check the license, LMF determines the hardware system model of the current processor. It then looks at the *M* LURT to determine the requirements of the hardware system model. Once it has determined the license unit requirement for the current hardware system, LMF compares the requirement to the value in the Number of Units field on the PAK. If the value is equal to or greater than the license unit requirement, LMF loads the license into the kernel cache. If the value is less than the license unit requirement, LMF issues an error message and does not load the license.

Suppose you buy a fictional product named ALLSUM. The PAK you receive specifies the letter M in the Availability Table Code field. The PAK specifies 1200 license units in the Number of Units field.

Suppose you register the 1200-unit license and attempt to load it on a system named SMALLSYS. When you attempt to load the license, LMF checks the Availability Table Code field. When it finds the letter M, LMF determines the model of SMALLSYS's hardware. LMF looks in the LURT M for SMALLSYS's model. Suppose the table indicates that SMALLSYS requires 1000 license units to run a product. Because your license has 1200 license units, LMF loads the product into the kernel cache.

However, suppose you register and attempt to load the same license on a system named BIGSYS. Suppose this processor requires 1500 license units to load a product. In this case, LMF does not load the product (because the license is 300 license units short of the requirement).

1.6.1.2 Running the Licensed Software

Each time a user attempts to run an availability-licensed product, the license-checking functions in the product check the kernel cache for a valid license for the product. The functions perform the following license checks when a user attempts to run a product:

- The software has the same product name and producer name as those on the license.
- The version number of the software is not greater than the version number (if specified) on the license.
- The product release date of the software is not later than the product release date (if specified) on the license.

- The current date is not later than the key termination date (if specified) on the license.
- The current date is not later than the cancellation date (if specified) on the license.

The license-checking functions do not check the number of license units for the product because any number of users can run availability-licensed products. LMF checks that the PAK has enough license units to run the product when it loads the license for the availability-licensed product into the kernel cache.

If a user attempts to run an availability-licensed product that does not have a valid license in the kernel cache, the product displays an error message and does not allow the user to run it. For details about how each product responds to an invalid license, see the documentation for the product.

1.6.2 Checking Activity-Licensed Products

An activity license defines the number of license units each user of a product needs to run the product. Activity licenses can have access to only a few simultaneous product users or to hundreds of simultaneous product users. The number of users depends upon how many license units the license has and how many license units each user needs to run the product on a particular processor.

The number of license units each user needs is defined when LMF loads the license into the kernel cache. The license-checking functions check the license each time a user runs the licensed product.

1.6.2.1 Loading a License into the Kernel Cache

When LMF loads an activity license from the license database into the kernel cache, it performs no license checking. LMF loads the license details and the per-user requirement for the license into the kernel cache.

LMF determines the per-user requirement by checking the Activity Table Code field on the PAK. Depending upon the contents of the field, LMF defines the per-user requirement as follows:

- The field contains `CONSTANT=0`.

  LMF defines the per-user license requirement to be zero (`0`). Any number of users can run this product simultaneously. LMF loads that information into the kernel cache.

- The field contains `CONSTANT=`*`integer`*.

  LMF defines the per-user license requirement to be *`integer`* license units. Each user that runs the product requires *`integer`* license units. LMF loads that information into the kernel cache.

- The fields contains a letter, such as K.

  The letter corresponds to a LURT that LMF uses internally. Each LURT contains a list of hardware system models and their corresponding per-user license unit requirements. For example, a LURT might specify that one hardware model requires 25 license units per user, while another, more powerful system requires 50 license units per user.

  To define the per-user license unit requirement, LMF determines the hardware system model of the current processor. It then checks the LURT to determine how many per-user license units the processor requires. LMF loads that information into the kernel cache.

1.6.2.2 Running the Licensed Software

Each time a user attempts to run an activity-licensed product, the license-checking functions in the product check the kernel cache for a valid license for the product. (In this context, “user” refers to a process, not necessarily a person.) These functions perform the following checks when a user attempts to run a product:

- The software has the same product name and producer name as those on the license.
- The version number of the software is not greater than the version number (if specified) on the license.
- The product release date of the software is not later than the product release date (if specified) on the license.
- The current date is not later than the key termination date (if specified) on the license.
- The current date is not later than the cancellation date (if specified) on the license.
- Enough license units are available to meet the per-user license requirement for the product on the current processor.

  If the product has a valid license in the kernel cache, the license-checking function in the product compares the number of units required for each user to the number of units available. If the number of units available matches or exceeds the license unit requirement for the current processor, the user can run the product.

If a user attempts to run an activity-licensed product that does not have a valid license in the kernel cache, the product usually displays an error message and does not allow the user to run it. For details about how each product responds to an invalid license, see the documentation for the product.

Suppose you have registered and loaded a license for a fictional product named ALLSUM. Suppose that on the current processor, the per-user license unit requirement is 25. Each time a user invokes a product, 25 license units must be available on the system. Suppose also that your license for a product has 1200 license units.

In this case, when the first user invokes the product, the license-checking function allocates 25 of the 1200 available license units to that user. As long as the first user is using the product, those 25 units remain allocated, leaving 1175 available in the kernel cache for other users.

When the next user attempts to use the product, the checking function repeats the authorization procedure again. Thus, when the second user invokes ALLSUM, the checking function looks for 25 available license units to authorize product use. Because the ALLSUM license now has 1175 license units unallocated in the kernel cache, the license-checking function again authorizes product use.

In this example, the first 48 concurrent users can run the product, but additional users are denied access.

As each user finishes using the product, the license units allocated to that user become available again.

