

---

# HP OpenVMS License Management Utility Manual

Order Number: BA322-90094

**August 2010**

This manual describes the use of the License Management Facility (LMF), the software license management tool for the OpenVMS operating system.

**Revision/Update Information:** This is a revised manual. This manual supersedes *HP OpenVMS License Management Utility Manual*, OpenVMS Version 8.3-1H1 for Integrity servers and OpenVMS Alpha Version 8.3.

**Software Version:** OpenVMS Version 8.4 for Integrity servers  
OpenVMS Alpha Version 8.4

**Hewlett-Packard Company**  
Palo Alto, California

---

© Copyright 2010 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Printed in the U.S.

ZK4584

The HP OpenVMS documentation set is available on CD-ROM.

This document was prepared using DECdocument, Version 3.3-1b.

---

# Contents

<b>Preface</b> .....	vii
<b>1 Overview</b>	
1.1 License Management .....	1-1
1.2 License Management Utility (LICENSE) .....	1-2
1.3 License Database .....	1-2
1.3.1 History Records .....	1-3
1.4 Licensing Differences on Alpha/VAX and Integrity servers .....	1-4
<b>2 Licensing on OpenVMS Alpha and OpenVMS VAX</b>	
2.1 License Units and License Unit Requirement Tables .....	2-1
2.2 Types of Licenses .....	2-4
2.2.1 Availability Licenses .....	2-4
2.2.1.1 Providing Sufficient License Units .....	2-5
2.2.1.2 Providing More License Units .....	2-5
2.2.1.3 Providing Availability in an OpenVMS Cluster Environment .....	2-6
2.2.1.4 Providing More Availability .....	2-7
2.2.2 Activity License .....	2-7
2.2.2.1 Providing Enough License Units .....	2-8
2.2.2.2 Determining Your License Needs .....	2-9
2.2.2.3 Sharing License Units in an OpenVMS Cluster Environment .....	2-9
2.2.3 Personal Use License .....	2-9
2.2.4 User License .....	2-10
2.2.5 Group License .....	2-10
2.3 License Combination .....	2-11
2.3.1 Licenses That can be Combined .....	2-11
2.3.2 Include, Exclude, and Reservation Lists .....	2-12
2.3.3 Termination Dates and Version Numbers .....	2-12
<b>3 Licensing on OpenVMS Integrity servers</b>	
3.1 Operating Environment and Tiering .....	3-1
3.2 License Types .....	3-3
3.2.1 Per Core Licenses .....	3-3
3.2.1.1 Per Socket Licenses .....	3-4
3.2.2 Activity Licenses .....	3-4
3.3 Compliance Reporting .....	3-5
3.3.1 Hardware Compliance .....	3-5
3.3.2 Soft Compliance .....	3-5
3.3.3 Sample License Checks .....	3-6

## 4 Licensing OpenVMS Guests on Integrity VM

4.1	License Distribution . . . . .	4-1
4.2	Standalone System Licensing Requirements . . . . .	4-1
4.3	Cluster Licensing Requirements . . . . .	4-1
4.3.1	New License Modify Qualifier <code>/[NO]VIRTUAL</code> Required for OpenVMS Guest Systems . . . . .	4-2
4.3.2	Virtual License Loading . . . . .	4-2
4.3.3	Virtual License Recommendations . . . . .	4-2
4.4	Compliance Checking and Reporting . . . . .	4-2

## 5 Using LMF

5.1	Preparing for License Registration . . . . .	5-1
5.2	Managing the License Database . . . . .	5-2
5.2.1	Database Location . . . . .	5-2
5.2.2	History Records . . . . .	5-3
5.3	Getting a Product Authorization Key (PAK) . . . . .	5-3
5.4	Registering Licenses . . . . .	5-5
5.4.1	When To Perform Registration . . . . .	5-5
5.4.2	Registration and Installation . . . . .	5-6
5.4.3	Registration Methods . . . . .	5-7
5.4.4	Using <code>VMSLICENSE.COM</code> . . . . .	5-7
5.4.4.1	Using Data Files with <code>VMSLICENSE.COM</code> . . . . .	5-11
5.4.4.2	Invoking <code>VMSLICENSE.COM</code> with a <code>VMSLICENSE</code> Data File . . . . .	5-11
5.4.4.3	Creating <code>VMSLICENSE</code> Data Files . . . . .	5-12
5.4.4.4	Using <code>VMSLICENSE.COM</code> Default Value Rules . . . . .	5-13
5.4.5	Using the <code>LICENSE REGISTER</code> Command . . . . .	5-14
5.4.6	Displaying License Information . . . . .	5-15
5.5	Loading a License . . . . .	5-15
5.5.1	Loading Licenses in an OpenVMS Cluster Environment . . . . .	5-15
5.6	Managing Licenses After Registration . . . . .	5-17
5.6.1	Restricting Product Use . . . . .	5-18
5.6.1.1	Fastest Method . . . . .	5-18
5.6.1.2	For Inactive Licenses . . . . .	5-18
5.6.1.3	Permanent Methods . . . . .	5-19
5.6.2	Controlling Node Access to Licenses in Clusters . . . . .	5-19
5.6.2.1	Effect of the <code>NO_SHARE</code> Option . . . . .	5-20
5.6.2.2	Editing Include and Exclude Lists . . . . .	5-21
5.6.3	Controlling User Access . . . . .	5-21
5.6.4	Controlling Loading Order . . . . .	5-23
5.7	Logical Name <code>LMF\$DISPLAY_OPCOM_MESSAGE</code> . . . . .	5-23
5.8	Troubleshooting Licensing Problems . . . . .	5-24

## 6 OpenVMS Galaxy Licensing Information (Alpha Only)

6.1	OpenVMS Galaxy Licensing Requirements . . . . .	6-1
6.1.1	OpenVMS Operating System License . . . . .	6-1
6.1.2	OpenVMS Galaxy License . . . . .	6-1
6.1.3	OpenVMS Layered Products License . . . . .	6-2
6.2	Clustering OpenVMS Galaxy Instances . . . . .	6-2
6.2.1	Clustering in a Galaxy System . . . . .	6-2
6.2.2	Clustering Outside a Galaxy System . . . . .	6-2
6.3	License Databases . . . . .	6-2

6.4	OpenVMS Galaxy License PAKs and LMF .....	6-3
-----	---	-----

## A Command Reference

LICENSE COPY .....	A-2
LICENSE CREATE .....	A-5
LICENSE DELETE .....	A-6
LICENSE DISABLE .....	A-9
LICENSE ENABLE .....	A-11
LICENSE ISSUE .....	A-13
LICENSE LIST .....	A-16
LICENSE LOAD .....	A-22
LICENSE MODIFY .....	A-25
LICENSE MOVE .....	A-32
LICENSE REGISTER .....	A-34
LICENSE START .....	A-38
LICENSE UNLOAD .....	A-39
SHOW LICENSE .....	A-41

## B Examples

B.1	Registering a System Integrated Product .....	B-1
B.2	Difference Between LICENSE LIST and SHOW LICENSE .....	B-5
B.3	Restricting Product Use .....	B-7

## Glossary

## Index

## Examples

5-1	Typical PAK Information .....	5-6
5-2	Sample VMSLICENSE Data File .....	5-12
5-3	Displaying License Database Information .....	5-14

## Figures

1-1	The Licensing Model .....	1-3
5-1	PAK Transfer Methods .....	5-4
5-2	From a PAK to the License Database .....	5-5
5-3	The PAK and Software Routes to a License .....	5-7
5-4	Loading a License .....	5-15
5-5	Loading a License in an OpenVMS Cluster Environment .....	5-17

## Tables

2-1	License Codes . . . . .	2-2
2-2	Sample License Unit Requirements for Alpha Systems . . . . .	2-3
2-3	Types of Licenses . . . . .	2-4
5-1	Parameters Used with VMSLICENSE.COM . . . . .	5-13
A-1	LICENSE REGISTER /OPTIONS Keywords . . . . .	A-35

---

# Preface

The License Management Facility (LMF) is the software license management tool for the OpenVMS operating system. To run any software product on OpenVMS systems, you must register and load its license. To perform these tasks, use LMF.

## Intended Audience

This manual is for managers of licenses for software products that run on the OpenVMS operating system. Typically, the system manager has this responsibility.

## Document Structure

This manual consists of the following parts:

- Chapter 1 provides an introduction to the OpenVMS LMF and to the licensing of OpenVMS layered products.
- Chapter 2 describes licensing requirements for OpenVMS Alpha and VAX systems.
- Chapter 3 describes licensing requirements for OpenVMS Integrity server systems.
- Chapter 4 describes licensing requirements for OpenVMS Guests.
- Chapter 5 describes each task required to manage software product licenses. This chapter also discusses LICENSE commands and shows how to use them.
- Chapter 6 describes licensing for OpenVMS Galaxy systems.
- Appendix A describes the syntax of the LICENSE commands.
- Appendix B provides examples of license-management tasks.
- The Glossary defines the LMF-related terms used in this manual.

## Related Documents

The following manuals contain information related to the License Management utility:

- *HP OpenVMS System Manager's Manual*
- *HP OpenVMS System Management Utilities Reference Manual*
- *OpenVMS Record Management Utilities Reference Manual*
- *HP OpenVMS DCL Dictionary*

For information about installing software, see the following documentation:

- Upgrade and installation manual for your version of OpenVMS software

- The installation guides, release notes, and Software Product Descriptions (SPDs) for any software products you install

For additional information about HP OpenVMS products and services, see the OpenVMS website:

<http://www.hp.com/go/openvms>

## Reader's Comments

HP welcomes your comments on this manual. Please send comments to:

[openvmsdoc@hp.com](mailto:openvmsdoc@hp.com)

## How to Order Additional Documentation

For information about how to order additional documentation, see the OpenVMS Documentation Ordering website:

<http://www.hp.com/go/openvms/doc/order>

## Conventions

The following typographic conventions pertain to in this manual:

Ctrl/ <i>x</i>	A sequence such as Ctrl/ <i>x</i> indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 <i>x</i>	A sequence such as PF1 <i>x</i> indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
<span style="border: 1px solid black; padding: 2px;">Return</span>	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
...	A horizontal ellipsis in examples indicates one of the following possibilities: <ul style="list-style-type: none"> <li>• Additional optional arguments in a statement have been omitted.</li> <li>• The preceding item or items can be repeated one or more times.</li> <li>• Additional parameters, values, or other information can be entered.</li> </ul>
.	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify more than one.
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.



