ULTRIX

digital

Guide to Software Licensing

ULTRIX

Guide to
Software Licensing

Order Number: AA-PBKRA-TE

May 1990

Product Version: License Management Facility, Version 1.0

Operating System and Version: ULTRIX, Version 4.0

digital	DECUS	ULTRIX Worksystem Software
	DECwindows	VAX
CDA	DTIF	VAXstation
DDIF	MASSBUS	VMS
DDIS	MicroVAX	VMS/ULTRIX Connection
DEC	Q-bus	VT
DECnet	ULTRIX	XUI
DECstation	ULTRIX Mail Connection	

Contents

The License Management Facility and License Agreements

1 Overview of License Management

2 PAKs and License Enforcement

3 License Management Activities

A   Error Messages

Glossary

Figures

About This Manual

This guide describes the software licensing and license management changes associated with the Digital Distributed Software Licensing Architecture (DDSLA). This includes the new style software license, the Product Authorization Key (PAK) and the License Management Facility (LMF).

The LMF is a software tool for ULTRIX system managers which performs two main functions: license checking and license management.

Audience

This manual is intended for those responsible for managing software licenses on ULTRIX systems. Primarily, this manual is intended for ULTRIX system managers.

This manual also provides information for anyone involved in the use of licensed software on ULTRIX systems.

Organization

This guide is divided into three chapters, an appendix, and a glossary:

Chapter 1   Overview of License Management
                   Provides an introduction to the tools and components associated with the Digital Distributed Software Licensing Architecture (DDSLA)

Chapter 2   PAKs and License Enforcement
                   Describes Product Authorization Keys (PAKs) and how the data they provide is used to prevent unlicensed software use.

Chapter 3   License Management Activities
                   Describes the tasks you can perform using the lmf(8) license management utility.

Appendix A   Error Messages
                   Lists and explains the error messages that you may see when you are using the LMF.

Conventions

The following conventions are used in this manual:

bold	Literals are printed in bold type. Literals define specific commands and options which should be entered exactly as shown.
command(x)	In text, some mentions of the ULTRIX commands include the section number in the reference manual where the commands are documented. For example, lmf(8) appears in Section 8 of the *ULTRIX Reference Pages*.
example	In examples, output text is printed in this typeface.
%	This is the default user prompt in multiuser mode.
#	This is the default superuser prompt.
italics	In syntax descriptions, this typeface indicates terms that are variable.
special	In text, each mention of a specific command, option, partition, path name, directory or file is presented in this typeface.
UPPERCASE	ULTRIX systems differentiate between uppercase and lowercase characters. In examples, enter uppercase characters only where specifically indicated by an example or a syntax line.
[ ]	In syntax descriptions, square brackets indicate terms that are optional.
. . .	In syntax descriptions, a horizontal ellipsis indicates that the preceding item may be repeated.
<KEYNAME>	In examples, a word or abbreviation in angle brackets indicates that you must press the named key on the terminal keyboard.

The License Management Facility and License Agreements

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

The License Management Facility is a system management tool that can help you comply with your license agreement, but use of the LMF does not indemnify you against non-compliance with the terms and conditions of your software license agreements. In other words, the LMF offers options for many kinds of license agreements, but using some of these options may 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 use legally.

This document describes some features of the LMF that Digital is not currently using. Digital may in future use some of the features described herein, but makes no commitment beyond the current Software Business Practices.

Overview of License Management   1

The ULTRIX License Management Facility (LMF) provides software tools for ULTRIX system managers to manage software licenses. The License Management Facility also performs software license checking when software is used that provides full LMF support.

The License Management Facility is part of an approach to software licensing called the Digital Distributed Software Licensing Architecture (DDSLA). DDSLA is an engineering architecture based on the idea of using software license units as a way to size customer computing environments. The license unit is a basic measurement that Digital Equipment Corporation uses to specify how much product use a software license provides. In general, processors that provide more performance need software licenses with more license units.

1.1 The Purpose of DDSLA

When every computer system was a single processor, and software products were dedicated to that processor, both software licensing and the management of software licenses were relatively simple and straightforward. But in today's distributed computing environment, a system manager faces more complexity.

Distributed computing allows a much wider variation in software use than a single processor. Software may be used system-wide, or by just a few users. Who uses it, and where it can be used, may change with the computer environment.

These factors have resulted in a new approach to software licensing and the introduction of new tools for managing license data.

The Digital Distributed Software Licensing Architecture (DDSLA) is an approach to software licensing that:

- Is consistent for all ULTRIX products

- Provides for organized record-keeping

- Includes useful tools to help you monitor and control software use

1.2 The Components of DDSLA

The implementation of the Digital Distributed Software Licensing Architecture (DDSLA) consists of:

- The License Management Facility

- Product Authorization Keys (PAKs)

The License Management Facility is part of the ULTRIX operating system and consists of:

- A License Database (LDB)

- License Unit Requirement Tables (LURTs)

- A license management utility, lmf(8)

- License-checking functions which are included in the licensed products

A Product Authorization Key (PAK) is a set of license information which the License Management Facility uses to confirm that a product is licensed. Licenses are normally printed on paper and sent to you as part of the product kit.

When you receive a PAK you should enter the license information into the License Database using the lmf(8) utility. This is called "registering a license" and some licensed products will not run unless you have registered a license for them. In some cases you may need to register a license before you install the product. The License Database is automatically created when you register the first license. The PAK is your proof of license, and should be stored in your files for future reference.

While the number of license units on the PAK defines the size of the license, each processor has a series of license unit requirements, also specified in license units. License Unit Requirement Tables define the number of license units required to run a product on a particular size of processor. The LMF compares the size of a registered license to the license unit requirement for the processor, and authorizes product use when a license supplies sufficient license units.

1.3 Types of Software License

There are two basic types of license:

- Availability (or Capacity) Licenses

- Activity (or Per-user) Licenses

The next two sections describe each type.

1.3.1 Availability Licenses

Availability Licenses are also called Capacity Licenses. Availability Licenses only allow a product to run on certain types of executing hardware; the more powerful the hardware, the more license units are needed for each product. Once a product has been licensed on a particular processor, users have unlimited access to the product. Unlike previous Capacity Licenses, DDSLA defined Availability Licenses are not restricted to a particular processor before they are registered; that is, the license does not state which processor it must be registered on.

1.3.2 Activity Licenses

Activity Licenses are also called Per-user Licenses. Activity Licenses restrict the number of simultaneous users of a product. The number of licensed users can be increased by purchasing additional license units.

1.4 Your Responsibility in License Management

Software is provided to customers under 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 a

license refers to the authorization you have to use a product.

Note

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

PAKs and License Enforcement 2

Product Authorization Keys (PAKs) and license checking by the License Management Facility are being gradually introduced for ULTRIX software. This chapter is designed to help you understand PAKs and how license enforcement by the LMF will affect your use of ULTRIX software.

This chapter describes:

- Which products are affected by the LMF

- Product Authorization Keys

- License enforcement for availability licensed and activity licensed products

2.1 Products Affected by the LMF

In general, Digital Equipment Corporation's software products fall into two categories:

- Software unaffected by the LMF
  These products do not use the LMF to authorize software access. License information is provided separate from the software and may be in the form of a PAK which can be registered in the License Database.

- Software that provides full LMF support
  License information is provided by a PAK which must be registered in the LDB. The LMF checks for unauthorized use of these products.

The release date of a product, relative to the release date of ULTRIX V4.0, normally determines how much support a product provides for the LMF:

- Software released before ULTRIX V4.0 is unaffected by the LMF.

- Software released after ULTRIX V4.0 provides full LMF support.

The software that could be affected by the LMF includes Digital layered products and operating systems based on ULTRIX, such as ULTRIX Worksystems Software (UWS).

Note

Layered products that provide full LMF support can only be used on operating systems based on ULTRIX V4.0 (or later). Operating systems based on prior releases of ULTRIX do not have the License Management Facility to authorize access to these products.

The LMF is not designed to be used exclusively by Digital products. Other companies may issue PAKs, or have Digital issue them on their behalf, and include license-checking functions in their software. However, for clarity, in this manual it is assumed that all software is supplied and produced by Digital.

You should refer to the product documentation to find out if the product provides support for the License Management Facility.

2.2 Product Authorization Keys

A Product Authorization Key is a unique set of data generated by Digital and used by the LMF to confirm that a product is licensed.

The License PAK is the "standard" PAK, provided when you purchase a software license. It is a valuable proof of purchase, represents your license from Digital to use a software product, and should be stored in your files. The license information is confidential and should not be publicly posted or widely distributed. In order to comply with Digital's license terms, you must always register a License PAK in the License Database.