2

License Management Utility, lmf

The License Management Facility (LMF) provides the lmf utility to help manage the software licenses for your system. You use the lmf utility to maintain a file of registered software licenses, the license database. In addition, you can also use the lmf utility to maintain the kernel cache. This chapter describes how to manage licenses by using the lmf utility to perform the following tasks:

- Register and load a Tru64 UNIX license
- Register a license
- Load a license
- Unload a license
- Disable a license
- Enable a license
- Issue a license
- Cancel a license
- Delete a license
- Update a license
- Combine licenses
- Display information about licensed products
- Review your license management activities

The LMF commands can be used only by a person logged in to the system as superuser (root login). You can allow nonprivileged users to use the list and history commands, but you must change the file mode permissions on the files that the commands access. The list command accesses /usr/var/adm/lmf/ldb, and the history command accesses /usr/var/adm/lmf/ldb_history.

When you use LMF commands, you can enter them on a single line, for example:

```
# lmf register
```

Alternatively, you can run the lmf utility and enter the commands after the lmf> prompt, for example:

```
# lmf
lmf> register
```

For more information about LMF commands, see the `lmf`(8) reference page.

You can copy or move the license database from the `/usr/var/adm/lmf` directory to another directory on your system. To access the copied or moved license database, specify the`-d`*`dir`* option on the `lmf` command line. When you use this option, LMF looks in the directory you specify for the license database. LMF modifies that license database as you use commands like `register`, `load`, `unload`, `modify`, and so on. LMF also creates the history file in that directory.

2.1 Registering and Loading a Tru64 UNIX License

When you install or upgrade to Tru64 UNIX an inherent user activity license for the operating system automatically is registered and loaded for you. The license is named the OSF-USR license, and contains the following two data elements:

```
Product Name: OSF-USR
Authorization: UNIX-SERVER-IMPLICIT-USER
```

If you obtain an unlimited user activity license for the operating system, you must delete the inherent user activity license before you can register and load the unlimited license. If you do not remove the inherent license before registering and loading the unlimited license, the command will fail while displaying the error `Multiple Licenses could not be combined for OSF-USR DEC`.

To prevent this problem, delete the inherent license by using the `lmf` utility:

```
# lmf
lmf> delete OSF-USR AUTHORIZATION UNIX-SERVER-IMPLICIT-USER
```

Then, use `lmf` to register and load the unlimited license.

2.2 Registering a License

When you install a product that provides LMF support, you must register the product. Registering the license records the license details listed on the Product Authorization Key (PAK) in the license database. Once the details are in the license database, LMF can load them into the kernel cache.

To register a license, you use the `register` command. The `register` command displays a template that contains the fields that appear on a PAK. By default, the fields on the template are blank. You register the license by entering the data from your PAK into the blank fields.

Some products provide a partially completed template that you can use to register a license. If you use a partially completed template, you need to complete only the missing information.

In some cases, you might need to register a completed template that has, for example, been sent to you through electronic mail.

The following sections describe the process of registering licenses in the license database and explain what to do if something goes wrong.

2.2.1 Editing a Blank Template

To register a license by editing a blank template, use the `register` command without arguments. When you issue the command without arguments, LMF displays a template that includes all the fields on a PAK and an additional field for your comments.

Suppose you need to register the information from the License PAK shown in Example 1–1. Enter the following command at the superuser prompt:

```
# lmf register
```

LMF displays a blank template and runs an editor to allow you to edit the template. LMF runs the editor that is defined by your `EDITOR` environment variable. If the environment variable is undefined, LMF runs the `vi` editor.

The following example shows the template that LMF displays for the ALLSUM product and, in bold type, the information you would supply from the ALLSUM PAK:

```
                 Licensed Software Product
                 Product Authorization Key

         Enter data on lines terminated with :

                      Issuer: dec
        Authorization Number: 3aws-pk-88229-2

                Product Name: allsum
                    Producer: dec

             Number of units: 1200

                     Version: Version 2.0
        Product Release Date: 31-jul-1999

           Key Termination Date: "

        Availability Table Code: M
            Activity Table Code:
                    Key Options:
                  Product Token:
                    Hardware Id:
                       Checksum: 1-odkm-niio-jepj-fclb

Comment: This is an example license.
```

If your PAK contains blank fields, leave the same fields blank on the template. You can enter the data in uppercase or lowercase characters.

You must enter the license information from the PAK carefully. LMF returns the following error message if you omit or incorrectly enter any license data:

```
Checksum does not validate
```

If you receive this message, carefully check the characters typed on each line, not just the checksum string.

When you finish entering the license data, exit from the editor.

LMF then scans the completed template to make sure all the license data has been entered correctly. If the license data is correct, LMF copies it into the license database. If the license data is incorrect, you can reenter the editor and correct mistakes.

2.2.2 Editing a Partially Complete Template

In some instances, you might have license data in files on your system as a result of using the `issue` command (see Section 2.7), or such data might have been copied to your system as part of a product installation. For example, the installation procedure for the fictional product ALLSUM might copy license data to the file `/usr/var/adm/lmf/ALLSUM`. This file would contain license data common to all ALLSUM licenses. To register the license, you add your specific license details from your ALLSUM PAK.

To register a license by using a partially complete template, use a command like the following one:

```
# lmf register /usr/var/adm/lmf/ALLSUM
```

In response to this command, LMF runs an editor to allow you to edit the existing license data. LMF runs the editor defined by the `EDITOR` environment variable. If the environment variable is undefined, LMF runs the `vi` editor.

The following example shows the template that is displayed by LMF for the ALLSUM product and, in bold type, the information you would supply from the ALLSUM PAK:

```
                 Licensed Software Product
                 Product Authorization Key
```

```
         Enter data on lines terminated with :

                         Issuer: DEC
           Authorization Number: aws-pk-88229-2s

                   Product Name: ALLSUM
                       Producer: DEC

                Number of units: 1200

                        Version: Version 2.0
           Product Release Date: 1-JUL-1999

           Key Termination Date:

        Availability Table Code: M
            Activity Table Code:

                    Key Options:
                  Product Token:
                    Hardware Id:
                       Checksum: 1-odkm-niio-jepj-fclb

Comment: This is an example license.
```

If your PAK contains blank fields, leave the same fields blank on the template. You can enter the data in uppercase or lowercase characters.