	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
<b>bold type</b>	Bold type represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason.
<i>italic type</i>	Italic type indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error <i>number</i> ), in command lines ( <i>PRODUCER=name</i> ), and in command parameters in text (where <i>dd</i> represents the predefined code for the device type).
Example	This typeface indicates code examples, command examples, and interactive screen displays. In text, this type also identifies URLs, UNIX commands and pathnames, PC-based commands and folders, and certain elements of the C programming language.
UPPERCASE TYPE	Uppercase type indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
-	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radices—binary, octal, or hexadecimal—are explicitly indicated.



---

# Overview

This overview introduces the OpenVMS License Management Facility (LMF) and outlines the tasks required to manage software licenses for software products.

A product license protects the intellectual property of the software vendor and provides customers with access to the product. Product authorization is usually defined in a contract with specific terms and conditions agreed upon by the software license issuer and the software user.

HP and other software vendors provide software to their customers under an agreement called a **license**. In this document the term refers to the authorization you have to run a software product on the OpenVMS operating system.

---

## License Management Facility and License Agreements

---

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

LMF is a management tool that can help you comply with your license agreement, but use of LMF does not indemnify you against noncompliance with the terms and conditions of your software license agreements. That is, LMF offers options for many kinds of license agreements, but use of some of these options might not authorize by your specific license agreement. Read your license carefully to determine which LMF options you can use legally.

---

## 1.1 License Management

To use a software product that requires a license, you must perform the following steps:

- Obtain a **Product Authorization Key (PAK)**, which provides information required to register the license.
- Use LMF to register the license in the **License Database**.
- Use LMF to load the license from the License Database into memory.
- Install the product specified by the PAK.

LMF provides additional features to modify licenses to satisfy specific requirements of individual sites.

To manage software product licenses for OpenVMS layered software, you need to understand the following information about licenses and the tool to manage them on OpenVMS systems:

- License Management utility (LICENSE) (Section 1.2)
- License Database (Section 1.3)

## Overview

### 1.2 License Management Utility (LICENSE)

### 1.2 License Management Utility (LICENSE)

The License Management utility (LICENSE) is the command line interface of the License Management Facility (LMF). Use LICENSE commands to interactively manage the licenses of OpenVMS layered software products and, in many cases, by third-party vendors.

LICENSE is a system-level tool that you use at the DCL prompt. LICENSE commands allow you to register licenses, load them into the License Database, and manage the licenses on your system.

For information on using LICENSE commands, see Chapter 5.

For reference information on commands, qualifiers, and examples, see Appendix A.

### 1.3 License Database

The License Database is a collection of information stored in a file called the License Database on a disk that contains information about each license on your system. The default database file is SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB, which is created by LMF when you install the OpenVMS software.

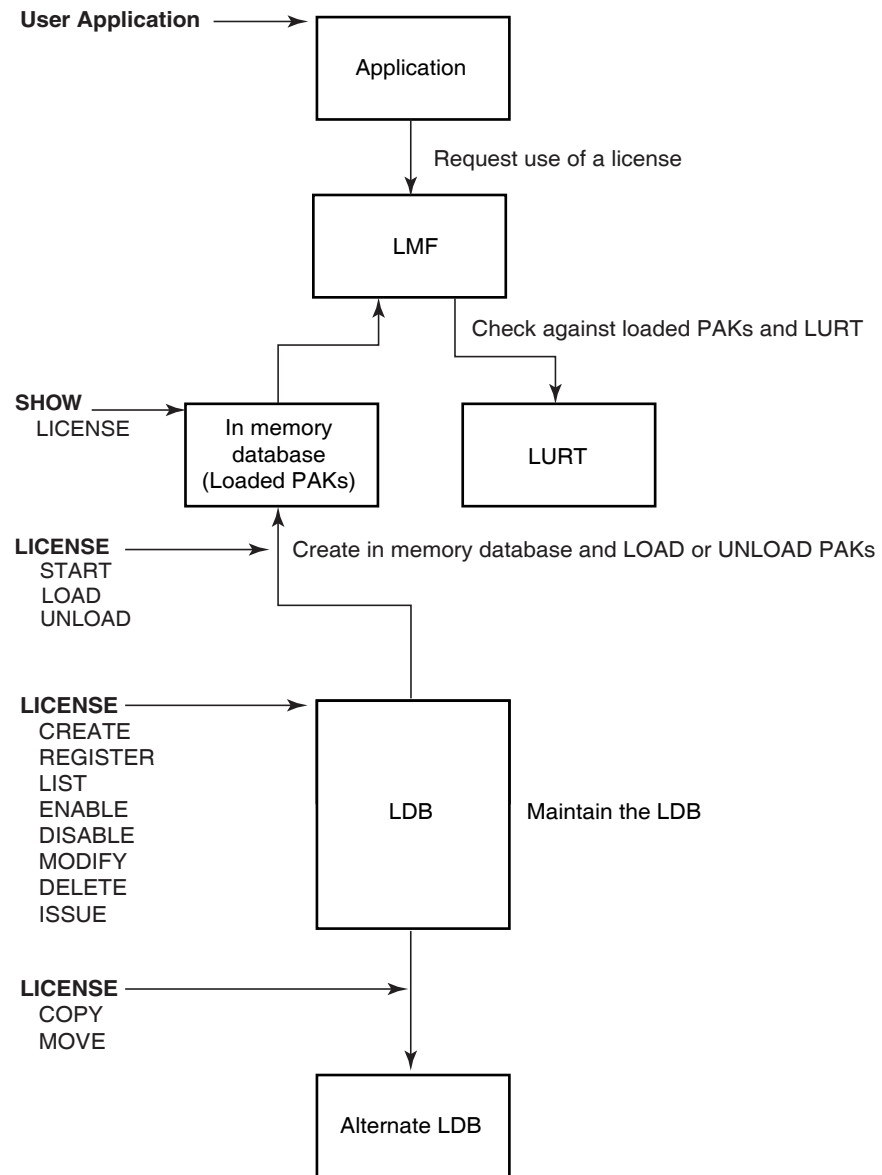
The terms of each license are stored in a collection of data fields in the database. You enter data by:

- Issuing LICENSE commands
- Executing the VMSLICENSE.COM command procedure

In addition, LMF enters data and keeps records. The collection of data fields representing a license at any one time is called a **record**. When you first register a license, you create the first record with data specified in your PAK. If you later modify the license, LMF creates a new record to define the modified terms of the license, and includes a notation that the license was modified.

Figure 1–1 illustrates the relationship between LMF, the license units required for your system, and the License Database.

Figure 1–1 The Licensing Model



VM-0777A-AI

### 1.3.1 History Records

LMF keeps track of the licensing activity on your system by writing a history record to the License Database every time you modify a PAK. Each history record contains an exact copy of the following:

- License record before modification
- LICENSE command you used to modify the record
- Date and time that you made the changes

## Overview

### 1.3 License Database

The history record also logs the username of the person who made the changes to the PAK. For information about viewing and purging these records, see Section 5.2.2.

### 1.4 Licensing Differences on Alpha/VAX and Integrity servers

While the basic LMF management functions remain the same, new licensing practices for OpenVMS Integrity servers differ from what is available for OpenVMS Alpha and OpenVMS VAX. The primary differences involve:

- The way license units are assigned and counted
- The types of licenses offered

See Chapter 2 for Alpha and VAX licensing.

See Chapter 3 for Integrity server systems licensing.

See Chapter 4 for OpenVMS guest licensing.

---

# Licensing on OpenVMS Alpha and OpenVMS VAX

This chapter describes licensing on OpenVMS Alpha and OpenVMS VAX systems.

On OpenVMS Alpha and VAX systems, the number of license units required are based on the rating of the system. LMF gets information about the units requirements from the console firmware. This information is contained in a License Unit Requirement Table (LURT).

Several different types of licenses are available on OpenVMS Alpha and VAX systems. In a cluster, these can be combined.

- Section 2.1 describes how license unit requirements are computed.
- Section 2.2 describes the types of licenses available.
- Section 2.3 describes how licenses can be combined.

## 2.1 License Units and License Unit Requirement Tables

A **license unit** is a measurement of the authorization granted for use of a product. License units define the size of each license. Each license has a size, specified in license units. Each hardware system has a series of license unit requirements, also specified in license units.

The license unit requirements of a system are expressed in a **rating**. LMF stores ratings (in license units) for all available and appropriate Alpha and VAX systems in a called the **License Unit Requirement Table (LURT)**. A LURT is associated with each category of software products as identified on the PAK. The PAK contains two fields, the Activity Table Code and the Availability Table Code, that include information to identify the category of the software product. Typically, systems that provide more performance have greater license unit requirements, but ratings can be unrelated to performance.

Alpha and VAX systems are classified into three levels by system class:

- Enterprise system class
- Department system class
- Workgroup system class

Generally, the higher the system class, the larger the license must be to support the system.

The size of a software product license must be large enough to support the number of either users or processes using the product and the system on which the product is to run. LMF compares the size of a registered license to the rating of the current system and authorizes product use when a license supplies sufficient license units.

## Licensing on OpenVMS Alpha and OpenVMS VAX

### 2.1 License Units and License Unit Requirement Tables

For license ratings for Alpha and VAX systems, see the Software License Management website:

<http://licensing.hp.com/swl/view.slm?page=refmat>

To locate the number of units required for your system, first find the system model number in the rating table. On Alpha systems, find the particular system configuration, which lists the number of processor cores. Then you can see the number of operating system units and layered product units required by your Alpha system. On VAX systems, only a single unit rating is used for both the operating system and layered products.

Once your system is up, either booted fully or minimally, you can determine its license unit requirements by using the following command:

```
$ SHOW LICENSE/UNIT_REQUIREMENTS
```

LMF compares the size of a registered license to the license unit requirement for the system and authorizes product use when a license supplies sufficient license units.

To check whether your license has an appropriate license unit value (size) for your system, LMF performs the following process:

1. It looks for a code (*A, B, C, D, E, F, G, H* or *I*) or a keyword (*CONSTANT integer*) next to either the Availability Table Code: field or the Activity Table Code: field in the registered license.
  - If the license specifies *CONSTANT* and an integer value, LMF stops and defines the license unit requirement as equal to the stated integer value. This value could be the decimal value 0, which means the license has no unit limitations.
  - If the license does not specify a constant unit requirement, LMF looks for a code that specifies license type and corresponds to a LURT.