2.2.1 Obtaining a PAK

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

2.3 Other Types of PAK

As well as License Product Authorization Keys, there are three other types of PAKs Digital uses to cover various situations:

- Service Update PAK
- Temporary Service PAK
- Product Authorization Amendment

2.3.1 Service Update PAK (SUP)

This is a transitional tool provided to software maintenance customers as part of a layered product's next scheduled update, when the product provides technical support for the LMF. A Service Update PAK contains the same fields as a License PAK and the information should be registered in the LDB using the lmf register command. For more information on lmf register, see Section 3.2. Unlike a License PAK, a SUP does not represent a software license; it is merely a temporary measure that you use if you do not yet have a License PAK for a layered product. A License PAK always supersedes a Service Update PAK.

2.3.2 Temporary Service PAK (TSP)

This is a temporary exception measure used by Digital Field Service in performing customer support activities. A Temporary Service PAK contains the same fields as a License PAK and the information should be registered in the LDB using the lmf register command (see Section 3.2). Again, this type of PAK does not represent a license and will be replaced with a License PAK if appropriate.

2.3.3 Product Authorization Amendment

A Product Authorization Amendment (PAAM) is similar to a License PAK but only includes the data needed to identify, update, and further authorize product use. For example, you may receive a PAAM if you want to increase the number of units for a license that does not have the MOD_UNITS option. The license for the product should be updated using the `lmf amend` command. For more information on `lmf amend`, see Section 3.12.2.

2.4 Information on a PAK

This section describes the information contained in the fields on the PAK. A typical PAK is shown in Figure 2.1 When you register the PAK in the LDB, you will be provided with a Comments field that you can use as needed. Section 3.2 describes how to register a PAK.

Figure 2-1: A Typical PAK

```
 _ _ _ _ _ _
| | | | | | | |                         LICENSE PAK
|d|i|g|i|t|a|l|              (PRODUCT AUTHORIZATION KEY)
| | | | | | | |
 _ _ _ _ _ _
Digital Equipment Corporation


The Software License Product Authorization Key is provided subject to
terms appearing on the back of this document.

                    Product: ALLSUM ACCOUNTING TOOL

License Model No. QL-010AB-AA     DEC No. 125436     Issue Date: 1-JAN-1989

*****************************************************************************

                    ISSUER:            DEC

        AUTHORIZATION NUMBER:          AWS-PK-88229-2

                PRODUCT NAME:          ALLSUM

                    PRODUCER:          DEC

            NUMBER OF UNITS:           1200

                     VERSION:

        PRODUCT RELEASE DATE:          1-JUL-1991

         KEY TERMINATION DATE:

      AVAILABILITY TABLE CODE:         M

          ACTIVITY TABLE CODE:

                 KEY OPTIONS:

               PRODUCT TOKEN:

                HARDWARE I.D:

                    CHECKSUM:          1-ODKM-NIIO-JEPJ-FCLB

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

2.4.1 Issuer

The Issuer is the LMF name for the entity that supplies the PAK. Most licenses specify "DEC" for the PAK issuer. However, Digital may further identify PAK issuers 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 provide support for the LMF can also issue PAKs.

2.4.2 Authorization Number

The Authorization Number, together with the Issuer, uniquely identifies each license, both for you and for Digital. It helps tasks such as reconciling records between customers and Digital, because it allows everyone to know immediately and with certainty which license is being referred to.

2.4.3 Product Name

The Product Name is the name used by the LMF to distinguish between different products. The Product Name that appears on the PAK may be slightly different from that in the Software Product Description. This is due to restrictions imposed by the LMF.

2.4.4 Producer

The Producer is the name of the company producing the software. For all software produced by Digital, the producer name will be "DEC." The Producer is used by the LMF to distinguish between products with the same name but produced by a different company. For example, you might have two FORTRAN compilers, one produced by Digital and one produced by another company.

2.4.5 Number of Units

The Number of Units shows how many license units have been supplied with the PAK. Licenses with a Number of Units shown as zero (0) indicate that the license has unlimited size, that is, it provides unlimited use of the product on any type of processor.

2.4.6 Version

This is the software product version number. The Version will not appear on every product's PAK. It is used by the LMF to restrict the use of the PAK to particular versions of the product. For example, if the Version number on the PAK is 2.0, then the PAK may be used with all versions of the product up to and including Version 2.0.

2.4.7 Product Release Date

The Product Release Date will not appear on every product's PAK. The Product Release Date is used by the LMF to restrict the use of the PAK to versions of the product released before a certain date. For example, if the Product Release Date on the PAK is 1-JUL-1991, then the PAK is valid for use with all versions of the product released on or before that date.

Digital does not issue PAKs that have both the Version and the Product Release Date on them. Some PAKs may be issued that are not restricted by Version or by Product Release Date.

2.4.8 Key Termination Date

This is the termination date of the PAK. After this date, the PAK no longer represents a valid license for the product.

2.4.9 Availability Table Code

This represents the number of units required to give unlimited use on a particular processor. If the product is availability licensed, this field will contain a letter or "CONSTANT=*integer*." A letter represents the License Unit Requirement Table that defines the number of units required for the product to run on a particular processor. 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 type of processor, regardless of size. For a complete explanation of how the Availability Table Code is used, see Section 2.6.1.

2.4.10 Activity Table Code

This represents the number of units required for each simultaneous user of the product. If the product is activity licensed, this field will contain a letter or "CONSTANT=*integer*." A letter represents the License Unit Requirement Table that defines the number of units required for each simultaneous user to run the product on a particular processor. If your PAK has an Activity Table Code with, for example, "CONSTANT=100," it means that each simultaneous user of the product needs 100 units to run the product on any type of processor, regardless of size. For a complete explanation of how the Activity Table Code is used, see Section 2.6.2.

2.4.11 Key Options

There are three options which can appear in this field:

* MOD_UNITS

* NO_SHARE

* P_FAMILY

The MOD_UNITS option indicates that you may modify the Number of Units field (using lmf modify). For a complete description of how to modify the Number of Units, see Section 3.12.1.

The NO_SHARE option indicates that you cannot combine two or more licenses for the product on the same processor. For a complete description of license combination, see Section 3.20.

The P_FAMILY option indicates that the LMF will attempt to allocate license units for the product to only one executing process, even if a user has several processes executing the same licensed software.

2.4.12 Product Token

This field is not currently used by the LMF. However, any data that is in this field must still be entered into the License Database to prevent a Checksum error.

2.4.13 Hardware-Id

This field is not currently used by the LMF. However, any data that is in this field must still be entered into the LDB to prevent a Checksum error.

2.4.14 Checksum

The Checksum is generated from the individual data elements on the PAK. The Checksum will be unique for each PAK and ensures that you have entered the PAK data correctly into the LDB.

2.5 License Unit Requirement Tables

License Unit Requirement Tables (LURTs) are provided as part of the License Management Facility. LURTs are a series of tables that specify a series of license units requirements, essentially performance ratings, for each System Marketing Model (SMM). Although this manual generally refers to computer systems as processors, the LMF actually identifies a system by its System Marketing Model, which is the model name used in marketing and pricing. The SMM generally corresponds to the name on the front panel of the processor cabinet.

Each LURT has a rating, in license units, for all currently available (and appropriate) processors. For example, the ULTRIX Layered Product LURT includes every processor that can run ULTRIX, and associates a number of units with each. When Digital releases new processors, the tables are updated as part of the new processor support. There are 14 LURTs, with the following codes and types:

A – H	VMS Tables
J	ULTRIX Capacity
K	ULTRIX 'n' User
L	System Integrated Products
M	Standard Layered Products
N	Reserved for future use
P	Reserved for future use

The LURT tables are not stored in a readable form on ULTRIX systems, but this will not prevent you ordering licenses of an appropriate size for your system. When you order a product, you should define your software and hardware needs to your Digital representative, who will ensure you receive a license with the appropriate number of units.

2.6 License Checking

The LMF authorizes a product to run only if there is a valid license for the product. Products that provide technical support for the LMF in their software have license-checking functions that check the following:

- Software has the same Product Name and Producer name as those on the license

- Version number of the software is not greater than the Version number (if specified) on the license

- Product release date of the software is not later than the Product Release Date (if specified) on the license

- Current date is not later than the Key Termination Date (if specified) on the license

- Current date is not later than the Cancellation Date (if specified) on the license

The license-checking functions check the license details in the kernel cache. The kernel cache contains details of those licenses that have been registered in the License Database and subsequently loaded into the kernel cache. License details are loaded into the kernel cache when the following occurs:

- The system is rebooted

- The lmf load command is used (see Section 3.5.1)

- The lmf reset command is used (see Section 3.5.2)

Note

You must load a valid license for a product into the kernel cache before attempting to access the product.

The number of license units registered with any license should match or exceed the number of license units required for the specified product to run on the specified processor. For example, suppose you obtain a license for the fictional product ALLSUM to run on a MicroVAX II. That ALLSUM license should specify at least the same number of license units as the ULTRIX Layered Product LURT requires for a MicroVAX II. The same license may not provide enough license units to authorize use of the product on a VAX 8800.

For a complete explanation of license checking for availability licensed products, see Section 2.6.1. For a complete explanation of license checking for activity licensed products, see Section 2.6.2.

2.6.1 Availability Licensed Products

A valid Availability License makes a product available to all the users of a system. The LMF makes a product accessible if the number of units on the license matches or exceeds the license unit requirement for the current processor. Availability licenses are checked:

- By the LMF, when the license details are loaded into the kernel cache from the LDB

- By the license-checking function in the product, when a user attempts to access the licensed software

2.6.1.1 Loading a License into the Kernel Cache — When you load a license into the
kernel cache the LMF looks for the Availability Table Code field of the registered
license. If the license specifies "CONSTANT=*integer*," the LMF defines the license
unit requirement as equal to the stated integer value. This value can be the decimal
value zero (0), which means the license has no unit requirements.

If the license does not specify a constant unit requirement, the LMF looks for a code
that corresponds to an entry in the LURT. The LMF determines the System
Marketing Model of the current processor and locates the SMM in the appropriate
LURT. The LMF selects the value that specifies the number of units required and
compares this against the number of units on the license.

If the number of units on the license matches or exceeds the license unit requirement
for the current processor, license details from the License Database are copied into
the kernel cache, and the product becomes accessible by all users on the system.

If the number of units on the license is less than the license unit requirement for the
current processor, the license details are not copied into the kernel cache and an error
message is displayed. If you are trying to load a license into the kernel cache with
the `lmf reset,` command, or by rebooting the system, the error message displayed
is:

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

If you are trying to load a license into the kernel cache with the `lmf load`
command, the error message displayed is:

```
License too small to load this many users
```

Consider an example where all the PAKs for the fictional layered software product
ALLSUM refer to LURT M, designated by the letter M next to the Availability Table
Code field on each PAK. Your PAK for ALLSUM may also provide 1000 license
units, designated by the number 1000 next to the number of units field on the PAK.

When you register and load the 1000-unit license, the LMF selects LURT M, and
compares the license unit value 1000 to the value found in LURT M next to the
current SMM. For this example, assume the current processor, VAXMID, requires
1000 license units to activate a layered product in LURT M. The LMF allows the
license details from the License Database to be copied into the kernel cache and the
product becomes accessible by all users on the system.

Now consider the current processor to be VAXBIG, and assume VAXBIG requires a
1500-unit license to activate the product ALLSUM. Because the number of units on
the license is less than the license unit requirement for the current processor, the
license details are not copied into the kernel cache, and the appropriate error message
is displayed.

2.6.1.2 Accessing the Licensed Software — Each time a user attempts to access an
availability licensed product, the license-checking function in the layered product
checks the kernel cache for a valid license for the product; that is, it performs the
checks described in Section 2.6. No check is made on the number of license units
for the product because the LMF only allows availability licenses with a sufficient
number of units to be loaded into the kernel cache.

Availability licensed products that have a valid license in the kernel cache can be
accessed by all users on the system.

If a user attempts to access an availability licensed product that does not have a valid
license in the kernel cache, the license-checking function prevents access to the

product and displays the following message on the terminal:

```
No license found for this product
```

This situation occurs when, for example, you have registered the license details for the product in the LDB and installed the software, but you have not yet rebooted the system or used `lmf load` or `lmf reset`.

Note

Not all license-checking functions behave as described in the previous section. Some may prevent access to the product without displaying an error message, others may allow users to access the product, even if the product does not have a valid license in the kernel cache. You should refer to the documentation supplied with the layered product to find out exactly what action the checking function will take.

2.6.2 Activity Licensed Products

An Activity License defines the number of simultaneous users allowed for a product. The LMF makes a product available to a user if the number of units on the license matches or exceeds the license unit requirement for the current processor. Activity license are checked:

*   By the LMF, when the license details are loaded into the kernel cache from the LDB

*   By the license-checking function in the product, when a user attempts to access the licensed software

2.6.2.1 Loading a License into the Kernel Cache – When you load a license into the kernel cache the LMF looks for the Activity Table Code field of the registered license. If the license specifies "CONSTANT=*integer*," the LMF defines the license unit requirement for each user as equal to the stated integer value. This value can be the decimal value zero (0), which means each user has no unit requirements.

If the license does not specify a constant unit requirement, the LMF looks for a code that corresponds to an entry in the LURT. The LMF determines the System Marketing Model of the current processor, locates the SMM in the appropriate LURT and selects the value that specifies the number of units per user required.

The number of units per user required by the current processor and the license details from the LDB are copied into the kernel cache.

2.6.2.2 Accessing the Licensed Software – Each time a user attempts to access an activity licensed product, the license-checking function in the product checks the kernel cache for a valid license for the product, that is, it performs the checks described in Section 2.6.

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 access the product. When the license-checking function allows the first user to access a product, it allocates the number of units required from the kernel cache. Using the fictional ALLSUM again, the license-checking function may allocate 25 of a registered 100 units to the first

user. As long as the first user is using the product, those 25 units remain allocated, leaving 75 available in the kernel cache for other users. If the number of units on the license is less than the license unit requirement for the current processor, the user is refused access to the product and the following message is displayed at the terminal:

```
Attempted usage exceeds active license units
```

When the next user attempts to use the product, the checking function repeats the authorization procedure again. For example, 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 75 license units unallocated in the kernel cache, the license-checking function again authorizes product use. In this example, the first four concurrent users can access the product, but additional users are denied access.

As each user finishes using the product, the kernel returns the allocated units for another user.

If a user attempts to access an activity licensed product that does not have a valid license in the kernel cache, the license checking function prevents access to the product and displays the following message on the terminal:

```
No license found for this product
```

This situation occurs when, for example, you have registered the license details for the product in the LDB and installed the software, but you have not yet rebooted the system or used `lmf load` or `lmf reset`.

Note

Not all license-checking functions behave as described in the previous section. Some may prevent access to the product without displaying an error message, others may allow users to access the product, even if the product does not have a valid license in the kernel cache. You should refer to the documentation supplied with the layered product to find out exactly what action the checking function will take.

License Management Activities 3

The License Management Facility provides the lmf utility to help manage the software licenses for your system. The lmf utility maintains a file of registered software licenses, the License Database. In addition to maintaining the LDB, you can also use the lmf utility to control the access to licensed software on the system. This chapter describes how to:

- Use the lmf utility
- Register a license
- Activate a license
- Update a license
- Restrict the use of a product
- Disable a license
- Cancel a license
- Delete a license
- Monitor the LDB and kernel cache
- Review your license management activities
- Change the number of active CPUs
- Combine licenses
- Manage availability licensed products
- Manage activity licensed products

3.1 Using the lmf Utility

The lmf commands can only be used by a person logged into the system as the superuser (root login). You can allow nonprivileged users to use the lmf list and lmf history commands, but you need to change the file mode permissions on the files the commands access. The lmf list command accesses /usr/var/adm/lmf/ldb, and the lmf history command accesses /usr/var/adm/lmf/ldb_history. You can change the directory containing the LDB file and the history file by using the -d *dir* option. This allows you to have more than one LDB on your system.

When you use the lmf commands you can type them on a single line, for example:

lmf register

You can also enter the lmf utility and type the commands after the prompt, for example:

lmf

```
lmf> register
```

3.2 Registering a License

This section describes how to use the `lmf register` command to register license details from a Product Authorization Key in the License Database. Using `lmf register` you can:

- Edit an empty template and register the license details from the completed template

- Edit an existing template and register the license details from the completed template

- Register details directly from a file or electronic-mail message

The LMF checks the license details you are trying to register, to ensure that there are entries against all the appropriate fields. The following fields always require an entry:

- Issuer

- Authorization Number

- Product Name

- Availability Table Code or Activity Table Code, or both

- Checksum

If the Producer field is blank, the LMF assumes the Producer to be "DEC." The LMF ensures there are entries against all the mandatory fields and that the Checksum validates all the license data. Licenses with incorrect or missing entries are not registered in the LDB.

The following sections describe the three ways to register licenses in the LDB and explain what to do if something goes wrong.

3.2.1 Editing an Empty Template

Use the `lmf register` command with no arguments to add license details to an empty template and register the details in the LDB. The command displays a template which includes all the fields on a PAK and an additional field for your comments. An editor is invoked so that you can add the license data to the appropriate fields. The editor used is defined by the EDITOR environment variable. If this is not set, `/usr/ucb/vi` is used.

You need to use `lmf register` to register license information from a paper License PAK, see Figure 3.1.

Figure 3-1: A Typical PAK

```
 _ _ _ _ _ _ _
| | | | | | | |                     LICENSE PAK
|d|i|g|i|t|a|l|             (PRODUCT AUTHORIZATION KEY)
| | | | | | | |
 _ _ _ _ _ _ _
Digital Equipment Corporation


The Software License Product Authorization Key is provided subject to
terms appearing on the back of this document.