You must enter the license information from the PAK carefully. LMF returns the following error message if you omit or incorrectly enter any license data:

```
Checksum does not validate
```

If you receive this message, carefully check the characters typed on each line, not just the checksum string.

Once you finish entering data into the fields on the template, exit from the editor.

LMF then scans the completed template to make sure all the license data has been entered correctly. If the license data is correct, LMF copies it into the license database. If the license data is incorrect, you can reenter the editor and correct mistakes.

In some cases, you might have a license data file that was not created by the `issue` command or a product installation. In this case, verify that the file contains all the license fields that have entries on your PAK. The license fields names must be in the same format as the template displayed with the `register` command, that is, the same combination of uppercase and lowercase characters, with a colon (`:`) separating the field name and the data. The license fields can be in any order.

Files created using the issue command or as part of product installations automatically have the field names in the correct format.

2.2.3 Registering a PAK Directly

In many cases, the license data file that you create with the issue command is complete. When you have a complete license data file, you can use the register – command to register the data in the file by redirecting the file to standard input.

The following example shows how to use the command:

```
# lmf register - < /usr/var/adm/lmf/ALLSUM
```

LMF does not display the contents of the file or allow you to edit it. However, it does scan the file to be sure that the format and data are correct. If the license data is correct, LMF copies it into the license database. If the license data is incorrect, LMF does not copy it into the license database, and it displays the appropriate error message. The command also returns an exit status: zero, if the license data has been copied into the license database; or nonzero, if the license data was not copied into the license database.

2.3 Loading a License

Once you have registered a license in the license database, load it into the kernel cache to make the license details available to the license-checking functions. The license-checking functions only allow a product to run if it has a valid license in the kernel cache.

To load license details into the kernel cache, use one of the following methods:

- Use the load command.

  This command copies the license details for a particular product from the license database to the kernel cache.

- Use the reset command.

  This command copies the license details for all products from the license database to the kernel cache.

- Start the system.

  The system startup process automatically executes the reset command.

2.3.1 Loading a License for One Product

Use the load command to copy the license details for a specified product from the license database to the kernel cache. As an argument to the load command, specify the number of users that you want to be able to use the product.

If a product is availability licensed, specify the number of users to be zero (0) when you use the load command. This practice makes sure the number of units loaded into the kernel cache is always enough to satisfy the requirements of the processor. For example, assume the product ALLSUM is availability licensed, and you have registered the license in the license database. To load the license into the kernel cache, enter the following command:

```
# lmf load 0 ALLSUM
```

Alternatively, assume the product ALLSUM is activity licensed, and you have registered enough license units for 10 product users in the license database. To load all the license units for the product into the kernel cache, enter the following command:

```
# lmf load 0 ALLSUM
```

If you want to load only enough license units for five users, enter the following command:

```
# lmf load 5 ALLSUM
```

When you use the load command, make sure you supply enough arguments to uniquely identify the license. If you have two products with the same name from different producers, you must supply the producer name as well as the product name, for example:

```
# lmf load 0 ALLSUM DEC
```

If your system has two or more licenses with the same product and producer name, the load command loads all the licenses into the kernel cache only if the licenses can be combined. For a complete explanation of license combination, see Section 2.11.

2.3.2 Loading the Licenses for All Products

Use the `reset` command to copy the license details for all products from the license database to the kernel cache. The following shows the command you issue to copy the license details to the kernel cache:

```
# lmf reset
```

If any license has an insufficient number of license units, LMF does not load that license into the kernel cache, and it issues an error message. For example, if the license for ALLSUM has an insufficient number of license units, LMF issues the following message:

```
Not enough units to load ALLSUM DEC
```

If this situation occurs, contact your sales representative, or register and load the license on a smaller system.

Note

LMF loads the license details for all enabled products into the kernel cache whenever a system is restarted.

2.4 Unloading a License

Use the `unload` command to restrict the number of users of a product. The `unload` command removes license units from the kernel cache, restricting the number of units available to LMF checking functions. The command affects only the number of license units available for a product in the kernel cache; it does not affect the number shown in the license database.

Remember that the license will be reloaded with its original number of users whenever you restart your system or issue the `reset` command. To avoid automatic reloading of the license, use the `disable` command. (See Section 2.5 for more information.)

2.4.1 Unloading an Activity License

Suppose you have registered and loaded the license for the product ALLSUM with a 10-user Activity License. You could restrict the license to be a seven-user license by entering the following command:

```
# lmf unload 3 ALLSUM
```

Existing users of the product are allowed to finish using it before LMF imposes the new limit. For example, if 10 users are using a product when you issue the preceding `unload` command, all 10 users are allowed to finish using the product. However, LMF does not allow new users to run the product until the number of active users has dropped below the new limit of seven.

2.4.2 Unloading an Availability License

In the case of an availability-licensed product, unload all the license units for the product. You do this by specifying zero (0) as the number of users. Specifying zero causes LMF to remove the product from the kernel cache. For example, if the product ALLSUM is availability licensed and you want to unload the license units for the product, enter the following command:

```
# lmf unload 0 ALLSUM
```

As with activity-licensed products, LMF allows existing users of the product to finish using it, but LMF does not allow new users to run the product.

2.5 Disabling a License

Occasionally, you might want a license to be registered in the license database, but not loaded into the kernel cache. Once you unload the license from the kernel cache, you can prevent LMF from reloading the license by disabling it. If you do not disable the license, LMF reloads it when you use the `load` or the `reset` command, or when you start your system.

For example, to disable the license for the product ALLSUM and to unload it from the kernel cache, enter the following commands:

```
# lmf disable ALLSUM
# lmf unload 0 ALLSUM
```

The `disable` command does not immediately affect the kernel cache, so you should use the `unload` command to unload the license details from the kernel cache. If you do not use the `unload` command, the product remains available on the system until the next `reset` command or system startup.