The codes and the types of licenses they represent as shown in Table 2–1.

**Table 2–1 License Codes**

License Code	License Type
A	VAX/VMS Capacity of OpenVMS Unlimited or Base
B	VAX/VMS F&A Server
C	VAX/VMS Concurrent User
D	VAX/VMS Workstation
E	VAX/VMS System Integrated Products
F	VAX Layered Products
G	Reserved
H	Alpha Layered Products
I	Layered Products

These codes and the units associated with them are shown when you issue a `SHOW/LICENSE/CHARGE` command.



## Licensing on OpenVMS Alpha and OpenVMS VAX

### 2.1 License Units and License Unit Requirement Tables

Each LURT has a rating, in license units, for all available (and appropriate) hardware systems. For example, the LURT for layered products includes the name of every hardware model that can run OpenVMS and associates a number of license units with each. The LURT for OpenVMS workstations includes the names of all OpenVMS workstations and their ratings. When HP releases new hardware systems, OpenVMS updates the tables as part of the new system support.

2. LMF determines the model name of your system by its **System Marketing Model (SMM)** name, which is the model name of a computer system used in marketing and pricing. The SMM is generally the name on the front panel of the system cabinet.
3. LMF locates the SMM in the appropriate LURT and selects the value that specifies the number of units required for the named SMM and type of license.
4. LMF compares the number selected from the LURT to the number of units registered for your product license. If you have registered a value sufficient for your license and system, the license is loaded successfully with the LICENSE LOAD command.

Table 2–2 shows an example of a License Unit Requirement Table.

**Table 2–2 Sample License Unit Requirements for Alpha Systems**

<b>System Model</b>	<b>Operating System Units</b>	<b>Layered Product Units</b>
AlphaServer GS160 with 2 CPUs	3000	1300
AlphaServer GS80 with 1 CPU	1000	1100
AlphaServer ES45 with 2 CPUs	100	1050

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 system. For example, when you obtain a license for HP Pascal to run on an AlphaServer ES45 system with 2 CPUs, that Pascal license must specify at least the same number of license units as the LURT requires for layered products on that system (1050 shown in Table 2–2). The same Pascal license may not provide enough license units to authorize use of Pascal on an AlphaServer GS160 system with 2 processors (1300 shown in Table 2–2). The size of the license to run a software product on an OpenVMS Cluster environment must reflect the total number of concurrent users or processes and the systems on which the product will run.

Not all licenses have a specific number of units. Some licenses specify zero units, which is equivalent to unlimited units.

# Licensing on OpenVMS Alpha and OpenVMS VAX

## 2.2 Types of Licenses

### 2.2 Types of Licenses

Different types of software product licenses enable you to allow access to each product in ways that range from access for a specific user on a specific system to general access for all users on all nodes within an OpenVMS Cluster. Table 2–3 describes the licenses that LMF supports on VAX and Alpha systems.

**Table 2–3 Types of Licenses**

Type	Identification on PAK	License by	See
Availability	Availability Table Code has a nonzero value.	System type (Requires Key Option ALPHA or VAX_ALPHA to load on Alpha system.)	Section 2.2.1
Activity	Activity Table Code has a nonzero value.	Concurrent uses (not users)	Section 2.2.2
User	Activity constant and Key Options: USER.	Concurrent users (not uses)	Section 2.2.4
Personal Use	Activity constant and Key Options: RESERVE_UNITS.	Named user	Section 2.2.3

The license descriptions that follow provide information to help you understand and manage the product authorization process on VAX or Alpha computers using LMF, rather than to help you order software licenses. HP provides licenses in many ways that may not always correspond to the examples in this manual. Check with your HP support representative for ordering information, and check the terms and conditions of your license contracts for restrictions.

#### 2.2.1 Availability Licenses

An **Availability License** makes a product available to all the users of a system. LMF can load a product when there are sufficient license units available in the LMF to satisfy the unit requirements of all the nodes in the cluster which have already loaded the product and enough units remaining available to satisfy the unit requirements of the requesting node. To authorize full availability on a system, LMF checks the Availability Table Code on the registered license and interrogates the LURT to determine the rating of the system. If the registered license provides enough license units, LMF loads the license, making the product available to all users on the named system.

For example, the PAK for the fictional layered product ALLSUM provides 1000 license units (Number of Units: 1000) and refers to LURT F, (Availability Table Code: F). When you register and load the license, LMF selects LURT F and checks whether the size of the license is at least as big as the number of license units required by the current system. If so, LMF authorizes full availability to ALLSUM on the current system.

In addition to authorizing use on a system, LMF allocates the required number of units to the system that loads the license. If a 1000-unit Availability License is registered in a common OpenVMS Cluster environment License Database, LMF can allocate a total of 1000 license units among several nodes in the cluster. For example, LMF can allocate the 1000 units to one node that requires 500 units, one that requires 300, and a third that needs 200 license units. This is known as **license sharing**.

## Licensing on OpenVMS Alpha and OpenVMS VAX

### 2.2 Types of Licenses

With an Availability License, LMF allocates license units to a system when you load a license. LMF returns the license units either when you use the LICENSE UNLOAD command or when the system is shut down.

---

#### Note

---

You can load an availability license on an Alpha system only if the PAK contains either Key Option: ALPHA or Key Option: VAX\_ALPHA. In addition, you can load a license authorized by a PAK with Key Option: ALPHA only on an Alpha system. However, the PAK can also safely reside in a License Database shared by both VAX and Alpha systems, and you can perform your License Database tasks from either a VAX or Alpha system.

---

#### 2.2.1.1 Providing Sufficient License Units

The license you register in the License Database should provide enough license units to satisfy the requirements specified in the LURT. Before you purchase a license, work with your software representative to assess your software and hardware requirements to ensure that you obtain a license of the correct size.

For standalone systems (including multiprocessors), HP offers licenses that exactly match the license requirements of a system. That is, there is a license size that matches each LURT entry.

Sometimes, users with multiple standalone systems cannot match their licenses to meet every circumstance. For example, you may manage two standalone systems: VAXBIG, which requires a 700-unit license, and VAXMID, which requires a 400-unit license. If you have one 700-unit license, you can load it on either system (but not both). If you have one 400-unit license, you can load it only on VAXMID. You can, however, still register the smaller license in the License Database of VAXBIG.

---

#### Note

---

If your license specifies the /MOD\_UNITS option, you can increase the number of units of the license (see Section 5.6.2). You can always decrease the number of license units, even if your PAK does not specify the MOD\_UNITS option.

---

#### 2.2.1.2 Providing More License Units

You may need to provide more license units than are currently registered in the License Database for a product. For example, you cannot load a 400-unit license on a system that requires 700 units. If you need more license units than are currently available, contact your software representative, who may recommend one of the following:

- A new license that provides all the required license units. This may involve deleting the current license (with the LICENSE DELETE command).
- Another license for the same product that provides 300 additional license units. If your license agreement allows it, you can register both licenses. LMF may combine the license units to produce the equivalent of a 700-unit license. For a description of license combination, see Section 2.3.

## Licensing on OpenVMS Alpha and OpenVMS VAX

### 2.2 Types of Licenses

- A different type of license. Some products offer Activity, Availability, Personal Use, and User licenses. Changing to a different type of license may provide greater access to the product.

#### 2.2.1.3 Providing Availability in an OpenVMS Cluster Environment

In an OpenVMS Cluster environment, you must consider all of the nodes that can load a product. To provide full availability for a product in an OpenVMS cluster environment with a common License Database, you must register licenses with a total number of license units at least as large as the total license unit requirements of all the nodes. For example, if the cluster consists of three VAX 8800 systems, each of which requires 1200 license units to run a specific product, you must register at least 3600 license units (1200 times 3) to provide full product availability across the cluster environment. Each node in the cluster will load 3600 units.

Why does each node in a cluster load all 3600 license units if it only uses 1200?

Think of the 3600 license units as a cluster-wide pool made up of the requirements of each of the nodes. Loading a license is like joining the pool: each node must be aware of the total size of the pool, the existing split, and its own requirements.

Assume that you have a three-node cluster, ALPHABET. ALPHABET consists of the nodes A, B, and C, which boot in that order whenever the cluster is rebooted. Also assume that ALPHABET has an Availability PAK and a valid LDB configured with 3600 units available. When node A boots, it sees 3600 license units. No other nodes are online, so it loads 3600 units and allocates 1200 of the units for itself.

Node B boots up and sees 3600 units registered and that A is using 1200 of the units. The remaining 2400 units is larger than the 1200 the node requires. Node B loads 3600 units and allocates 1200 for itself.

Node C boots up and sees 3600 license units registered. Nodes A and B have each allocated 1200 units, leaving a remainder of 1200, which is enough for Node C. Node C loads 3600 units and allocates 1200 for itself.

If you do not need product availability clusterwide, you can register licenses with total license units to authorize use by individual nodes. For example, in a cluster with three nodes each requiring 1200 license units, a 1200-unit license allows any one node to run the product, and a 2400-unit license allows any two cluster nodes to run the product concurrently. You can also use the LICENSE MODIFY command to allow or deny access to specific cluster nodes (see Section 5.6.2).

For example, suppose that on the ALPHABET cluster, you only want nodes A and C to use the product. You only need 2400 license units. The way to insure that only A and C can use the product is to place an INCLUDE or EXCLUDE list on all PAKs for that product.

To do this, issue the LICENSE MODIFY /INCLUDE command:

```
$ LICENSE MODIFY product /ALL/INCLUDE=(A,C)
```

You could also issue the LICENSE MODIFY /EXCLUDE command:

```
$ LICENSE MODIFY product /ALL/EXCLUDE=(B)
```

## Licensing on OpenVMS Alpha and OpenVMS VAX

### 2.2 Types of Licenses

Without the LICENSE MODIFY /INCLUDE or /EXCLUDE command, the first node to boot would find and load 2400 license units. The second node would find 2400 license units, note that another node is already using 1200 units and that the remainder, 1200 units, is sufficient to satisfy its requirements and load the entire 2400 units. The third node would also try to load the product, realize all the license units are already in use and fail.