                    Product: ULTRIX OPERATING SYSTEM

License Model No. QR-234AB-DC    DEC No. 458394    Issue Date: 1-JAN-1989

*********************************************************************

                    ISSUER:            DEC

      AUTHORIZATION NUMBER:            TPQ-PK-88229-23

              PRODUCT NAME:            ULTRIX

                  PRODUCER:            DEC

           NUMBER OF UNITS:            1600

                   VERSION:

      PRODUCT RELEASE DATE:            1-JUL-1991

       KEY TERMINATION DATE:

    AVAILABILITY TABLE CODE:

       ACTIVITY TABLE CODE:            CONSTANT=100

               KEY OPTIONS:            NO_SHARE,P_FAMILY

             PRODUCT TOKEN:

               HARDWARE I.D:

                  CHECKSUM:            1-AFAP-ICFD-BCPJ-FCGL


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

To register the license data from the example PAK you type:

lmf register

The command displays an empty template, and you should use the editor to type in the PAK data against the appropriate field:

```
                    Licensed Software Product
                    Product Authorization Key

Enter data on lines terminated with :

                Issuer: dec
   Authorization Number: tpq-pk-88229-23

          Product Name: ultrix
              Producer: dec

        Number of units: 1600

               Version:
  Product Release Date: 1-jul-1991

    Key Termination Date:

Availability Table Code:
     Activity Table Code: constant=100

           Key Options: no_share,p_family
         Product Token:
          Hardware-Id:
              Checksum: 1-afap-icfd-bcpj-fcgl

Comment: This is an example license
```

Any fields that are blank on the PAK should be left blank when you type in the data. You can type the data in uppercase or lowercase; when the data is copied into the LDB, it is automatically put into uppercase.

Note

You must type in the license information from the PAK carefully. The LMF may return only a checksum error message if you omit or incorrectly enter any license data. Carefully check the characters typed on each line, not just the checksum string.

When you leave the editor, the LMF scans the completed template to ensure that all the license data has been entered correctly. If the license data is correct, it is copied into the License Database. If the license data is incorrect, you are given the opportunity to re-enter the editor and correct any mistakes.

3.2.2   Editing an Existing Template

Use the lmf register *filename* command to edit and register license data from a file on your system. The editor invoked is defined by the EDITOR environment variable. If this is not set, /usr/ucb/vi is used.

You may have license data in files on your system as a result of using the lmf issue command (see Section 3.9), or they may be copied on to your system as part of a product installation. For example, the installation software for ULTRIX operating systems copies license data to the file /usr/var/adm/lmf/ultrix. This file contains license data common to all ULTRIX operating system licenses. To register the license, you just need to add your specific license details from your ULTRIX PAK. Using the example PAK shown in Figure 3-1, to register the

additional data on to the file created during installation, type the following:

lmf register /usr/var/adm/lmf/ultrix

The command displays the file, in this example /usr/var/adm/lmf/ultrix, and you should use the editor to complete the license details:

```
                    Licensed Software Product
                    Product Authorization Key

  Enter data on lines terminated with :

                    Issuer: DEC
        Authorization Number: tpq-pk-88229-23

              Product Name: ULTRIX
                  Producer: DEC

          Number of units: 1600

                   Version:
      Product Release Date: 1-JUL-1991

      Key Termination Date:

  Availability Table Code:
       Activity Table Code: CONSTANT=100

              Key Options: NO_SHARE,P_FAMILY
            Product Token:
              Hardware-Id:
                 Checksum: 1-afap-icfd-bcpj-fcgl

Comment: This is an example license
```

You must ensure that the file you are registering contains all the license fields that have entries. The license fields must be in the same format as the template displayed with lmf register, that is, the same combination of uppercase and lowercase letters, with a colon (:) separating the field name and the data. The license fields can be in any order. Files with license data created using the lmf issue command or as part of product installations automatically have the field names in the correct format.

When you leave the editor, the LMF scans the completed template to ensure that all the license data has been entered correctly. If the license data is correct, it is copied into the License Database. If the license data is incorrect, you are given the opportunity to re-enter the editor and correct any mistakes.

3.2.3 Registering a PAK Directly

Use the lmf register – < *filename* command to register a file containing PAK data. The file can be created by the lmf issue command or may be an electronic-mail message containing PAK data.

The command does not display the contents of the file or allow you to edit it. However, the LMF does scan the file to ensure the format and data are correct. If the license data is correct, it is copied into the LDB. If the license data is incorrect, it is not copied into the LDB and the appropriate error message is displayed. The command also returns an error status: zero (0) if the license data has been copied into the LDB, or nonzero, if the license data has not been copied into the LDB.

3.3 Registering Licenses for ULTRIX Systems

This section describes how to register the license for your ULTRIX operating system. It provides an overview of the licensing events that occur when you install your ULTRIX system and describes two situations in particular:

- Installing a new ULTRIX system

- Upgrading an existing ULTRIX system

When you install your ULTRIX operating system, the installation software reboots the system. When a system is rebooted, the LMF tries to load the kernel cache with the license details from the License Database. If the LMF finds an ULTRIX license in the LDB file /usr/var/adm/lmf/ldb), it (as well as any other licenses) is loaded into the kernel cache. If the ULTRIX license was generated automatically or if there is no ULTRIX license, the LMF searches for the /upgrade file. The /upgrade file contains the maximum number of concurrent users allowed for ULTRIX.

If the /upgrade file exists, the LMF loads enough license units into the kernel cache to allow the number of concurrent users specified by the /upgrade file. The LMF also creates an entry for an ULTRIX license in the LDB, with a comment saying that the entry was created automatically.

If the /upgrade file does not exist, the LMF loads enough units into the kernel cache to allow two concurrent users. The LMF does not create an entry in the LDB for ULTRIX, if there is no /upgrade file.

When you receive your Product Authorization Key for ULTRIX, you should use the lmf delete command to remove the ULTRIX license data generated automatically by the LMF. Register the PAK for ULTRIX using the lmf register command and then load the license into the kernel cache using the lmf load command. The following sections provide more details on these commands.

3.3.1 New Systems

When you install ULTRIX on a new system and the installation software reboots the system, the LMF loads enough units into the kernel cache to allow two concurrent users.

When you receive your ULTRIX PAK, you should register it in the LDB. License data common to all ULTRIX operating system licenses is contained in the file /usr/var/adm/lmf/ultrix. You can use this data when you register your ULTRIX PAK. To display /usr/var/adm/lmf/ultrix, type the following command line:

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

Use the editor and the information on your ULTRIX PAK to complete the license details (see Section 3.2.2).

When you have left the editor and the license has been registered in the LDB, you should load the license into the kernel cache using the command:

```
# lmf load 0 ultrix
```

3.3.2 Existing Systems

When you install ULTRIX on an existing system there may be an existing LDB or a binary upgrade file. If there is an ULTRIX license in the LDB, when the installation software reboots the system the LMF loads the license into the kernel cache. In this case you need take no further action to register your ULTRIX license.

If the ULTRIX license was generated automatically or if there is no ULTRIX license when the system reboots, the LMF searches for the /upgrade file. If the /upgrade file exists, the LMF loads enough license units into the kernel cache to allow the number of concurrent users specified by the /upgrade file. The LMF also creates an entry for an ULTRIX license in the LDB, with a comment saying that the entry was created automatically.

If the /upgrade file does not exist, the LMF loads enough units into the kernel cache to allow two concurrent users. The LMF does not create an entry in the LDB for ULTRIX, if there is no /upgrade file.

When you receive your ULTRIX PAK, you should remove the ULTRIX license generated automatically by the LMF. Use the command:

```
# lmf unload 0 ultrix
```

This unloads the ULTRIX license from the kernel cache. If the ULTRIX license was created without using data from the /upgrade file, there is no entry in the LDB for you to delete. If the ULTRIX license was created from data in the /upgrade file, use the command:

```
# lmf delete ultrix
```

This deletes the license from the LDB.