2.6 Enabling a License

Once you have disabled a license, LMF ignores it when you issue the `load` or the `reset` command or when you start the system. Use the `enable` command to permit LMF to load the license. After you issue the `enable` command for a license, LMF once again loads the license into the kernel cache when you enter the `load` or the `reset` command or when you start the system.

For example, to enable the license for the product ALLSUM and to load the license into the kernel cache enter the following commands:

```
# lmf enable ALLSUM
# lmf load 0 ALLSUM
```

When you register a license in the license database, it automatically is enabled; that is, you can load it into the kernel cache immediately.

2.7 Issuing a License

To move the license details for a product from the license database to a file on your system, use the `issue` command. The command reconstructs a PAK from the license data in the license database and writes the PAK to the file you specify. If LMF issues the PAK correctly, it unloads the license from the kernel cache and deletes the license from the license database. You can use this command, for example, to move a license for a product from one system to another.

To issue the license for the product ALLSUM to the file `allsum.pak`, enter the following command:

```
# lmf issue allsum.pak ALLSUM
```

Although the command removes the license data from the kernel cache, existing users of the product are allowed to finish using it.

2.8 Canceling a License

You can cause LMF to cancel a license earlier than the date shown by the Key Termination Date field on the PAK. Use the `cancel` command to cancel a license on a specific date.

To cancel the license for the product ALLSUM on the first day of July 1999, enter the following commands:

```
# lmf cancel 1-jul-99 ALLSUM
# lmf load 0 ALLSUM
```

The first command line does not immediately affect the kernel cache, so you should use the `load` command to update the license for the product in the kernel cache.

You can change the cancellation date more than once by issuing the `cancel` command with a different date argument. If you set the cancellation date to be after the key termination date shown on the license, LMF ignores the cancellation date.

2.9 Deleting a License

Once a license expires (for example, because the license has passed its key termination date), delete it from the kernel cache and the license database. Use the `delete` command to delete a license from the kernel cache and license database.

Before you use the `delete` command, be sure that you have a copy of the license data in your files. If you delete a license by mistake, you can restore the license database file (`/usr/var/adm/lmf/ldb`) from a backup, or extract the PAK data from the history file and reregister the license.

For example, to delete the license for the product ALLSUM, enter the following command:

```
# lmf delete ALLSUM
```

Although the command removes the license data from the kernel cache, existing users of the product are allowed to finish using it.

2.10 Updating a License

In some cases, you might need to update an existing license on your system. You can use one of the following two methods to update a license:

- Modify the license with the `modify` command if the license has the MOD_UNITS Key Option.
- Amend the license with the `amend` command if you have a Product Authorization Amendment (PAAM) for the product.

Note

It is current business policy not to issue Product Authorization Amendments (PAAMs). Do not use the `amend` command unless you have a PAAM.

2.10.1 Modifying a License

Use the `modify` command if you want to change the entry in the Comment field or if you want to change the entry in the Number of Units field on a license with the MOD_UNITS Key Option. For example, if the product ALLSUM has the MOD_UNITS Key Option and you want to increase the number of units on the license from 1200 to 2000, enter the following command:

```
# lmf modify ALLSUM
```

In response to this command, LMF runs an editor to allow you to edit the existing license data. LMF runs the editor defined by the `EDITOR` environment variable. If the environment variable is undefined, LMF runs the `vi` editor.

The following example shows the license data that is displayed by LMF for the ALLSUM product and, in bold type, the information you would supply:

```
                 Licensed Software Product
                 Product Authorization Key

         Enter data on lines terminated with :

                         Issuer  DEC
           Authorization Number  aws-pk-88229-2

                   Product Name  ALLSUM
                       Producer  DEC

                Number of units: 2000

                        Version  Version 2.0
           Product Release Date  1-JUL-1999

           Key Termination Date

        Availability Table Code  M
            Activity Table Code

                    Key Options
                  Product Token
                    Hardware Id
                       Checksum  1-odkm-niio-jepj-fclb

Comment: Number of Units increased to 2000.
```

As the preceding example shows, LMF displays the current license for ALLSUM with colons (:) before the Comment field and the Number of Units field. In this example, you use the editor to change the Number of Units from 1200 to 2000. If you change data next to fields without colons, LMF ignores the changes you make. Once you finish entering data, exit from the editor.

When you leave the editor, LMF scans the license data to make sure the license has been updated correctly. If it has not, you are given the opportunity to reenter the editor and correct mistakes.

When you have successfully modified a license, use the `load` command to copy the modified license into the kernel cache.

2.10.2 Amending a License

At times, a Product Authorization Amendment (PAAM) might be issued for your license. A PAAM is used to update an existing license and might have data only in fields that are different from that for the existing license for the product. A PAAM always contains a checksum that is different from that for the existing license. The checksum validates the amended license data; that is, the checksum is generated from the data elements as they appear after the license has been updated with the PAAM data.

Use the `amend` command when you want to update a license in the license database after receiving a PAAM.

Suppose, for example, you already have a license registered for the product ALLSUM and that the license is valid for all versions of the product up to and including Version 2.0. If you want to use the license with versions up to and including Version 2.4, you could contact your support representative who might arrange for you to receive a PAAM. The PAAM might contain entries for product name, version, and checksum. To modify the license database, enter the following command:

```
# lmf amend ALLSUM
```

In response to this command, LMF runs an editor to allow you to edit the existing license data. LMF runs the editor defined by the `EDITOR` environment variable. If the environment variable is undefined, LMF runs the `vi` editor.