Note that you cannot always manage licenses as previously described. For example, some licenses restrict a product to a certain system type, and other licenses with the NO\_SHARE option cannot share license units. As always, check the terms and conditions of your license contract.

#### 2.2.1.4 Providing More Availability

If you change the configuration of an OpenVMS Cluster by adding a node, or upgrading an existing node to a more powerful one, you may need to increase the number of available license units. You can provide more license units in the following ways:

- Register additional licenses.

If you choose to register an additional license (sometimes referred to as an additive PAK), you gain flexibility; the new license can be moved to different cluster environments or standalone systems independently. When you register the new license, LMF automatically combines the license size of the new license with the license size of the existing license to establish a new, higher level of license authorization. Management of separate licenses for the same product can be more complicated, however.

- Replace the current license with a larger one.

A new license with more license units (sometimes referred to as a replacement PAK), is easier to manage but is not as flexible, because it cannot be split among different cluster environments or standalone systems. You typically delete or disable the older license, then register the new one.

Your software representative can help you choose the option that fits your needs.

#### 2.2.2 Activity License

An **Activity License** defines the number of concurrent *uses* allowed for a product at any one time. Each product defines an activity as either an interactive user, a running process, volume shadowing, or a job. For example, when you register a 4-Activity License, LMF authorizes four concurrent uses of the product. Each time an activity invokes the product, LMF checks whether there are sufficient license units available to use the product on the current system, and if so, allocates the license units to that activity, reducing the number available to additional activities. When all license units are allocated, no new activity can invoke the product until an activity terminates use of the product (thereby deallocating license units).

As with an Availability License, an Activity License authorizes use through license units and LURTs. A product may require a certain number of license units per activity, regardless of the hardware system. For example, a 4-Activity License for a product that requires 100 license units per activity has a size of 400 license units, and allows up to four activities, whether on a MicroVAX system or a VAX 8800 system. The license unit requirement of the product is designated on the PAK as Activity Table Code: Constant=100.

## Licensing on OpenVMS Alpha and OpenVMS VAX

### 2.2 Types of Licenses

Other products require a certain number of license units per activity on a particular system. A 4-Activity License for a product that requires 100 license units per activity on a VAX 8800 system requires 400 license units. The same 100-unit license provides five concurrent uses on a system that requires only 75 units per activity (of that product).

One primary difference between an Availability License and an Activity License is the time at which LMF checks the number of license units authorized by a license, as follows:

- The Availability License allows unlimited access to the product after you successfully load the license on a system.
- The Activity License requires allocating available license units each time the product is invoked, and denies access when the activity limit is reached.

#### 2.2.2.1 Providing Enough License Units

As with Availability Licenses, you should try to match system, product, and license to your user requirements. Software vendors offer a variety of licenses that can match the license requirements of your users and your system. Before you obtain a product license, consult your software representative to define your software and hardware requirements to ensure that you obtain a license of the correct size.

The license you register in the License Database should provide enough license units to allow a predetermined number of activities access to the product. For example, if a software product requires 25 license units per activity on your system and PAKs come in 4-activity increments, the license you register should provide at least 100 units. Note that a 120-unit license provides no more use than a 100-unit license on such a system.

Different systems can have different license unit requirements per activity. Therefore, the number of users authorized by a license varies according to the system. For example, you may manage the following standalone systems:

- VAXBIG, which requires 25 license units per activity to authorize a product
- VAXMID, which requires 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 (with the LICENSE COPY command) to VAXMID when you shut down VAXBIG for maintenance. The 125-unit license, which allows 5 concurrent activities on VAXBIG, provides 6 concurrent activities on VAXMID. Note that you can also move an 80-unit (4-Activity) license originally intended for VAXMID to VAXBIG. However, on VAXBIG, the license provides access to only 3 activities.

As with Availability Licenses, you can register a license in the License Database even if that license cannot be successfully loaded. For example, if you register a 40-unit license that provides product access to two activities on a MicroVAX A system, the same license does not allow access to any activities on a system that requires 50 units per activity.

## Licensing on OpenVMS Alpha and OpenVMS VAX

### 2.2 Types of Licenses

#### 2.2.2.2 Determining Your License Needs

You may need to provide more license units than are currently registered in the License Database for a product. Each time a user is denied access to a product because of insufficient license units, LMF produces the following message:

```
-LICENSE-F-EXCEEDED, attempted usage exceeds active license limits
```

Analyzing the frequency of these messages can help you determine your license needs. If you need additional Activity License units, contact your HP support representative, who may recommend one of the following:

- A new license that provides all the required license units. This may involve deleting the current license (with the `LICENSE DELETE` command).
- Another license for the same product that provides additional license units. If the terms of your license agreement allow it, you can register both licenses. This allows LMF to combine the license units, providing more product use. For a description of license combination, see Section 2.3.
- A different type of license. Some products offer Activity, Availability, Personal Use, and User licenses. Changing to a different type of license may provide greater access to the product.

#### 2.2.2.3 Sharing License Units in an OpenVMS Cluster Environment

All cluster activities can access a product that has an Activity License registered in the common License Database. If your PAK specifies a constant number of license units per activity regardless of the system size, the cluster always provides access to the same number of activities. A 4-Activity License provides access to 4 activities whether the cluster has 1 node or 12 nodes, 1 MicroVAX system or 12 VAX 9000 systems.

In other cases, an Activity License may not specify a constant number of license units per activity on all nodes. Because the Activity License unit requirement can be different on each node in the cluster, the number of available activities depends on the system class of nodes involved.

For example, a particular Activity License might provide access to any 12 activities for a product on an OpenVMS Cluster with three VAX 8200 systems. If you add a node to the cluster that has a higher license unit requirement (than a VAX 8200), the number of concurrent uses allowed can decrease, because LMF allocates more license units per activity of the product on the additional node. You can modify the Activity License (using the `LICENSE MODIFY` command) to include or exclude specific nodes.

Note also that when the system starts up, LMF, by default, loads any licenses that do not have include or exclude lists. To control license loading, limit access with a `LICENSE MODIFY/EXCLUDE` or `LICENSE MODIFY/INCLUDE` command for each license that can be combined when licenses are loaded.

#### 2.2.3 Personal Use License

A **Personal Use License** designates the names of specific users for unlimited use of a product. Before you load the license, you specify the users allowed access. LMF adds these users to a **reservation list**, which is checked before granting access to each user who tries to invoke the product. A PAK for a Personal Use License specifies `RESERVE_UNITS` in the Key Options field. This license shares some characteristics with both the Availability and Activity Licenses.

## Licensing on OpenVMS Alpha and OpenVMS VAX

### 2.2 Types of Licenses

Although a personal use PAK includes an Activity Table Code, it does not limit access to concurrent use. While an Availability License authorizes product use by system, a Personal Use License authorizes product use by user name. LMF processes a list of authorized users when the license is loaded. After the license is loaded, any user on the list can access the product.

To calculate the allowed number of names on the reservation list, LMF divides the number of license units by the constant value listed in the Activity Table Code field of the PAK. If you register a 400-unit Personal Use License with a constant value of 100, LMF authorizes four specifically named users to access the product. If more than four names are associated with the license, LMF rejects extra names from the reservation list and denies access when those users attempt to access the product.

Personal Use Licenses are subject to combination rules that allow long lists of authorized users. See Section 2.3 for information about combining Personal Use Licenses.

Each Personal Use License must have an associated reservation list that specifies the name of each user with authorized access to the product. You cannot load a Personal Use License that does not have an associated reservation list with at least one user name. See Section 5.6.3 for information about controlling access to licenses with reservation lists using the LICENSE MODIFY command.

#### 2.2.4 User License

The **User License** shares some characteristics with the Activity and Personal Use Licenses, as follows:

- Each user is counted once towards the total number of concurrent users allowed. The terms of your license determine that total number.
- Each user granted access has unlimited access to the product until exiting the last invocation, but is counted against the authorized limit only once.
- Once the user has exited the last invocation of the product, renewed access will be denied if the total number of users of the product has reached the maximum authorized by the license.

As specified in the terms of your license, users can be people, disks, queues, applications, and others.

#### 2.2.5 Group License

A **Group License** authorizes access to a group of software products — usually related — that are licensed as one product. This enables you to license a group of products by registering only one PAK. A Group License can be any type of license: Availability, Activity, Personal Use, or User License.

All LICENSE commands use the group name for the *product-name* parameter instead of the individual product names. For example, a Software Group called COMPILER\_1 might include HP Fortran, HP Pascal, and HP COBOL. You register the Group License as COMPILER\_1 with one PAK and enter all LICENSE commands using COMPILER\_1 as well.

Group PAKs do not look different from single product PAKs. For information about the products licensed by a group PAK, see the Software Product Description (SPD) for the group product.



## 2.3 License Combination

License combination allows LMF to create large licenses by adding together the license units of multiple licenses. For example, two 50-unit Availability Licenses become equal to one 100-unit Availability License. Ten 100-unit Personal Use Licenses become equal to one 1000-unit Personal Use License.

License combination and loading are controlled by both the terms of your PAK and options you set with the LICENSE MODIFY/COMBINE and LICENSE MODIFY/NOCOMBINE commands.

LMF automatically combines some licenses by default.

### 2.3.1 Licenses That can be Combined

When a system loads a license, LMF scans the License Database for all **combinable licenses** and makes a pool of license units available for use. Licenses are combinable if they have matching data in each of the following data fields:

- Product name
- Producer
- Availability
- Activity
- Key Options: RESERVE\_UNITS, USER, NO\_SHARE (assigned node must match), ALPHA, or VAX\_ALPHA
- Product Token
- Hardware-ID

VAX Availability, ALPHA Availability, VAX\_ALPHA Availability, User, Activity, and Personal Use licenses are different types of licenses. Therefore, they do not combine.

LMF matches any two empty data fields and, in the Availability and Activity fields, also matches the entry CONSTANT=0 with an empty data field. Licenses with the NO\_SHARE option can combine, but they must have matching include lists that assign each license to the same node. This is the only time either an include list or an exclude list has an effect on license combination.