The installation software copies license data common to all ULTRIX operating systems licenses to the file, /usr/var/adm/lmf/ultrix. You can use this data when you register your ULTRIX PAK. Use the command:

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

This displays /usr/var/adm/lmf/ultrix. Use the editor and the information on your ULTRIX PAK to complete the license details (see Section 3.2.2).

When you have left the editor and the license has been registered in the LDB, you should load the license into the kernel cache using the command:

```
# lmf load 0 ultrix
```

3.4 Registering Licenses for Layered Products

You can register a license for a layered product by:

- Editing a blank template
- Editing an existing file

3.4.1 Editing an Empty Template

Use this method to register a license for a product that:

- Has not been installed yet
- Does not create a file containing license data as part of its installation procedure

Some layered products require a valid license in the kernel cache before they can be installed. These layered products run as part of their installation procedure.

To edit an empty template, use the command:

```
# lmf register
```

This displays a template which includes all the fields on a PAK. You should use the editor to type in the PAK data against the appropriate fields on the template (see Section 3.2.1)

When you have left the editor and the license has been registered in the LDB, you should load the license into the kernel cache using the following syntax:

```
lmf load 0 product_name
```

In this command example, *product_name* is the same as the Product Name on the PAK.

3.4.2 Editing an Existing Template

Some layered products create a file containing license data as part of their installation procedure. The license data is common to all PAKs for the product; to register the license, you just need to add your specific license details from the PAK you receive. The license data is copied to the file /usr/var/adm/lmf/*product_name* where *product_name* is the LMF Product Name as it appears on the PAK. For example, the product ALLSUM would copy license data to the file /usr/var/adm/lmf/allsum.

To register a license using data from a file created as part of the product installation, use the command:

```
lmf register /usr/var/adm/lmf/product_name
```

Use the editor to type in the PAK data against the appropriate fields on the template (see Section 3.2.2).

When you have left the editor and the license has been registered in the LDB, you should load the license into the kernel cache using the following command syntax:

```
lmf load 0 product_name
```

3.5 Loading a License into the Kernel Cache

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

To load license details into the kernel cache, copy license details from the LDB by:

- Using the lmf load command. This command copies the license details for a particular product from the LDB to the kernel cache.

- Using the lmf reset command. This command copies the license details for all products from the LDB to the kernel cache.

- Rebooting the system. The reboot process automatically executes the lmf reset cpus command.

3.5.1 Loading a License for One Product

Use the `lmf load` command to copy the license details for a specified product from the LDB to the kernel cache. The `lmf load` command has the following syntax (see also the `lmf(8)` reference page):

lmf load *users product* [*producer* [*authorization*]]

The LMF loads the number of license units corresponding to the number of *users* specified in the command into the kernel cache (assuming that there are enough license units registered in the LDB). If you specify zero (0) as the *users* argument, the LMF loads all the license units registered for the product in the LDB into the kernel cache.

If the product is availability licensed, you must specify the number of users to be zero when you use the `lmf load` command. This ensures that 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 LDB. To load the license into the kernel cache, you should type:

```
# lmf load 0 ALLSUM
```

Alternatively, assume the product ALLSUM is activity licensed, and you have registered a 10-user license for the product in the LDB. To load all the license units for the product into the kernel cache, you should type:

```
# lmf load 0 ALLSUM
```

If you only wanted to load enough license units for 5 users, you should type:

```
# lmf load 5 ALLSUM
```

When you use the `lmf load` command you must ensure that you supply enough arguments to uniquely identify the license. If you have the same product but from different producers, you must supply the producer name as well as the product name, for example:

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

If there are 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 3.20.

3.5.2 Loading the Licenses for All Products

Use the `lmf reset` command to copy the license details for all products from the LDB to the kernel cache. The `lmf reset` command has the following syntax (see also the `lmf(8)` reference page).

```
lmf reset [ cpus [ n ]]
```

In addition to copying the license details from the LDB to the kernel cache, `lmf reset cpus` checks the number of active CPUs and uses this number 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.

The *n* argument represents the number of active CPUs on the system when determining the SMM.

3.6 Unloading a License from the Kernel Cache

Use the lmf unload command to restrict the number of users of a product. You can do this by removing license units from the kernel cache, thus restricting the number of units available to the 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 LDB.

For example, suppose you have registered and loaded the license for the product ALLSUM with a 10-user Activity License. You could restrict the license to a 7-user license by typing:

```
# lmf unload 3 ALLSUM
```

Existing users of the product are allowed to finish using it before the new limit is imposed. For example, if there are 10 users of a product and the lmf unload command is used to restrict the number of users to 7, all 10 users will be able to finish using the product. However, new users of the product will not be allowed until the number of current users has dropped to less than the new limit of 7.

In the case of an availability licensed product, you must unload all the license units for the product. You do this by specifying zero (0) as the number of users. This indicates that all the license units for the product should be removed from the kernel cache. For example, if the product ALLSUM was availability licensed, and you wanted to unload the license units for the product, you would type:

```
# lmf unload 0 ALLSUM
```

As with activity licensed products, existing users of the product are allowed to finish using it, but new users are refused access.

If you do not want the license to be reloaded when the system is rebooted or when you issue the lmf reset command, you should disable the license with the lmf disable command.

3.7 Enabling a License

Use the lmf enable command to enable a license to be loaded into the kernel cache. If a license is disabled, it is ignored when you use lmf load, lmf reset or when you reboot the system.

For example, to enable the license for the product ALLSUM and load the license into the kernel cache, you type:

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

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

3.8 Disabling a License

Use the lmf disable command to prevent a license from being loaded into the kernel cache when you use the lmf load command, the lmf reset command, or reboot the system. For example, to disable the license for the product ALLSUM, you would type:

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

The `lmf disable` command does not immediately affect the kernel cache, so you should use the `lmf unload` command to unload the license details from the kernel cache. If you do not use the `lmf unload` command, the product remains available on the system until the next `lmf reset` command or system reboot. You cannot disable the ULTRIX license generated by the LMF from the `/upgrade` file.

3.9 Issuing a License

Use the `lmf issue` command to move the license details for a product from the LDB to a file on your system. The command reconstructs a Product Authorization Key from the license data in the LDB and outputs the PAK to a specified file. If the PAK is issued correctly, the license is deleted from the LDB and unloaded from the kernel cache. You can use this command to move a license for a product from one system to another (see Sections 3.16 and 3.18).

The `lmf issue` command has the following syntax (see also the `lmf`(8) reference page):

`lmf issue` *file product* [*producer* [*authorization*]]

For example, to issue the license for the product ALLSUM to the file `allsum.pak` type:

`# 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. You cannot issue the ULTRIX license generated by the LMF from the `/upgrade` file.

3.10 Cancelling a License

Use the `lmf cancel` command to cancel a license from a specific date. This means that you can stop the use of a product earlier than the day shown by the Key Termination Date field on the PAK. The `lmf cancel` command has the following syntax (see also the `lmf`(8) reference page):

`lmf cancel` *date product* [*producer* [*authorization*]]

The *date* argument can be specified in most common formats, but the order must be: day, month, year. You do not need to use a separator between the day and the month, or the month and the year. For example, 1st July 1990 could be specified as: 1-jul-1990, 1/7/90, 010790, or 1.july.90. To cancel the license for the product ALLSUM on 1st July 1990, you would type:

```
# lmf cancel 1-jul-90 ALLSUM
# lmf load 0 ALLSUM
```

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

You can change the Cancellation Date more than once; you just need to reissue the `lmf cancel` command with a different *date* argument. If you set the Cancellation Date to be after the Key Termination Date shown on the license, the Cancellation Date is ignored.

3.11 Deleting a License

Use the `lmf delete` command to delete a license from the LDB and kernel cache. You can use this command to remove a license from the LDB, when it no longer represents a valid license for the product; for example, if the license has passed its Key Termination Date.