The following example shows the license data that is displayed by LMF for the ALLSUM product and, in bold type, the information you would supply from the PAAM:

```
                 Licensed Software Product
                 Product Authorization Key

         Enter data on lines terminated with :

                         Issuer  DEC
           Authorization Number  aws-pk-88229-2

                   Product Name  ALLSUM
                       Producer  DEC

                Number of units  2000

                        Version: Version 2.4
           Product Release Date  1-JUL-1999

           Key Termination Date

        Availability Table Code  M
            Activity Table Code

                    Key Options
                  Product Token
                    Hardware Id
                       Checksum: 1-odkm-qiio-jdpj-fclb
```

```
Comment: Version increased to Version 2.4.
```

As the preceding example shows, LMF displays the current license for ALLSUM with colons (:) before the Version field, the Checksum field, and the Comment field. In the example, you use the editor to change the version number from Version 2.0 to Version 2.4. You also enter the new checksum from the PAAM. LMF removes the checksum entry from the current license because PAAMs always contain a new checksum. If you change data next to fields without colons, LMF ignores the changes that you make.

When you leave the editor, LMF scans the license data to make sure the license has been updated correctly. If it has not, you are given the opportunity to reenter the editor and correct mistakes.

When you have successfully amended a license, use the `load` command to copy the amended license into the kernel cache.

2.11 Combining Licenses

Combining licenses means registering two or more licenses for the same product in the license database and loading them into the kernel cache to form a single license. Licenses that have the NO_SHARE Key Option cannot be combined.

To combine licenses, the following fields must be the same on both original licenses:

- Issuer
- Product Name
- Producer
- Hardware I.D.
- Product Token

Note

Each license must have a different authorization number. LMF does not allow the same license to be registered more than once in the same license database.

When you want to combine licenses, register the licenses in the license database in the usual way, that is, by using the `register` command (see Section 2.2). The licenses appear as separate entries in the license database.

To combine licenses to form a single license in the kernel cache, use the `load` command. For example, if you have registered two licenses for the product

ALLSUM, you can form a single license in the kernel cache by entering the following command:

```
# lmf load 0 ALLSUM
```

The combined licenses appear as one entry in the kernel cache. (For more information about the `load` command, see Section 2.3.)

The number of units for the license is the total number supplied by the original licenses. For example, if the Number of Units field entries on the original licenses are 1000 and 200, the combined license is a 1200-unit license.

The product release date and the key termination date for the combined license are the earlier of those supplied by the original licenses. For example, if the product release dates for the licenses are 1-JAN-1999 and 1-AUG-1999, the combined license has a product release date of 1-JAN-1999.

The version number for the combined license is the lower of those supplied by the original licenses. For example, if the version numbers for the original licenses are Version 1.2 and Version 1.4, the combined license has a version number of Version 1.2.

To remove the combined license from the kernel cache, use the `unload` command (see Section 2.4).

2.12 Displaying Information About Licensed Products

You can use LMF to display the details of products registered on the system, in the following ways:

- Display a summary of all the products registered in the license database or kernel cache or both
- Display the complete license details for all the products in the license database or kernel cache or both
- Display the details of specific products only

To display these details, use the `list` command.

The following example shows how to display a one-line summary of all the products registered in license database:

```
# lmf list
Product      Status       Users: Total      Active
ALLSUM       active       unlimited
NOSUM        disabled
PARTSUM      enabled
ANOTHERSUM   terminated
```

The Status column indicates the current status of the license. Each product can have one of the following six status conditions:

- `active`

  The license has been loaded into the kernel cache and can be used to authorize product use.

- `enabled`

  The license has been registered in the license database but has not been loaded into the kernel cache.

- `disabled`

  The license has been disabled in the license database.

- `terminated`

  The current date is later than the key termination date specified on the license.

- `canceled`

  The current date is later than the cancellation date specified on the license.

- `multiple`

  This license is one of multiple licenses registered for this product name and producer.

The two right-hand columns indicate the amount of product use. For availability-licensed products, the amount of product use is shown as `unlimited`. For activity-licensed products the Total column shows the maximum number of concurrent users allowed for a product. The Active column shows the current number of users for the product.

The following example shows how to display all the license details in the kernel cache for the fictional product ALLSUM (assuming an availability license):

```
# lmf list full cache for ALLSUM


           Product Name: ALLSUM
               Producer: DEC
                Version: Version 2.0
   Product Release Date: 1-JUL-1999
   Key Termination Date:
            Total Units: 1200
           Usable Units: 0
        Activity Charge: 0
```

The Total Units field shows the number of license units in the kernel cache. The Usable Units field shows the number of unallocated license units. The Activity Charge field shows the number of license units required for each product user.

2.13 Reviewing Your License Management Activities

LMF maintains a history file, which is a record of the operations you perform on the license database. LMF records the following commands in the history file:

- `amend`
- `cancel`
- `delete`
- `disable`
- `enable`
- `issue`
- `modify`
- `register`

LMF also creates an entry in the history file when it creates a new license database.

The `history` command displays data in the history file as described in the following list:

- Use the `history` command with no arguments to display the history data for each command issued. The history data comprises the product name, date and time of the command, and the fields that were changed on the license.
- Use the `full` argument to display the history data and the license as it appeared before the command was issued.
- Use the `short` argument to display a one-line summary of the history data for each command issued.
- Use the `from` *`date`* argument to display the history data for commands issued after a certain date.
- Use the `for` *`product`* argument to display the history data for specific products.

The following example shows how to display the history data for the ALLSUM product:

```
# lmf history for ALLSUM
Product Name: ALLSUM
Producer    : DEC
Command     : ENABLE
Date        : 26-FEB-1999
Time        : 12:02:32
Product Name: ALLSUM
Producer    : DEC
```

```
Command     : DISABLE
Date        : 15-JAN-1999
Time        : 11:57:26
Product Name: ALLSUM
Producer    : DEC
Command     : REGISTER
Date        : 4-JAN-1999
Time        : 11:54:15
```

The next example shows the one-line summary of the history data for the same set of LMF commands:

```
# lmf history short for ALLSUM
Product Name  Producer   Command    Date        Time
ALLSUM        DEC        ENABLE     26-FEB-1999 12:02:32
ALLSUM        DEC        DISABLE    15-JAN-1999 11:57:26
ALLSUM        DEC        REGISTER   4-JAN-1999  11:54:15
```

2.14 Changing the Number of Active CPUs

When a system is rebooted, LMF checks the maximum possible number of active CPUs on the system, and uses this value to determine the System Marketing Model. The SMM is used by some products to define the number of license units needed in the kernel cache before access to the product is granted.