By default, LMF does not automatically combine otherwise combinable licenses if any one of the following attributes do not match:

- Termination Date
- Release Date
- Version
- Reservation List

If two or more licenses are combinable except for the above attributes, you can force LMF to combine them with the following command:

```
LICENSE MODIFY product-name /COMBINE
```

## Licensing on OpenVMS Alpha and OpenVMS VAX

### 2.3 License Combination

#### 2.3.2 Include, Exclude, and Reservation Lists

If you register a combinable license without an include or exclude list, any system can load the license with access to the entire pool of combined license units, with the following results:

- The entire pool of Availability License units becomes available to the system that loads the license. LMF allocates the number of license units required for each system as it loads the license. The appropriate LURT defines this number.
- The entire pool of Activity License units can be available to any user (or activity) that loads the product. LMF allocates the license units as each user accesses the product.

By default, when combining Activity Licenses, LMF combines those without reservation lists into one license without a reservation list, and those with reservations lists into one license with a reservation list that combines the separate reservation lists.

By default, when combining User Licenses, LMF combines those without reservation lists into one User License without a reservation list, and those with reservations lists into one User License with a reservation list that combines the separate reservation lists.

By default, when combining Personal Use Licenses, LMF combines any reservation lists associated with each license into one large reservation list that applies to all the combined licenses.

#### 2.3.3 Termination Dates and Version Numbers

With the forced combination of multiple licenses, LMF sets the termination date, release date, and version number of the combined license to the earliest dates and version numbers that apply to the individual licenses being combined. The following table shows the combined license that results from the forced combination of two licenses:

	<b>License 1</b>	<b>License 2</b>	<b>Combined License</b>
Version Number	2.3	2.0	2.0
Release Date	1-JAN-2005	30-NOV-2005	1-JAN-1995
Termination Date	1-JAN-2008	30-SEP-2005	30-SEP-2005

---

## Licensing on OpenVMS Integrity servers

This chapter describes licensing on OpenVMS Integrity server systems, which differs from licensing on Alpha and VAX in several ways. Key differences are described in the following sections:

- Operating environments and tiering
- New license type—per core licenses (PCL), which replaces per processor licenses (PPL)
- Compliance checking and reporting

### 3.1 Operating Environment and Tiering

On OpenVMS Integrity server systems, operating system licenses are bundled with additional products into **operating environments**, called OEs. These environments offer base operating system functionality along with additional capability, based on the OE. The operating environments are tiered in a hierarchy. Each higher-level OE contains everything in the lower tiers plus additional functionality.

With OpenVMS Version 8.4, the three-tier Operating Environment model that consisted of Foundation Operating Environment (FOE), Enterprise Operating Environment (EOE) and Mission Critical Operating Environment (MCOE) is replaced with a two-tier Operating Environment model consisting of Base Operating Environment (BOE) and High Availability Operating Environment (HA-OE).

#### Base Operating Environment (BOE)

This OE includes the base operating system plus networking transport, internet capability, and other basic functions. BOE license includes all components of FOE and two components from EOE. The following products that were formerly part of the EOE are now part of BOE:

- DECram for OpenVMS
- HP OpenVMS Management Station

#### High Availability Operating Environment (HA-OE)

The HA-OE contains everything in BOE and additional system management capability, volume shadowing, clustering, and other advanced capabilities. HA-OE license includes four components of EOE, all components of MCOE, and Global Work Load Manager (gWLM). The following products that were formerly part of the EOE are now part of HA-OE:

- HP Availability Manager
- OpenView Performance Agent (OVPA) for OpenVMS
- RMS Journaling for OpenVMS

## Licensing on OpenVMS Integrity servers

### 3.1 Operating Environment and Tiering

- HP Volume Shadowing for OpenVMS

These operating environments have new licenses associated with them as follows:

- OPENVMS-I64-BOE
- OPENVMS-I64-HAOE

The FOE licenses automatically upgrade to BOE and EOE or MCOE licenses automatically upgrade to HA-OE, upon upgrading to OpenVMS Version 8.4.

The operating environment license grants use of all products within the OE. Additionally, some components of the OEs (like clustering, which is part of the HA-OE) are available individually. For example, you can add a cluster license to the Base operating environment. The combination of OE tiering and the ability to add individual components allows you to tailor your environment to best meet your needs.

Information on the products contained in each operating environment is stored in a datafile (LMF\$OE.DAT). The contents of the datafile are loaded into memory when the system is booted. Over time, the contents of the various OEs may change or new OEs may be offered by HP. When that occurs, HP will provide a new LMF\$OE.DAT datafile that contains information on the new or changed OEs. You can use the LICENSE LOAD/OEDB command to update the OE database on your system with the new information. Consult the current Operating Environment SPD (SPD 82.34.xx) for information on the contents of all available operating environments and their contents.

Once you have registered your PAK and loaded your OE license, you can see information about the available operating environments, the hierarchy among them, and the products contained in each OE by using the following command:

```
$ SHOW LICENSE/HIER/FULL
```

```

                                Operating Environment Hierarchy
                                -----
----- Operating Environment ----- Units -----
Name      Description              Type Level  Loaded  Total
HAOE      High Availability            H     5     2      2
  GWLM
MCOE      Mission Critical                H     4     -      2
  RTR-SVR
  VMSCUSTER
  VMSCUSTER-CLIENT
EOE       Enterprise                        H     3     -      2
  AVAIL-MAN
  RMSJNL
  VOLSHAD
BOE       Base                              H     2     -      2
  DEGRAM
  OMS
FOE       Foundation                         H     1     -      2
  OPENVMS-I64
  OPENVMS-USER
  DVNETEND
  DW-MOTIF
  UCX
  TDC
  X500-ADMIN-FACILITY
  X500-DIRECTORY-SERVER
  CIFS

```

## Licensing on OpenVMS Integrity servers

### 3.1 Operating Environment and Tiering

The example lists each OE and its contents in a hierarchical fashion so that the products contained in each OE are identified. The display also shows the number of units loaded.

To see the contents of a single OE, for example HA-OE, use the following command:

```
$ SHOW LICENSE/OE=HAOE/FULL
----- Operating Environment ----- Units -----
Name      Description          Type Level  Loaded  Total
HAOE      High Availability      H   5      4       4
  GWLM
  RTR-SVR
  VMSCUSTER
  VMSCUSTER-CLIENT
  AVAIL-MAN
  RMSJNL
  VOLSHAD
  DEGRAM
  OMS
  OPENVMS-I64
  OPENVMS-USER
  DVNETEND
  DW-MOTIF
  UCX
  TDC
  X500-ADMIN-FACILITY
  X500-DIRECTORY-SERVER
  CIFS
```

This example shows all products within the HA-OE without distinguishing between operating environment hierarchies. All products contained in HA-OE and BOE are listed together.

You can upgrade or downgrade your OE without a reboot using LMF. For example, you may want to change the number of processor cores on a system, change the OE available on a particular node, or move a license to another node. Any of these actions may require upgrading or downgrading your OE. This flexibility allows you to make maximum use of the products for which you are licensed. To upgrade to a higher OE tier or to add license units, register and load the new license into the database using the LICENSE REGISTER and LICENSE LOAD commands. To downgrade, use the LICENSE UNLOAD command.

## 3.2 License Types

On OpenVMS Integrity server systems, there are two license types:

- Per core licenses - for OpenVMS Integrity server systems, replaces per processor licenses (PPL)
- Activity licenses

The following sections describe these licenses.

### 3.2.1 Per Core Licenses

An Integrity-specific license type, **per core license (PCL)**, implements the licensing model on OpenVMS Integrity server systems. The PCL model licenses a product based on the number of active processor cores on the system, not the static rating scheme as on Alpha and VAX systems. Each active processor core requires one PCL unit. If you increase or decrease the number of active processor cores on a system, the requirement for PCL licenses changes.

## Licensing on OpenVMS Integrity servers

### 3.2 License Types

A PCL license is required to run operating environments, OE products purchased separately (like clustering), and many standalone products on OpenVMS Integrity server systems.

PCL licenses offer flexibility as you can purchase licenses in the exact number you need and you can move the licenses to other processors. If you upgrade or reconfigure your system with additional processor cores, you purchase additional PCL licenses.

LMF constantly checks the number of PCL licenses against the number of active processor cores and enforces a soft compliance model described in Section 3.3.2. Any changes to the system are noted and checked for compliance.

---

#### Note

---

If older PAKs are installed, "PPL" may still be displayed; however, the licenses will be managed as if they are the new PCL type.

---

#### 3.2.1.1 Per Socket Licenses

HP Integrity server blades (BL860c i2, BL870c i2, and BL890c i2) are licensed on a per-socket basis for operating environments and layered products. However, LMF manages the per socket licenses (PSL) identical to PCL. The quantity of LMF units on a per socket license is equal to the maximum number of cores available on the licensed socket. Hence, the per socket licenses are shipped with the PCL option in the **license option** field. For example, purchase of a two socket license for quad core processors results in a quantity of eight PCL units on the issued license.

#### 3.2.2 Activity Licenses

For layered products such as compilers, activity licenses like those on Alpha and VAX systems are used. The units for these products are expressed in multiples of 1, rather than the 100s as on Alpha and VAX. A sample PAK for C might look like the following:

```
ISSUER: HP
AUTHORIZATION NUMBER: USA126087
PRODUCT NAME: C
PRODUCER: HP
NUMBER OF UNITS: 3
VERSION:
PRODUCT RELEASE DATE:
KEY TERMINATION DATE: 31-DEC-2004
AVAILABILITY TABLE CODE:
ACTIVITY TABLE CODE: CONSTANT=1
KEY OPTIONS: MOD_UNITS
PRODUCT TOKEN:
HARDWARE I.D.:
CHECKSUM: 1-BGON-IAMA-LEEH-EPEL
```

In this example, up to 3 users are licensed to use the C compiler concurrently. If a fourth user attempts to use the C compiler, that user receives the following message:

```
-LICENSE-F-EXCEEDED, attempted usage exceeds active license limits
```

LMF will not allow the fourth user access to the C compiler. This behavior is identical to that on OpenVMS Alpha and VAX systems.

### 3.3 Compliance Reporting

On OpenVMS Integrity server systems, LMF checks for two types of compliance:

- Hardware compliance - checks license against hardware system rating
- Soft compliance - checks number of PCL licenses against the number of active processor cores

The following sections describe each.