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

To delete the license for the product ALLSUM, for example, you should type:

```
# 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.

3.12 Updating a License

There are two ways to update a license:

* Modify a license with the `lmf modify` command, if the license has the MOD_UNITS Key Option

* Amend the license with the `lmf 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 `lmf amend` command unless you have a PAAM.

3.12.1 Modifying a License

Use the `lmf modify` command if you want to change the entry in the Comments 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 100 to 200, type:

```
$ lmf modify ALLSUM
```

This displays the current license for ALLSUM with colons (:) before the Comments field and the Number of Units field, for example:

```
                  .
                  .
         Product Name  ALLSUM
              Producer  DEC

    Number of Units: 100
                  .
                  .
```

In this example you should use the editor to change the Number of Units from 100 to 200. Changes to lines without colons are ignored.

When you leave the editor, the LMF scans the template to ensure the license has been updated correctly. If it has not, you are given the opportunity to re-enter the editor and correct any mistakes.

When you have successfully modified a license, use the `lmf load` command to copy the modified license into the kernel cache. In this example, type:

```
# lmf load 0 ALLSUM
```

3.12.2 Amending a License

Use the `lmf amend` command when you want to update a license in the LDB after receiving a Product Authorization Amendment (PAAM). A PAAM is used to update an existing license and may only have data in fields that are different to the existing license for the product. A PAAM will always have a different checksum from 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.

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 V2.0. If you wanted to use the license with versions up to and including Version V2.4, you could contact your Digital representative who would arrange for you to be sent a PAAM. The PAAM would contain entries for Version and Checksum (and possibly Product Name to ensure you amend the correct license). To enter the PAAM data into the LDB type:

```
# lmf amend ALLSUM
```

This displays the current license for ALLSUM with colons (:) before the fields that can be changed, for example:

```
                     .
                     .
        Product Name   ALLSUM
            Producer   DEC

    Number of Units:   100

            Version:   2.0
Product Release Date:
                     .
                     .
```

The current license has the Checksum entry removed, because PAAMs always come supplied with a new checksum. In this example you should use the editor to change the Version from 2.0 to 2.4, and enter the checksum supplied with the PAAM. Changes to lines without colons are ignored.

When you leave the editor, the LMF scans the template to ensure the license has been updated correctly. If it has not, you are given the opportunity to re-enter the editor and correct any mistakes.

When you have successfully amended a license, use the `lmf load` command to copy the amended license into the kernel cache. In this example, type:

```
# lmf load 0 ALLSUM
```

3.13 Monitoring the LDB and Kernel Cache

Use the `lmf list` command to display the details of the registered products on the system. Using `lmf list`, you can:

- Display a summary of all the products registered in the LDB or kernel cache or both

- Display the complete license details for all the products in the LDB or kernel cache or both

- Display the details of specific products only

The `lmf list` command has the following syntax (see also the `lmf(8)` reference page):

```
lmf list [ full ] [ source ] [ for product [ producer ] ]
```

Use the *full* argument to display the full license details for the product.

Use the *source* argument to choose the source of the license information; there are three choices:

ldb   Displays a summary for each product in the LDB.

cache   Displays a summary for each product in the kernel cache. This shows you the license data that is being checked by the license checking functions.

all   Displays a summary for each product in the LDB, and for each product in the kernel cache.

The following example shows how to display a 1-line summary of all the products registered in LDB:

```
# lmf list
```

Product	Status	Users: Total	Active
MESSAGE-ROUTER	active	unlimited	
MR-PROGRAMMER-KIT	disabled		
WAN-DEVICE-DRIVERS	enabled		
FORTRAN-FOR-ULTRIX	terminated		
ALLSUM	active	unlimited	
ULTRIX	active	16	5

The Status column indicates the current status of the license. There are six possible license 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 LDB but has not been loaded into the kernel cache.

- `disabled`
  The license has been disabled in the LDB.

- `terminated`
  The current date is later than the Key Termination Date specified on the license.

- cancelled
  The current date is later than the Cancellation Date specified on the license.

- multiple
  This 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 of the product.

The following example shows how to display all the license details in the kernel cache for ULTRIX:

```
# lmf list full cache for ultrix

          Product Name: ULTRIX
              Producer: DEC
               Version:
  Product Release Date: 1-JUL-1991
  Key Termination Date:
           Total Units: 160
          Usable Units: 140
       Activity Charge: 10
```

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. For Availability Licensed products the Usable Units and Activity Charge fields are zero (0).

3.14 Reviewing Your License Management Activities

Use the lmf history to display a list of the lmf commands that have been used. The LMF maintains a history file that is a record of the license management operations. The commands recorded in the history file are: register, enable, disable, issue, cancel, delete, modify, and amend. The creation of a new LDB is also recorded in the history file. For these lmf commands, you can:

- Display the history data, which comprises product name, date and time of the command, and the fields that were changed on the license

- Display the history data and the license (as it appeared before the command was issued)

- Display a 1-line summary of the history data for each command issued

- Display the history data for commands issued after a certain date

- Display the history data for specific products

The lmf history command has the following syntax (see also the lmf(8) reference page):

lmf history [*length*] [from *date*] [for *product* [*producer*]]

Use the *length* argument to choose the length of the history data for each command. There are two choices:

short   Displays a 1-line summary of the history data for each command issued.

full   Displays the history data for each command issued, and the license as it appeared before the command was issued.

Use the `from` *date* argument to display the data for each command issued after the *date* specified. The *date* argument can be specified in most common formats, but the order must be: day, month, year. You do not need to use a separator between the day and the month, or the month and the year. For example, 1st March 1989 could be specified as: 1-mar-1989, 1/3/89, 010389, or 1.march.89.

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

```
# lmf history for ALLSUM

Product Name: ALLSUM
Producer    : DEC
Command     : ENABLE
Date        : 26-FEB-1989
Time        : 12:02:32

Product Name: ALLSUM
Producer    : DEC
Command     : DISABLE
Date        : 15-JAN-1989
Time        : 11:57:26

Product Name: ALLSUM
Producer    : DEC
Command     : REGISTER
Date        : 4-NOV-1988
Time        : 11:54:15
```

The next example shows the 1-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-1989 12:02:32
ALLSUM           DEC             DISABLE    15-JAN-1989 11:57:26
ALLSUM           DEC             REGISTER   4-NOV-1989  11:54:15
```

3.15 Changing the Number of Active CPUs

When a system is rebooted, the 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

3.15.1 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. The 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.

3.15.2 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, you must 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.

3.16 Managing Availability Licensed Products

Before you order a PAK, you should define your software and hardware requirements to your Digital representative so that you get a license of the correct size. For availability licensed products, the license you register in the License Database should provide enough license units to allow full access to the licensed product. For example, the PAK for a software product to be used on a processor which requires 400 license units should specify 400 license units.

Sometimes, users with multiple stand-alone systems cannot match their licenses to meet every circumstance. For example, you may manage two stand-alone processors; VAXBIG, which requires a 700-unit license, and VAXMID, which requires a 400-unit license. If you get a 700-unit license for VAXBIG, you can move that license (with an lmf issue command) to VAXMID when you shut down VAXBIG for a memory upgrade. You may not, however, be able to move and register a license intended for VAXMID to VAXBIG.

3.17 Providing More Availability License Units

Sometimes you may need to provide more license units than are currently registered in the LDB for the product. In the previous example, the 400-unit license did not provide enough product availability for VAXBIG, which required 700 units. If the license has the MOD_UNITS Key Option, and you need to move the license to VAXBIG, increase the number of units on the license to 700 using the lmf modify command. If the license does not have the MOD_UNITS Key Option, and you need to move the license to VAXBIG, contact a Digital representative, who will probably recommend one of the following:

- A new license that provides at least 700 license units.

- Another license for the same product that provides at least an additional 300 license units. If the terms of your license contract allow it, you can register the two licenses, allowing the LMF to combine the license units and producing the equivalent of a 700-unit license. This can authorize the product on VAXBIG. For a complete explanation of license combination, see Section 3.20.

- A Product Authorization Amendment (PAAM) that increases your current license to at least 700 units.

3.18 Managing Activity Licensed Products

As with availability licensed products, you should define your software and hardware requirements to your Digital representative so that you get a license of the correct size. The Activity License you register in the LDB should provide enough license units to allow some predetermined number of processes access to the product.

For example, if a software product requires 25 license units per activity on your processor, and PAKs come in 4-activity increments, your license may provide units in a multiple of 100: 100, 200, 400, and so forth. Note that a 120-unit license would provide no more use than a 100-unit license on such a processor.

Because processors can have different license unit requirements per activity, the number of users authorized by a license can vary according to the processor used. For example, you may manage two stand-alone processors; VAXBIG which requires 25 license units per activity to authorize a product, and VAXMID which requires only 20 license units per activity to authorize a product.

If you obtain a 125-unit Activity License for VAXBIG, you can temporarily move that license (using the lmf issue command) to VAXMID when you shut down VAXBIG for maintenance. The 125-unit license, which provided five users product access on VAXBIG, provides six users product access on VAXMID. This provides the backup you need. Also, unlike the situation with Availability Licenses, you can move a 4-user 80-unit license originally for VAXMID to VAXBIG. On VAXBIG the license provides access to only three users, however.