If you change the number of active CPUs the SMM may change, and so may the number of license units needed in the kernel cache to access a product. Use the `lmf reset cpus` command to determine a new SMM.

This section describes the license management actions you should take if you change the number of active CPUs:

- For system maintenance purposes
- To reduce the license unit requirement of the system

2.14.1 System Maintenance

If you reduce the number of active CPUs for system maintenance purposes, you do not need to take any special license management actions. LMF continues to use the current SMM even though the number of active CPUs has changed.

When you return to the original number of active CPUs, the LMF continues to use the current SMM, which now accurately reflects the number of active CPUs. Again, you do not need to take any license management actions.

2.14.2 Reducing the License Unit Requirement

You can reduce the license unit requirement of your system by reducing the number of active CPUs on the system. For example, assume you have reduced the number of active CPUs from two to one. To determine the new SMM for the system you would type:

```
# lmf reset cpus
```

Before you return to the original number of active CPUs, determine the new SMM. For example, assume you are ready to increase the number of active CPUs from one to two. To determine the new SMM for the system you would type:

```
# lmf reset cpus 2
```

If you do not determine the new SMM before returning to the original number of active CPUs, the LMF will prevent any further access to the licensed products, although existing users will be able to finish using them.

A

LMF Messages

This appendix lists and explains the messages you might encounter as you use the LMF software. The messages are listed in alphabetical order, and, where appropriate, a course of action is recommended to correct the error that caused the message.

A.1 Accessing Licensed Software

You might encounter the following messages when attempting to access software that provides full support for LMF:

`Attempted usage exceeds active license units`

> You tried to access an activity-licensed product, but the number of units available is less than the license unit requirement for the current processor. This message indicates that the maximum number of simultaneous users of the product has been reached.
>
> Contact your system manager to determine if the maximum number of simultaneous users can be increased. Alternatively, wait until the number of users of the product falls below the maximum and try to access the product again.

`License is invalid for this version of the product`

> You tried to access a product but the version number on the license is lower than the version number of the product you are trying to use.
>
> Contact your system manager. Your system manager might need to install an earlier version of the product or contact your sales representative for a new Product Authorization Key (PAK).

`No license found for this product`

> You tried to access a product that does not have a valid license in the kernel cache.
>
> Contact your system manager.

A.2 Using the lmf Utility

You might encounter the following messages when using the `lmf` utility:

"ALPHA" option missing from PAK entry

You attempted to register a PAK for an availability license that does not contain the ALPHA Key Option. Your system requires this Key Option.

Contact your sales representative to obtain a new PAK that contains the ALPHA Key Option.

A license for *product* cannot be disabled

You tried to disable a license for the product specified.

Certain licenses cannot be disabled; for example, the license created by LMF from information in the /upgrade file.

A license for *product* cannot be issued

You tried to issue a license for the product specified.

Certain licenses cannot be issued; for example, the license created by LMF from information in the /upgrade file.

A license that has been cancelled cannot be enabled

You tried to enable a license that passed its cancellation date.

If you want to enable the license, you should change the cancellation date. For more information, see Section 2.8.

A license that has terminated cannot be enabled

You tried to enable a license that passed its termination date.

You should delete licenses that are passed their termination date from the license database. For information about deleting a license, see Section 2.9.

Activity charge has changed - reboot to load new license for *product producer*

The activity charge has changed for the product specified. This might happen if the license type has changed, for example, from an activity to an availability license, or if the model of system hardware has changed.

Reboot your system to fully reset the kernel cache.

"Activity Table Code" amended - protected field

You changed the Activity Table Code field.

The activity table code should be changed only by using the amend command with a suitable Product Authorization Amendment (PAAM).

`"Activity Table Code" – invalid format`

The activity table code you entered for the license either does not match a valid License Unit Requirement Table code or is not of the form `CONSTANT=`*`integer`*.

When you register the license, you must enter the activity table code exactly as it appears on the Product Authorization Key (PAK).

`"Activity Table Code" missing from PAK entry`

You did not enter an activity table code for the license.

When you register the license, you must enter all the data from the Product Authorization Key (PAK).

`Ambiguous command` *`string`*

The string used as an abbreviation for a command is ambiguous.

When you enter a command, you must use enough letters to distinguish it from other commands.

`"Authorization Number" amended - protected field`

You changed the Authorization Number field.

The authorization number should never be changed because it helps to uniquely identify each license.

`"Authorization Number" missing from PAK entry`

You did not enter an authorization number for the license.

When you register the license, you must enter all the data from the Product Authorization Key (PAK).

`"Availability Table Code" amended - protected field`

You changed the Availability Table Code field.

The availability table code should be changed only by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

`"Availability Table Code" – invalid format`

The availability table code you entered for the license either does not match a valid License Unit Requirement Table code or is not of the form `CONSTANT=`*`integer`*.

When you register the license, you must enter the availability table code exactly as it appears on the Product Authorization Key (PAK).

`"Availability Table Code" missing from PAK entry`

You did not enter an availability table code for the license.

When you register the license, you must enter all the data from the Product Authorization Key (PAK).

`Cannot unload this many users`

You specified too many users with the `unload` command.

Reissue the `unload` command with a lower number of users. To remove all the license units for the product from the kernel cache, specify zero (`0`) as the number of users.

`"Checksum" amended - protected field`

You changed the Checksum field.