#### 3.3.1 Hardware Compliance

To run an operating environment on an Integrity server system, you must have a license appropriate for your system rating based on **sockets**. A socket is a receptacle into which a processor module can be installed. Each processor module can contain one or more processor cores. Your PAK for an Integrity server system may have an entry in the `HARDWARE_ID` field (expressed as `SOCKETS=n`). For example, if your PAK has the entry `SOCKETS=4` in the `HARDWARE_ID` field, you can load that license on a 1, 2, 3 or 4-socket system. If you try to load a license for a 2-socket system on a 4-socket system, the license will not load. In this case, LMF is enforcing a hard compliance check.

You can check the number of sockets on a system by using the following command:

```
$ SHOW LICENSE/CHARGE_TABLE
OpenVMS I64/LMF Charge Information for node ADI26B
This is an HP rx2600 (900MHz/1.5MB), with 2 cores active
Type: PPL, Units Required: 2 (I64 Per Processor)
Type: PCL, Units Required: 2 (I64 Per Core)
```

This example shows that node ADI26B has 2 sockets. Also, note that the example displays both the PPL and PCL types, because of the number of licenses sold.

You can buy a license for an unlimited number of sockets. In that case, there is no keyword specified in the `HARDWARE_ID` field.

To ensure hardware compliance, add an include or exclude list to your licenses by using the `/INCLUDE` or `/EXCLUDE` parameter to the `/HARDWARE_ID=SOCKET` tag. For example, if you are using a common license database in a cluster with one HP Integrity rx4640 server (4 sockets), two HP Integrity rx2620 servers (2 sockets), and one HP Integrity rx8620 server (16 sockets), verify that the units in the 16-socket license are used only on the rx8620. For a description of adding an include or exclude list to a license, see Section 5.6 or the `LICENSE_MODIFY` command in Appendix A.

#### 3.3.2 Soft Compliance

In addition to having a license appropriate for your system hardware rating, you must also have a per core license (PCL) for each active processor core. The PCL units for Integrity server systems are in units of 1 per processor core.

To assist you in maintaining the terms and conditions of your licensing agreement, HP provides a reporting mechanism that flags noncompliance for per core licenses. For example, if you load a license with only 2 units a system with 4 active processor cores, the license will load but a message indicating that the system is out of compliance is logged to the OPCOM facility and a mail message is sent to the SYSTEM account. You can bring this system back into

## Licensing on OpenVMS Integrity servers

### 3.3 Compliance Reporting

compliance in two ways: load a license with 2 additional units or deactivate 2 of the 4 processor cores.

This soft compliance mechanism gives you flexibility to alter your system configuration temporarily and reminds you that you need additional per processor licenses to run in a compliant mode. LMF checks for compliance periodically and continues to log messages to the OPCOM facility and send mail to the SYSTEM account at predetermined intervals until sufficient PCL units are loaded on to the system to bring it into compliance.

Vendors who utilize LMF to manage their product licensing may choose to use a hard compliance model. If a vendor wants to enforce hard compliance, they can generate a PAK with the keyword `HARD_COMPLIANCE` in the `OPTIONS` field. Under a hard compliance policy, the license will not load and a user cannot run the application if they do not have a sufficient number of PCL units for each active processor core.

#### 3.3.3 Sample License Checks

The following examples show how LMF combines checking the hardware rating and the PCL licensing requirements for OpenVMS Integrity server systems.

An rx2600 is a 2-socket system and each socket can accept only a single-processor core module. Hence, the maximum number of processors on an rx2600 is 2. A PAK for for base operation environment on this system might look like the following:

```
ISSUER: HP
AUTHORIZATION NUMBER: USA126087
  PRODUCT NAME: OPENVMS-I64-BOE
    PRODUCER: HP
      NUMBER OF UNITS: 1
        VERSION:
          PRODUCT RELEASE DATE:
            KEY TERMINATION DATE: 31-DEC-2012
              AVAILABILITY TABLE CODE:
                ACTIVITY TABLE CODE:
                  KEY OPTIONS:
                    PRODUCT TOKEN:
                      HARDWARE I.D.: SOCKETS=2
                        CHECKSUM: 2-NFCA-BAPO-FEJF-BDLL
```

The example shows the maximum number of sockets for this system as 2, as specified by the `SOCKETS=2` keyword to the `HARDWARE_ID` field. This license could be loaded on any system with 1 or 2 sockets. The number of processor cores authorized to run the Base Operating Environment (BOE) by this license is 1, as specified in the `NUMBER OF UNITS` field. If you wanted to add another processor, you would need to purchase an additional PCL license with 1 unit.

An rx4640 system is a 4-socket system and each socket can accept either a single-processor core or a dual-processor-core module. This system may have up to 8 processor cores, depending on how it is configured. A PAK for the base operating environment on this system might look like the following:



## Licensing on OpenVMS Integrity servers

### 3.3 Compliance Reporting

```
ISSUER: HP
AUTHORIZATION NUMBER: USA126088
  PRODUCT NAME: OPENVMS-I64-BOE
    PRODUCER: HP
      NUMBER OF UNITS: 2
        VERSION:
          PRODUCT RELEASE DATE:
            KEY TERMINATION DATE: 31-DEC-2012
              AVAILABILITY TABLE CODE:
                ACTIVITY TABLE CODE:
                  KEY OPTIONS:
                    PRODUCT TOKEN:
                      HARDWARE I.D.: SOCKETS=4
                        CHECKSUM: 2-FODI-MDHC-PAIF-CEEG
```

The example shows the maximum number of sockets for this system as 4, as specified by the SOCKETS=4 keyword in the HARDWARE\_ID field. This license could be loaded on any system with 1 to 4 sockets. The number of processor cores authorized to run the base operating environment by this license is 2, as specified in the NUMBER OF UNITS field. If you wanted to add more processor cores, you would need to purchase additional PCL units.



---

## Licensing OpenVMS Guests on Integrity VM

This chapter describes the licensing of OpenVMS guests on Integrity VM, which differs from licensing physical OpenVMS Integrity server systems in a few ways. Key differences are described in the following sections:

- License Distribution
- Standalone System Licensing Requirements
- Cluster Licensing Requirements
- Compliance Checking and Reporting

### 4.1 License Distribution

OpenVMS PCL licenses for the Integrity VM environment are distributed based on the number of host machine cores that can be used for any OpenVMS guest instance. Any OpenVMS guest number may use virtual CPUs up to the number of host machine cores that are licensed.

### 4.2 Standalone System Licensing Requirements

If the OpenVMS guest system that you are running is not a member of a cluster, there is no special licensing operation that you need to perform. You can run as many standalone OpenVMS guests on a host system that are physically possible. Each guest loads its own copy of the license database.

If you plan to add your OpenVMS guest system to a cluster in the future, HP recommends that you modify the PCL licenses with the `/VIRTUAL` qualifier option described in Appendix A.

### 4.3 Cluster Licensing Requirements

Licensing on OpenVMS guests allows the running guests to consume more units than are available on the OpenVMS PCL licenses issued for the host system. As long as the OpenVMS guest system is running with a number of virtual CPUs less than or equal to the number of physical cores licensed on the host system, the guest is licensed. LMF allows as many OpenVMS Guests that can be physically created on the host system to run. To support the new unlimited usage virtual licensing, a new license option; `VIRTUAL` is required for all PCL licenses that you want to use on OpenVMS guests in an OpenVMS Cluster. The `VIRTUAL` option is not required for Activity licenses.

## Licensing OpenVMS Guests on Integrity VM

### 4.3 Cluster Licensing Requirements

#### 4.3.1 New License Modify Qualifier `/[NO]VIRTUAL` Required for OpenVMS Guest Systems

Prior to adding your OpenVMS guest system to a cluster, you must modify all PCL licenses you want to load on OpenVMS guests with the `LICENSE MODIFY /VIRTUAL` command. If you are using a common `LMF$LICENSE` database, load your HOST system licenses into the database and modify them with the `/VIRTUAL` qualifier. OpenVMS guests in clusters only load licenses that are tagged with the `VIRTUAL` option. If you use private LMF databases for each guest, you must modify the licenses with the virtual flag prior to adding the OpenVMS guest to a cluster. You can do this from the "Execute DCL Command" option of the OpenVMS Installation Menu or by booting the system standalone prior to adding it to an OpenVMS Cluster.

As previously required by LMF, if you use multiple `LMF$LICENSE.LDB` databases, ensure that you register all licenses in every `LMF$LICENSE.LDB` database used in the cluster. Additionally, modify all licenses intended to run on OpenVMS guests with the `/VIRTUAL` qualifier in all the databases.

#### 4.3.2 Virtual License Loading

Loading the licenses with the `Virtual` option in a cluster is restricted as follows:

- OpenVMS guest cluster members can only load licenses with the `VIRTUAL` option. Use the `/LOG` qualifier with the `LICENSE LOAD` command to log the licenses that are ignored when loading on an OpenVMS guest.
- Non-guest OpenVMS Version 8.4 systems cannot load licenses with the `VIRTUAL` option. Use the `/LOG` qualifier with the `LICENSE LOAD` command to log the virtual licenses that are ignored when loading on a physical OpenVMS system.

#### 4.3.3 Virtual License Recommendations

To facilitate the ease of use and migration to OpenVMS guests, the initial implementation of guest licensing is not restrictive. Compatible OpenVMS PCL licenses can be shared among OpenVMS guests from different host systems. After the OpenVMS Version 8.4 release, LMF may be changed to restrict the usage of issued licenses to the host system, which loads the license first.

To ensure that your licenses load on the OpenVMS guest cluster members you intend to load in the future, it is recommended you use `/INCLUDE` or `/EXCLUDE` lists to define which OpenVMS node must load the license. This is similar to modifying licenses with the `/HARDWARE_ID=SOCKETS=n` with include or exclude lists to ensure that the licenses load on the intended systems.

### 4.4 Compliance Checking and Reporting

Licenses for OpenVMS guests must have sufficient units to license all of the virtual CPUs running on the guest system. Unlike physical machines, you are not allowed to run in a non-compliant mode, where the number of units on the license is less than the number of active cores. The compliance check is performed during the license load.

Issuing the `SHOW LICENSE/USAGE` command from an OpenVMS guest cluster member displays `Virtual Machine guest`, no usage information for PCL licenses loaded on the system. There is essentially no usage charge against the license units for OpenVMS guest nodes since multiple guests can run on the same host using the same license units.