As with Availability Licenses, you can successfully register a license in the License Database that a user cannot successfully activate. If you register a 40-unit license that provides product access to two users on a MicroVAX II, the same license may not allow access to one user on a VAX 8800 that might require 50 units per access.

3.19 Providing More Activity License Units

If the license has the MOD_UNITS Key Option, you can increase the number of units on the license using the lmf modify command. If the license does not have the MOD_UNITS Key Option, contact a Digital representative, who will probably recommend one of the following:

- A new license with more units.

- Another license for the same product that provides 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 3.20.

- A different kind of license. Because some products offer both Activity and Availability Licenses, a change to an Availability License may be recommended.

- A Product Authorization Amendment (PAAM) that increases the size of your current license.

3.20 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 which have the NO_SHARE Key Option cannot be combined.

The following fields must be the same on the original licenses:

- Issuer

- Product Name

- Producer

- Hardware-Id

- Product Token

Note

The Authorization Number must be different on the licenses; the LMF does not allow the same license to be registered more than once in the same LDB.

Register the licenses in the LDB in the usual way; that is, using the lmf register command (see Section 3.2). The licenses registered appear as separate entries in the LDB. To combine licenses to form a single license in the kernel cache, use the lmf load command (see Section 3.5.1). For example, if you have registered two licenses for the product ALLSUM, you can form a single license in the kernel cache by typing:

```
# lmf load 0 ALLSUM
```

The combined license appears as one entry in the kernel cache.

The Number of Units for the license is the total number supplied by the original licenses. For example, if the Number of Units entries on the original licenses were 300 and 400, the combined license would be a 700-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 were 1-JAN-1990 and 1-AUG-1990, the combined license would have a Product Release Date of 1-JAN-1990.

The Version for the combined license is the lower of those supplied by the original licenses. For example, if the Version for the original licenses were V1.2 and V1.4, the combined license would have a Version of V1.2.

To remove the combined license from the kernel cache, use the `lmf unload` command (see Section 3.6).

This Appendix lists and explains the error messages you may encounter. The error messages are listed in alphabetical order, and where appropriate a course of action is recommended to correct the error.

A.1 Accessing Licensed Software

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

Attempted usage exceeds active license units

You have 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 means that the maximum number of simultaneous users of the product has been reached.

Contact your system manager to find out 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 have tried to access a product but the version number on the license is lower than the product version.

Contact your system manager. Your system manager may need to install an earlier version of the product or contact Digital for a new PAK.

No license found for this product

You have 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(8) Utility

You may encounter the following messages when using the lmf utility.

A license for *product* cannot be disabled

You have tried to disable a license for the *product* specified. Certain licenses cannot be disabled, for example, the license created by the LMF from information in the /upgrade file, see Section 3.3.

A license for *product* cannot be issued

You have tried to issue a license for the *product* specified. Certain licenses cannot be issued, for example, the license created by the LMF from information in the /upgrade file, see Section 3.3.

A license that has been cancelled cannot be enabled

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

If you want to enable the license, you should change the cancellation date, see Section 3.10.

A license that has terminated cannot be enabled

You have tried to enable a license that has passed its termination date. A license that has passed its termination date should be deleted from the LDB, see Section 3.11.

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

The activity charge has changed for the *product* specified. This may have happened if the license type has changed, for example, from an activity to availability license, or if the SMM has changed.

Reboot your system to fully reset the kernel cache.

"Activity Table Code" amended - protected field

You have changed the Activity Table Code. The Activity Table Code can only be changed when using lmf amend with a suitable Product Authorization Amendment.

"Activity Table Code" - invalid format

The Activity Table Code you have 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 PAK.

"Activity Table Code" missing from PAK entry

You have not entered an Activity Table Code for the license.

When you register the license, you must enter all the data from the PAK.

Ambiguous command *string*

The *string* used as an abbreviation for a command was ambiguous.

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

"Authorization Number" amended - protected field

You have changed the Authorization Number. The Authorization Number should never be changed, as it helps to uniquely identify each license.

"Authorization Number" missing from PAK entry

You have not entered an Authorization Number for the license.

When you register the license, you must enter all the data from the PAK.

"Availability Table Code" amended - protected field

You have changed the Availability Table Code. The Availability Table Code can only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

"Availability Table Code" - invalid format

The Availability Table Code you have 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 PAK.

"Availability Table Code" missing from PAK entry

You have not entered an Availability Table Code for the license.

When you register the license, you must enter all the data from the PAK.

Cannot unload this many users

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

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

"Checksum" amended - protected field

You have changed the Checksum. The Checksum can be changed only when using `lmf amend` with a suitable Product Authorization Amendment.

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 PAK. If you enter inaccurate license information, you receive this message.

Carefully review all licensing 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 have not entered a Checksum for the license.

When you register the license, you must enter all the data from the 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 has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error closing file *filename*

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

Error closing license database *filename*

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

Error closing LURT file *filename*

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

Error closing temporary file *filename*

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

Error closing the history file *filename*

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

Error creating license database *filename*

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

Error creating the history file *filename*

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

Error determining SMM

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

Error locking license database *filename*

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

Error locking the history file *filename*

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

Error no entry in LURT for this SMM

The LURT version and the SMM version are inconsistent.

Contact your Digital representative.

Error opening file *filename*

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

Error opening license database *filename*

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

Error opening LURT file *filename*

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

Error opening temporary file *filename*

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

Error opening the history file *filename*

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

Error reading kernel cache

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

Error reading license database *filename*

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

Error reading LURT file *filename*

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

Error reading the history file *filename*

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

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

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

Error setting the number of cpus

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

Error unlocking license database *filename*

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

Error unlocking the history file *filename*

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

Error updating kernel cache

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

Error writing to license database *filename*

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

Error writing to the history file *filename*

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

Error writing to temporary file *filename*

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

Failed to create process for editor

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

"Hardware-Id" amended - protected field

You have changed the Hardware-Id. The Hardware-Id should only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

History file locked - retrying ...

You have tried to access the history file at the same time as another user. The lmf utility automatically grants 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 one license for the product.

If you have more than one license 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 has 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 LURT version and the LMF utility are inconsistent.

Contact your Digital representative.

Invalid LURT entry for *product producer*

The LURT version and the PAK information are inconsistent.

Contact your Digital representative.

"Issuer" amended - protected field

You have changed the Issuer. The Issuer can only be changed when using lmf amend with a suitable Product Authorization Amendment.

"Issuer" missing from PAK entry

You have not entered an Issuer for the license.

When you register the license, you must enter all the data from the PAK.

"Key Options" amended - protected field

You have changed the Key Options. The Key Options can only be changed when using lmf amend with a suitable Product Authorization Amendment.

"Key Termination Date" amended - protected field

You have changed the Key Termination Date. The Key Termination Date can only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

License already registered

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

License database locked - retrying ...

You have tried to access the LDB at the same time as another user. The `lmf` utility automatically grants you access to the LDB as soon as the other user has finished with it.

License too small to load this many users

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

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

License unchanged

You have left the editor after a `lmf amend` or `lmf modify` command, without making any changes to the existing license.

Missing arguments

You have not specified 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 have tried to combine licenses, at least one of which has the NO_SHARE Key Option. The LMF will not combine licenses if any of the following fields are different: Issuer, Product Name, Producer, Product Token, Hardware-Id. Only licenses without the NO_SHARE Key Option can be combined.

No entries in license database

You have tried to list the contents of the LDB, but the LDB is empty.

No entry in the history file for this product

You have specified a product name with the `lmf history` command, but no `lmf` commands have been 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, it should be specified exactly as it appears on the PAK. If you use the `from` *date* argument, ensure 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 have specified in the command does not have an entry in the kernel cache.

When you use the *product* argument, it should be specified exactly as it appears on the PAK.

No entry in the license database for this product

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

When you use the *product* argument, it should be specified exactly as it appears on the PAK.

No valid license was found for this product

You have tried to load a nonvalid license for a product.

Ensure that you entered the correct product name, and that the license is not terminated, disabled, or cancelled.

Not enough memory

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

Not enough units to load *product producer*

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

For a full description of how to provide more license units, see Section 3.17 and Section 3.19.

"Number of Units" amended - protected field

You have changed the Number of Units. The Number of Units can be changed either when using the lmf modify command on a license which has the MOD_UNITS Key Option, or when using lmf amend with a suitable Product Authorization Amendment.

PAK not registered

You have left the editor after a lmf register command without saving the file.