The checksum should be changed only by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

`Checksum does not validate`

When you attempted to register a license, the checksum did not validate the license information you entered. The checksum contains, in encrypted form, all the license information from the Product Authorization Key (PAK). If you enter inaccurate license information, you receive this message.

Carefully review all license information on the PAK. When you register the license, you must enter all the information exactly as it appears on the PAK.

`"Checksum" missing from PAK entry`

You did not enter a checksum for the license.

When you register the license, you must enter all the data from the Product Authorization Key (PAK).

`Combine` *`product-authorization-number`* `with` *`product-authorization-number`*

The two licenses shown have been combined to form a single license in the kernel cache.

`Error adding to kernel cache`

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error closing file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error closing license database` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error closing LURT file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error closing temporary file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error closing the history file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error creating license database` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error creating the history file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error determining SMM`

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error locking license database` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error locking the history file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error no entry in LURT for this SMM`

The model of your system hardware does not appear in the License Unit Requirement Table.

Contact your support representative.

`Error opening file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error opening license database` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error opening LURT file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error opening temporary file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error opening the history file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error reading kernel cache`

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error reading license database` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error reading LURT file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error reading the history file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error renaming temporary file` *`filename`* `to license database` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error setting the number of cpus`

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error unlocking license database` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error unlocking the history file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error updating kernel cache`

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error writing to license database` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error writing to the history file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Error writing to temporary file` *`filename`*

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Failed to create process for editor`

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`"Hardware-Id" amended - protected field`

You changed the Hardware I.D. field.

The hardware I.D. should be changed only by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

`History file locked - retrying ...`

You tried to access the history file at the same time as another user.

The `lmf` utility automatically will grant you access to the history file as soon as the other user has finished with it.

`Information provided was ambiguous; multiple licenses were found`

You did not provide enough information for the command to identify an individual license for the product.

If you have two or more licenses with the same product name, you can distinguish them by specifying the producer and the authorization number.

`Internal LMF error was encountered`

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Invalid argument` *`string`*

The string specified was not recognized as a valid argument for the command.

For a complete description of the syntax of the LMF commands, see the `lmf`(8) reference page.

`Invalid entry for availability/activity table code for` *`product producer`*

The availability table code or activity table code names a nonexistent License Unit Requirement Table.

Contact your support representative.

`Invalid LURT entry for` *`product producer`*

The availability table code or activity table code names an invalid License Unit Requirement Table.

Contact your support representative.

`"Issuer" amended - protected field`

You changed the Issuer field.

The issuer should be changed only by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

`"Issuer" missing from PAK entry`

You did not enter an issuer for the license.

When you register the license, you must enter all the data from the Product Authorization Key (PAK).

`"Key Options" amended - protected field`

You changed the Key Options field.

The key options should be changed only by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

`"Key Termination Date" amended - protected field`

You changed the Key Termination Date field.

The key termination date should be changed only by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

`License already registered`

You tried to register a license that is already in the license database.

`License database locked - retrying ...`

You tried to access the license database at the same time as another user.

The `lmf` utility automatically will grant you access to the license database as soon as the other user has finished with it.

`License too small to load this many users`

You specified too many users with the `load` command.

Reissue the `load` command with a lower number of users. To load all the license units for the product into the kernel cache, specify zero (`0`) as the number of users.

`License unchanged`

You left the editor, after you entered an `amend` or `modify` command, without making any changes to the existing license.

`Missing arguments`

You did not specify enough arguments with the command.

For a complete description of the syntax of the LMF commands, see the `lmf`(8) reference page.

`Multiple licenses could not be combined for` *`product producer`*

You tried to combine licenses, at least one of which has the NO_SHARE Key Option.

You cannot combine licenses that have the NO_SHARE Key Option. LMF will not combine licenses if any of the following fields are different: Issuer, Product Name, Producer, Product Token, Hardware I.D.

`No entries in license database`

You tried to list the contents of the license database, but the license database is empty.

`No entry in the history file for this product`

You specified a product name with the `history` command, but no LMF commands were recorded for this product. The history file records only the following LMF commands: `register`, `enable`, `issue`, `cancel`, `delete`, `modify`, and `amend`.

When you use the *`product`* argument, you must specify the product name exactly as it appears on the Product Authorization Key (PAK). If you use the `from` *`date`* argument, verify that the date you specify is not later than the date of the last command recorded for the product.

`No entry in the kernel cache for this product`

The product name you specified in the command does not have an entry in the kernel cache.

When you use the *`product`* argument, you must specify the product name exactly as it appears on the Product Authorization Key (PAK).

`No entry in the license database for this product`

The product name you specified in the command does not have an entry in the license database.

When you use the *`product`* argument, you must specify the product name exactly as it appears on the Product Authorization Key (PAK).

`No valid license was found for this product`

You tried to load an invalid license for a product.

Make sure that you entered the correct product name and that the license is not terminated, disabled, or canceled.

`Not enough memory`

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

`Not enough units to load` *`product producer`*

You tried to load a license for the product specified, but the license does not have enough license units for the current hardware system model.

For a full description of how to provide more license units, see Section 2.10.1.

`"Number of Units" amended - protected field`

You changed the Number of Units field.

The Number of Units field can either be changed by using the `modify` command on a license that has the MOD_UNITS Key Option or by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

`PAK not registered`

You left the editor, after entering a `register` command, without saving the file.

`Permission denied`

You tried to execute an `lmf` utility command but are not logged in to the system as superuser (root login).

Log in to the system as superuser and reissue the command.

`"Producer" amended - protected field`

You changed the Producer field.

The producer should never be changed because it helps to uniquely identify each license.

`"Product Name" amended - protected field`

You changed the Product Name field.

The product name should never be changed because it helps to uniquely identify each license.

`"Product Name" missing from pak entry`

You did not enter a product name for the license.