## Licensing OpenVMS Guests on Integrity VM 4.4 Compliance Checking and Reporting

For example:

```
$ SHOW LICENSE/USAGE
```

```
View of loaded licenses from node HOVMS2
```

```
20-DEC-2009 08:38:17.13
```

```
----- Product ID ----- ---- Unit usage information -----  
Product      Producer      Loaded  Allocated  Available  Compliance  
OPENVMS-I64-HAOE  HP          Virtual Machine guest, no usage information
```



This chapter provides details about the tasks involved in managing software licenses. Topics covered include:

- Preparing for license registration (see Section 5.1)
- Managing the License Database (see Section 5.2)
- Getting a Product Authorization Key (PAK) (see Section 5.3)
- Registering licenses (see Section 5.4)
- Loading a license (see Section 5.5)
- Managing licenses after registration (see Section 5.6)

In addition, this chapter contains a clarification about using logical name LMF\$DISPLAY\_OPCOM\_MESSAGE (see Section 5.7).

## 5.1 Preparing for License Registration

To license and use many software products on the OpenVMS operating system, follow at least these four steps:

1. Obtain a PAK for your product.

This is usually a hardcopy or electronic document containing information similar to that shown in Example 5–1. Order it from the software license issuer or software product producer.

2. Register information from the PAK into the License Database.

Use command procedure VMSLICENSE.COM to prompt for license registration information or enter the LICENSE REGISTER command directly. Example 5–3, produced with a LICENSE LIST command, shows a license registered in the License Database. In this manual the PAK information registered in the License Database is called a license.

3. Ensure that the system loads the registered license.

LMF requires that a registered license be loaded before you can use the product. When you register a license with VMSLICENSE.COM, you can confirm an option to load the license automatically. If you register a license with the LICENSE REGISTER command, you must also load it with a LICENSE LOAD command in order to use the product. At system startup, LMF automatically loads registered licenses.

4. Install the product that corresponds to the license.

Although the terms and conditions of license contracts vary, generally a license correlates with a particular release of a product. Because there are multiple factors that can affect the use of a license, such as the product release date, a version check, or a termination date, and because LMF allows

## Using LMF

### 5.1 Preparing for License Registration

products to check the License Database for properly registered licenses, you must match the license to the product.

After performing these steps, you can modify the license for a system or involve multiple systems in a licensing scheme (if your license agreement allows it).

For example, you want to restrict a license used in an OpenVMS Cluster environment to a specific node. If you register a license that uses the NO\_SHARE option (an OpenVMS operating system license, for instance), assign the license to a specific node. Either enter a LICENSE MODIFY/INCLUDE=*node-name* command or respond to the prompt for a System Communications Services (SCS) node name in VMSLICENSE.COM (see Section 5.6.2 for details).

### 5.2 Managing the License Database

LMF stores all information about licenses in the License Database. By default, LICENSE commands refer to the default license database, and you usually do not need to know the name and location of the database. However, for system management reasons, you may need to move the database. This section describes techniques for accessing license information and moving the license database.

Most of the data fields in the License Database correspond to either the LICENSE qualifiers or to responses to command procedure prompts. For example, the authorization field contains the data entered with the following command:

```
$ LICENSE REGISTER /AUTHORIZATION=string product-name
```

If you enter USA1234 for the string, USA1234 becomes the data in that field.

When you first register a license, you create the first record with data matching your PAK. When you enter other LICENSE commands, LMF creates new records to include any changes you make. For example, when you enter a LICENSE MODIFY command, LMF creates a new record marked with the new information, including a notation that the license was modified.

For performance reasons, License Database information is duplicated in memory while your system is running. LICENSE commands impact the database stored on disk. To update the License Database information in memory, use the LICENSE LOAD or LICENSE UNLOAD commands.

#### 5.2.1 Database Location

If you move the database to another directory or disk, or rename the database file, you must either define the logical name LMF\$LICENSE at the system level to point to the new database, or you must use the /DATABASE=*filespec* qualifier with all LICENSE commands. Place permanent systemwide logical name definitions in the file SYS\$COMMON:[SYSMGR]SYSLOGICALS.COM.

If you have multiple system disks in an OpenVMS Cluster environment where all the systems can access one of the system disks, put your common License Database on the readable disk. For any systems that boot from a separate system disk, you must redirect LMF to the License Database. Define the logical name LMF\$LICENSE to be the disk where the database exists.

If you have multiple system disks in an OpenVMS Cluster environment where some systems cannot access one of the system disks, you must keep separate identical License Databases. Whenever one database is modified, you must copy it to update the other databases.

HP recommends you back up the License Databases after every modification.



### 5.2.2 History Records

Your system maintains history records. Each history record contains an exact copy of the license record before modification, the LICENSE command used to modify the record, the issuing username, and the date and time of modification.

History records accumulate over time and provide a comprehensive audit trail of all modifications you make to the License Database. Most software issuers, including HP, require that you retain this information to demonstrate that you are complying with license terms and conditions.

To display history information, enter the following command:

```
$ LICENSE LIST /HISTORY
```

To create a hard copy, enter the following command:

```
$ LICENSE LIST /HISTORY /OUTPUT=LICENSE.LIS
$ PRINT LICENSE.LIS
```

Over time, LICENSE commands, including the LICENSE START command issued automatically during system startup, might take longer than usual to execute. This could be due to an accumulation of license history records in the License Database.

If you notice delays, HP recommends that you purge the history records in your active License Databases, but only after first preserving this information in one or more backup locations. Use the DCL command COPY or the Backup utility to make a copy of the License Database, thereby preserving the current version of the License Database, including history records.

To purge history records, enter the following command:

```
$ LICENSE DELETE /STATUS=EXTINCT *
```

---

#### Caution

---

Ensure that you do not omit the /STATUS=EXTINCT qualifier in the above command. If you do, all license records are deleted, leaving your License Database empty.

---

LICENSE DELETE deletes all history records, making them invisible to subsequent LICENSE commands.

Creating a new, compressed version of the License Database reclaims the disk space formerly occupied by the now deleted history records. To create a compressed License Database, use the DCL Convert utility (CONVERT).

## 5.3 Getting a Product Authorization Key (PAK)

Generally, you obtain both a PAK and the product from a representative of a company that distributes software. You order a PAK just as you order another product from HP or another company. HP provides PAKs on paper certificates, traditional media, compact disc read-only memory (CD-ROM), or by telephone or network so that you can register product data in the License Database. LMF needs specific values from a PAK to identify the source of the PAK and the source of a product.

## Using LMF

### 5.3 Getting a Product Authorization Key (PAK)

A PAK comes from a **PAK issuer**—the LMF name for the entity that supplies the PAK. Currently, licenses for Alpha and VAX systems specify DEC for the PAK issuer, and currently DEC is the default character string when you register a PAK with VMSLICENSE.COM. A PAK issuer string can also be DEC-USA or DEC-EUROPE to differentiate regions or departments within HP. For Integrity server systems, licenses specify HP for the PAK issuer. Other software vendors provide their own PAK issuer strings with their licenses. LMF uses the string to differentiate between different sources of licenses.

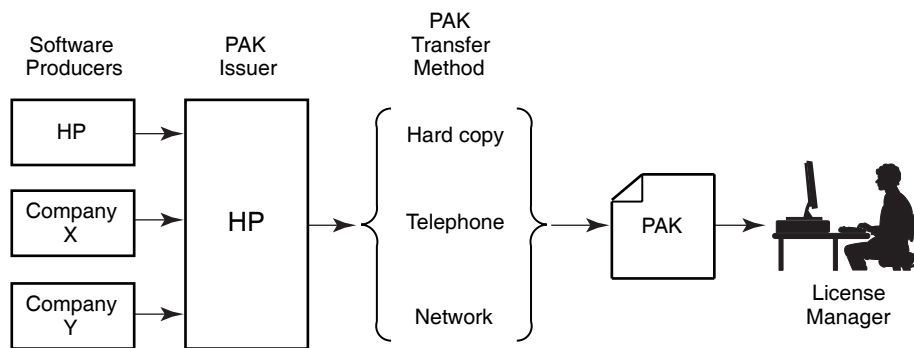
HP may distribute and issue a PAK for a product that it does not produce. Thus, LMF also uses a string that identifies a **software producer**. A producer is the company that supplies the software product. Generally, a producer and a PAK issuer are the same. The current default producer name when you register a PAK with VMSLICENSE.COM is DEC.

The OpenVMS operating system and LMF use PAKs to authorize most products for use. For example, after you install OpenVMS, you may have all the software required to use the System Integrated Products (SIPs) such as networking, RMS Journaling, and Volume Shadowing. To enable a SIP, register its PAK and load the license (there is no separate installation media). Even when you receive multiple software products on one HP CD-ROM, register a PAK for each product to enable the software.

Some products follow the older product distribution and license approach, providing installation kits that include distribution media and documentation. If a kit does not include the PAK, order it separately.

Figure 5–1 illustrates the PAK transfer process.

**Figure 5–1 PAK Transfer Methods**



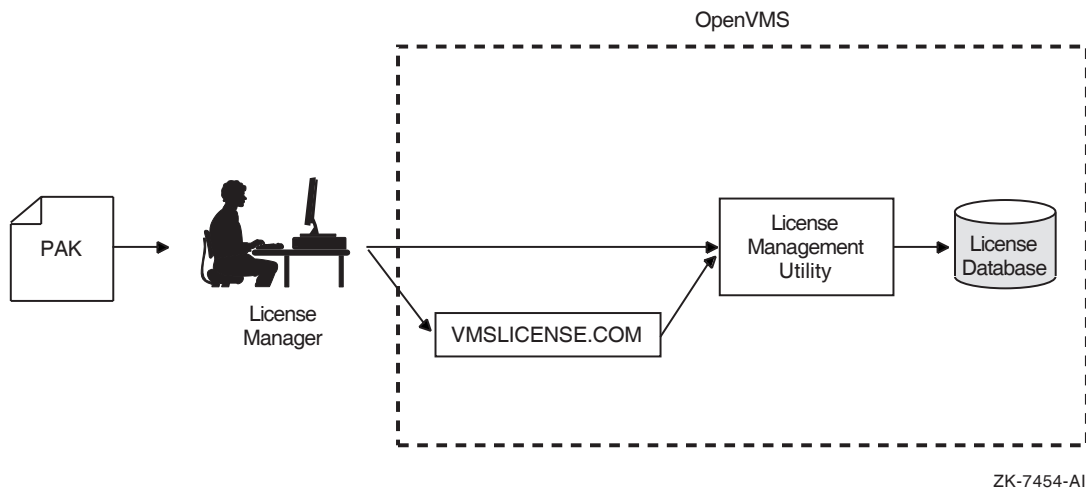
VM-0776A-AI

## 5.4 Registering Licenses

To run most HP software products, including the OpenVMS operating system, you must first register the product license in the License Database and then load the registered license. In addition, many third-party vendors of OpenVMS layered software also require you to use LMF to complete the same licensing tasks for their products.