Permission denied

You have tried to execute a lmf command but are not logged into the system as the superuser (root login).

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

"Producer" amended - protected field

You have changed the Producer. The Producer should never be changed, as it helps to uniquely identify each license.

"Product Name" amended - protected field

You have changed the Product Name. The Product Name should never be changed, as it helps to uniquely identify each license.

"Product Name" missing from pak entry

You have not entered a Product Name for the license.

When you register the license, you must enter all the data from the PAK.

"Product Release Date" amended - protected field

You have changed the Product Release Date. The Product Release Date can only be changed when using lmf amend with a suitable Product Authorization Amendment.

"Product Token" amended - protected field

You have changed the Product Token. The Product Token can only be changed when using lmf amend with a suitable Product Authorization Amendment.

The kernel cache is empty

You have 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 LDB 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 LDB version and the lmf utility are inconsistent.

Contact your Digital representative.

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

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

Unrecognized cpu for *product producer*

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

"Version" amended - protected field

You have changed the Version. The Version can be changed only when using lmf amend with a suitable Product Authorization Amendment.

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 LDB and is creating a new one.

Glossary

This glossary defines a number of terms and acronyms that may be encountered.

activity license

A license which defines the number of concurrent users allowed access to a product. For example, a 4-activity license can have enough license units to allow four 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. The LMF makes a product accessible when the number of license units on a license matches or exceeds the license unit rating of the current processor.

CDROM

See the entry for compact-disc read-only memory (CDROM).

checksum

An encoded number calculated from the other information supplied with a PAK. The checksum is used by the LMF 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 sixteen positions are always alphabetic characters from A to P.

compact-disc read-only memory (CDROM)

A media for the consolidated distribution of software. ULTRIX operating systems and layered products can be distributed on single CDROMs, with software access authorized by PAKs and the LMF.

DDSLA

See the entry for Digital Distributed Software Licensing Architecture (DDSLA).

Digital Distributed Software Licensing Architecture (DDSLA).

This engineering architecture is based on the concept of license units as an abstract mechanism for counting and sizing customer computing environments.

key termination date

A field on a PAK that defines when a license contract is no longer valid, that is, when the LMF no longer authorizes product use.

LDB

See the entry for License Database (LDB).

license amendment

Updating an existing license by entering data from a Product Authorization Amendment (PAAM) in the License Database.

license combination

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 option cannot be combined.

License Database (LDB)

A system file which contains the licenses registered on the system. The LMF also maintains a kernel cache with the license information in it, and this is used by the LMF to prevent unlicensed product use.

License Management Facility (LMF)

Part of ULTRIX operating systems that enables the on-line management of software license data, and also helps prevent accidental unlicensed use of software.

license registration

The task you perform when you enter license data from a Product Authorization Key into the License Database. To register a license, use the `lmf register` command.

license unit

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

License Unit Requirement Tables (LURTs)

A table provided by Digital as part of ULTRIX operating systems that specifies a series of license unit requirements, essentially performance ratings, for each System Marketing Model. Processors that provide more performance (other ratings may be unrelated to performance) have greater license units requirements.

LMF

See the entry for License Management Facility (LMF).

LURT

See the entry for License Unit Requirement Tables (LURTs).

PAAM

See the entry for Product Authorization Amendment (PAAM)

PAK

See the entry for Product Authorization Key (PAK).

PAK identification

The Product Authorization Key issuer name and the authorization number. Together, they 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. PAK issuers are usually the same as software producers but can operate under agreement with the producer.

Product Authorization Amendment (PAAM)

Provides information to amend the license for an existing licensed software product. Without a current PAK or the appropriate PAAM, you may not be able to use an installed software product. A PAAM contains a unique authorization checksum and the information needed to amend current license information.

Product Authorization Key (PAK)

A list of essential information about a software license that must be registered in the License Database in order to use a product. It is produced by a PAK issuer and delivered to you by mail, electronic transfer, or by telephone.

Product Identification

The software producer name and product name. Together they uniquely identify a software product for licensing.

SMM

See the entry for System Marketing Model (SMM).

software license

A contract between a license producer (Digital) 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.

SPD

See the entry for 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 product version and official product release date.

System Marketing Model (SMM)

The model name of a computer system, as used in marketing and pricing. The SMM generally corresponds to the name on the front panel of the processor cabinet. The LMF uses this value rather than hardware CPU-type because different marketing models may use the same CPU with different pricing and licensing rules.

Index

A

Activity License, 1–2
  accessing the licensed software, 2–9
  Activity Table Code on a PAK, 2–5
  loading a license into the kernel cache, 2–9
  managing Activity Licenses, 3–18
  providing more license units, 3–19
Activity Table Code
  field on a PAK, 2–5
  use in license checking, 2–9
Authorization Number, 2–4
Availability License, 1–2
  accessing the licensed software, 2–8
  Availability Table Code on a PAK, 2–5
  loading a license into the kernel cache, 2–8
  managing Availability Licenses, 3–17
  providing more license units, 3–18
Availability Table Code
  field on a PAK, 2–5
  use in license checking, 2–8

C

Cancellation Date
  lmf cancel command, 3–11
  use in license checking, 2–7
Capacity License
  See Availability License
Checksum
  field on a PAK, 2–6
combining licenses, 3–19

E

error messages
  system manager, 2–8, A–1
  user, 2–9, 2–10, A–1

H

Hardware-Id
  field on a PAK, 2–6
history file location, 3–1

I

Issuer
  field on a PAK, 2–4

K

kernel cache, 2–7
  loading activity license units, 2–9
  loading availability license units, 2–8
Key Termination Date
  field on a PAK, 2–5
  use in license checking, 2–7

L

layered products
  LURT code, 2–6
  registering a license, 3–7
LDB
  See License Database
license checking
  accessing the licensed software, 2–8, 2–9
  for activity licensed products, 2–9

P

PAAM

See Product Authorization Amendment

PAK

See Product Authorization Key

Per-user License

See Activity License

P_FAMILY

Key Option field on a PAK, 2–6

Producer

field on a PAK, 2–4

use in license checking, 2–7

Product Authorization Amendment, 2–3

Product Authorization Key

definition of a PAK, 1–2

example PAK, 2–3, 3–3

fields on a PAK, 2–3 to 2–6

getting a PAK, 2–2

License PAK, 2–2

ordering enough license units, 2–6

Service Update PAK, 2–2

Temporary Service PAK, 2–2

Product Name

field on a PAK, 2–4

use in license checking, 2–7

Product Release Date

field on a PAK, 2–5

use in license checking, 2–7

Product Token

field on a PAK, 2–6

R

reboot

as part of installation, 3–6, 3–7

executing lmf reset, 3–8

to load license details into the kernel cache, 2–7, 3–6

use with lmf disable, 3–11

use with lmf enable, 3–10

registering a license

choosing an editor, 3–2

definition, 1–2

registering a license (cont.)

editing an empty template, 3–2

editing an existing template, 3–4

fields requiring an entry, 3–2

for existing operating systems, 3–7

for layered products, 3–7

for new operating systems, 3–6

registering a PAK directly, 3–5

S

Service Update PAK

See Product Authorization Key

SMM

See System Marketing Model

superuser

lmf command privileges, 3–1

System Marketing Model

relationship to LURTs, 2–6

use in license checking, 2–8, 2–9

T

Temporary Service PAK, 2–2

TSP

See Temporary Service PAK

V

Version

field on a PAK, 2–4

use in license checking, 2–7

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call
800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-
baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need
assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal*	————	SSB Order Processing - WMO/E15 or Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

_____

_____

What do you like best about this manual? _____

_____

_____

What do you like least about this manual? _____

_____

_____

Please list errors you have found in this manual:

Page   Description

_____   _____

_____   _____

_____   _____

_____   _____

_____   _____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

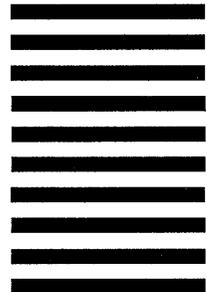Mailing Address _____

_____ Email _____ Phone _____

digital ™

BUSINESS REPLY MAIL
FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Reader's Comments

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

_____

_____

What do you like best about this manual? _____

_____

_____

What do you like least about this manual? _____

_____

_____

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

What version of the software described by this manual are you using? _____
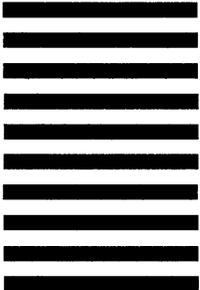
Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

d i g i t a l TM

BUSINESS REPLY MAIL
FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062–9987