When you register the license, you must enter all the data from the Product Authorization Key (PAK).

```
"Product Release Date" amended - protected field
```

You changed the Product Release Date field.

The product release date should be changed only by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

```
"Product Token" amended - protected field
```

You changed the Product Token field.

The product token should be changed only by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

```
The kernel cache is empty
```

You tried to list the contents of the kernel cache, but it is empty.

```
The license database file filename is corrupt - restore
most recent backup
```

The license database file has been corrupted by some means and cannot be read by the `lmf` utility.

```
The license database is incompatible with this version of
lmf
```

The license database was created by a version of LMF that is different from the one you are using. The version of LMF you are using cannot use the license database.

Contact your support representative.

```
The LURT file filename is corrupt - restore most recent
backup
```

The License Unit Requirement Table has been corrupted by some means and cannot be read by the `lmf` utility.

Restore the named file from a system backup.

```
Unrecognized cpu for product producer
```

A system error occurred, causing the `lmf` utility to exit with a nonzero error status.

```
"Version" amended - protected field
```

You changed the Version field.

The version number should only be changed by using the `amend` command with a suitable Product Authorization Amendment (PAAM).

```
Warning creating new history file
```

The `lmf` utility has not found an existing history file and is creating a new one.

```
Warning creating new license database
```

The `lmf` utility has not found an existing license database and is creating a new one.

Glossary

This glossary defines a number of terms and acronyms that you should know to understand the LMF documentation.

activity license

A license that defines the number of concurrent users allowed access to a product. For example, a license that provides 1200 license units might provide enough license units to allow 48 users to access the product simultaneously.

authorization number

The unique number assigned by the PAK issuer to a specific PAK. The PAK issuer name and authorization number identify a license.

availability license

A license that makes a product available to all the users of a system. To run on a system, the product must have a certain number of license units. For example, one system might require 1000 license units to run a product, while a more powerful system might require 2000 license units to run the same product.

Any number of users can use the product.

Activity Table Code field

The field on a PAK that represents the number of units required for simultaneous user of the product.

Availability Table Code field

The field on a PAK that represents the number of units required to give unlimited use of a product on a particular hardware system model.

capacity license

See *availability license*

CD-ROM

A medium for the consolidated distribution of software. Layered products can be distributed on a single CD-ROM, with software access authorized by PAKs and LMF.

checksum

An encoded number calculated from the other information supplied with a PAK. LMF uses the checksum to validate the rest of the PAK data. The checksum string always begins with a number, which is the only number in the string. The other 16 positions are always alphabetic characters from A to P.

compact disc read-only memory

See *CD-ROM*

key termination date

A date specified on a PAK that defines when a license contract is no longer valid. Once this date passes, LMF no longer allows users to invoke the product.

LDB

See *license database*

license amendment

The process of updating an existing license by entering data from a Product Authorization Amendment (PAAM) in the license database.

license combination

The process of using the license units from two or more licenses for the same product to provide more product access. Two licenses each with 100 units combine to equal a 200-unit license. Licenses that specify the NO_SHARE Key Option cannot be combined.

license database

A system file that contains the licenses registered on the system.

License Management Facility (LMF)

The part of the operating system that enables the online management of software license data and helps prevent accidental unlicensed use of software.

license PAK

See *Product Authorization Key (PAK)*

license registration

The task you perform when you enter license data from a PAK into the license database. To register a license, use the `register` command.

license unit

The basic unit of measurement used to specify how much product use a license provides. Each license is intended to be used with LMF a size, specified in license units. For example, a license can be a 20-unit license, a 50-unit license, or a 700-unit license.

License Unit Requirement Table (LURT)

A table provided as part of the operating system that specifies a series of license unit requirements, essentially performance ratings, for each processor. Processors that provide more performance (other ratings might be unrelated to performance) have greater license unit requirements.

LMF

See *License Management Facility (LMF)*

LURT

See *License Unit Requirement Table (LURT)*

Number of Units field

The field on a PAK that shows how many license units are supplied with the PAK.

PAAM

See *Product Authorization Amendment (PAAM)*

PAK

See *Product Authorization Key (PAK)*

PAK identification

The PAK issuer name and the authorization number compose the PAK identification. Together, these two items uniquely identify a license.

PAK issuer

The company that creates the license contract for the software. The PAK issuer name and license authorization number uniquely identify a license. In most cases, the software producer is also the PAK issuer. However, PAK issuers can operate under agreement with a separate software producer.

per-user license

See *activity license*

producer

The name of the company that produces the software licensed by LMF.

Product Authorization Amendment (PAAM)

A list of information that amends the license for an existing licensed software product. Without a current PAK or the appropriate PAAM, you might not be able to use an installed software product. A PAAM also contains a unique authorization checksum that validates the license data.

Product Authorization Key (PAK)

A list of essential information about a software license that must be registered in the license database for users to have access to a product. The PAK issuer produces the PAK and delivers it to you, usually as part of your product shipment.

product identification

The software producer name and product name compose the product identification. Together these items uniquely identify a software product for licensing.

product name

The name used by LMF to distinguish among different software products.

Product Release Date field

The field on a PAK that controls when the PAK expires.

software license

A contract between a license producer and a license receiver (customer) that grants permission to use a specific software product as described by the applicable Software Product Description (SPD), and the terms and conditions of the license contract. A PAK supplies the information that results from a software license contract.

System Marketing Model

The hardware system model of the processor.

SPD

See *Software Product Description (SPD)*

Software Product Description (SPD)

The legal document that describes the software product. This document contains the precise product release level that comprises the product version and official product release date.

version number

The number assigned to a particular release of a software product.

user

An entity that meets the license requirements for use of a software product on a certain processor at a particular time. Typically, the user is a user process attempting to run on a processor. The LMF software determines whether or not a user process is affected by LMF. If the user process is affected, LMF then determines whether or not it is valid (licensed to run at this time).

Index