Section 5.1 describes the registration options and presents examples of registration. Figure 5–2 illustrates the routes from a PAK to the License Database.

Figure 5–2 From a PAK to the License Database



### 5.4.1 When To Perform Registration

Most HP software that runs on OpenVMS systems and many third-party software layered products use LMF. To check a product's licensing requirements, see its installation manual or release notes. These documents explain which products use LMF registration.

If a product uses LMF, you must obtain a PAK, which includes the appropriate data for you to enter. Example 5–1 show a typical PAK for an Alpha system.

## Using LMF

### 5.4 Registering Licenses

#### Example 5–1 Typical PAK Information

```
ISSUER: DEC
AUTHORIZATION NUMBER: USA126087
PRODUCT NAME: CRYPTICALMENT
PRODUCER: DEC
NUMBER OF UNITS: 460
VERSION: 8.2
PRODUCT RELEASE DATE:
KEY TERMINATION DATE: 31-DEC-2004
AVAILABILITY TABLE CODE: E
ACTIVITY TABLE CODE:
KEY OPTIONS: MOD_UNITS
PRODUCT TOKEN:
HARDWARE I.D.:
CHECKSUM: 1-BGON-IAMA-GNOL-AIKO
```

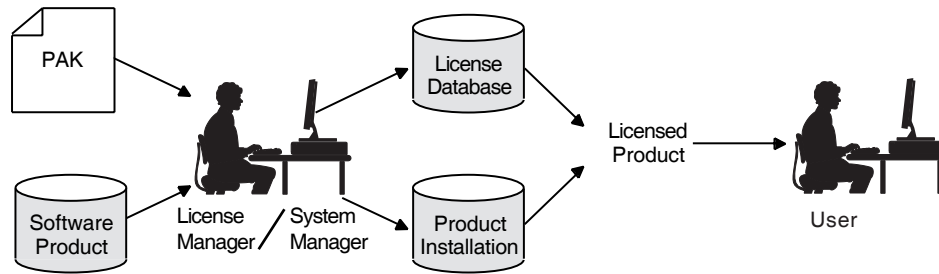
#### 5.4.2 Registration and Installation

Follow the licensing and installation procedure provided with each product. You can save time if you consider the following variations and consequences for product installation and license registration:

- If you register a license before you install a product, the product installation can be somewhat faster. You should register the license first, even though some products may allow installation first.
- If you start to install a product and realize you need to register a license for it first, you can register the product from another session while the installation session waits at the “Is there a license PAK registered for this product?” After you register and load the license, you can use the product. Be sure to reply correctly to any licensing questions during the product installation. Check your product installation guide for specific restrictions.
- To add a new node to an OpenVMS Cluster, you can register the new OpenVMS license before you add the node. You do not usually have to install the product again, unless the new node uses a new system disk.
- If you are upgrading an OpenVMS Cluster environment, you may want to register all the OpenVMS licenses at one time after one node is operating. This eliminates some messages when the other nodes start up and keeps your nodes more available for interactive use. Typically, on Alpha and VAX systems you assign licenses by processor type. For example, you should not assign a license intended for an Alphaserver 8400 system to a VAX 6000 system.

Figure 5–3 illustrates the license registration and product installation route both for processors running the OpenVMS operating system and for layered products.

Figure 5–3 The PAK and Software Routes to a License



ZK-7453-AI

### 5.4.3 Registration Methods

Before you install a product,<sup>1</sup> register licenses in the License Database by entering PAK information in one of the following ways:

- In response to prompts from SYS\$UPDATE:VMSLICENSE.COM. This command procedure provides some default data and a menu-driven interface to help register the license.
- With a LICENSE REGISTER command. The qualifier descriptions for the LICENSE REGISTER command describe the meaning of the PAK information. Each piece of PAK data correlates to a LICENSE REGISTER command qualifier.

Some products register their licenses during their own installation procedure. Unless you have a special circumstance, choose the registration method you prefer or the one recommended by your installation guide.

After a license is registered, it must be loaded to make it known on the current system. Section 5.1 describes the primary methods for registering and loading your licenses.

### 5.4.4 Using VMSLICENSE.COM

The following steps show how to use the VMSLICENSE.COM procedure to register a license for a product called CRYPTICALMENT. The PAK information is shown in Example 5–1.

1. Log in to the system manager's account, SYSTEM.
2. Enter the following command and press Return:

```
$ @SYS$UPDATE:VMSLICENSE
```

The procedure displays the following menu:

<sup>1</sup> With the OpenVMS operating system, you start the installation first. Although HP does not recommend it, you can install some software products first and license them later. See your software product's documentation for details.

## Using LMF

### 5.4 Registering Licenses

VMS License Management Utility Options:

1. REGISTER a Product Authorization Key
2. AMEND an existing Product Authorization Key
3. CANCEL an existing Product Authorization Key
4. LIST the Product Authorization Keys
5. MODIFY an existing Product Authorization Key
6. DISABLE an existing Product Authorization Key
7. DELETE an existing Product Authorization Key
8. COPY an existing Product Authorization Key
9. MOVE an existing Product Authorization Key
10. ENABLE an existing Product Authorization Key
11. SHOW the licenses loaded on this node
12. SHOW the unit requirements for this node
  
99. EXIT this procedure

Type '?' at any prompt for a description of the information requested. Press Ctrl/Z at any prompt to return to this menu.

Enter one of the above choices [1]

3. Enter 1. The procedure displays the following message:

\* Do you have your Product Authorization Key? [YES]:

4. Enter Y. The procedure displays the following information and prompts:

Use the REGISTER option to add a new license to a license database. A Product Authorization Key (PAK) provides the product name and information you need to register the license. You must enter all the information provided by your PAK exactly as it appears.

Issuer [DEC]:  
Authorization Number []:

5. Press Return to specify DEC<sup>1</sup>

Enter USA126087 for the authorization number that appears on the PAK. The procedure prompts for the product name:

Product Name []:

6. Enter CRYPTICALMENT for the product name string that appears on the PAK. The procedure prompts for the producer:

Producer [DEC]:

7. Press Return to specify DEC as the producer. If the product you are registering is for OpenVMS Integrity servers, your PAK will list HP as the producer. Type HP, then Return. The procedure prompts for the number of units:

Number of Units []:

8. Enter 460 for the number of units. Note that you need to enter the number of units specified on your PAK. On Integrity server systems, the number of units will be much smaller as units are counted differently (as described in Chapter 3. The procedure prompts for the version:

Version []:

---

<sup>1</sup> Although the License Management Facility software is now produced by HP, DEC is still listed as the default issuer of the license on Alpha and VAX systems. On Integrity server systems, HP is listed as the default issuer.

9. Enter 8.2 for the version number from the PAK. The procedure prompts for the key termination date:

Key Termination Date []:

10. Enter 31-DEC-2004 for the key termination date. The procedure prompts for the following information:

Availability Table Code []:

Activity Table Code []:

11. Enter E for the Availability Table Code. Press Return after the Activity Table Code prompt. The procedure prompts for the following information:

Key Options []:

Product Token []:

Hardware-Id []:

12. Enter MOD\_UNITS for the option after the Key Options prompt. Press Return after the Product Token Prompt and the Hardware-ID prompt. The procedure prompts for the checksum:

Checksum []:

13. Enter 1-BGON-IAMA-GNOL-AIKO for the checksum.

---

**Note**

---

The checksum string always begins with a number. The other 16 characters are always alphabetic characters from A through P.

---

The procedure displays the information you entered. For example:

Here is a list of the license information just entered:

```
Issuer: DEC
Authorization: USA126087
Producer: DEC
Product Name: CRYPTICALMENT
Units: 460
Release Date:
Version: 8.2
Termination Date: 31-DEC-2004
Availability: E
Activity:
Options: MOD_UNITS
Token:
Hardware ID:
Checksum: 1-BGON-IAMA-GNOL-AIKO
```

Is that correct? [YES]:

14. Compare the information on the screen with the information on the PAK. If the information is correct, enter Y.

---

**Note**

---

If you enter PAK information incorrectly, you receive an error message, and the license is not registered. A checksum error can result when you enter incorrect information for the other items on the PAK. If you get an error, carefully check all the data that you entered.

---

## Using LMF

### 5.4 Registering Licenses

If the information is incorrect, enter N.

When the procedure displays the following question, enter Y:

Do you wish to make corrections? [YES]:

15. To make corrections, the procedure asks all of the questions again but supplies the data just entered as defaults for each data field.
  - If the procedure displays correct information, press Return.
  - If the procedure displays incorrect information, enter the new data.
  - If the procedure displays incorrect information that you wish to cancel without entering new data, enter the backslash ( \ ) character.

If you entered all the information correctly, the procedure displays the following message:

```
Registering CRYPTICALMENT license in SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB...
```

If you entered some information incorrectly but did not choose YES to make corrections, the procedure may display the following message:

```
Registering CRYPTICALMENT license in SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB...
%LICENSE-F-BADCHK, checksum does not validate for CRYPTICALMENT
Please review all entered PAK data, including the checksum.
Do you wish to make corrections? [YES]:
```

To correct the data, enter Y.

If you enter an incorrect checksum string, the procedure responds as follows:

```
1-BGON-IAMA-GNOL-AIKO is not a valid license checksum string.
Press RETURN for more information
```

```
The license checksum is a 17-character verification string created by
the PAK issuer for each PAK. The checksum string is presented in
the format n-cccc-cccc-cccc-cccc, where n is an integer and c is a
character from A through P. A PAK presents the checksum string with
hyphen (-) characters for readability. Because the LMF does not count
them for authorization, you can leave them out. Otherwise, you must
enter the checksum string exactly as specified on your PAK.
```

```
If a default value is displayed and you wish to use it just press
the RETURN key. To cancel the use of default data without
entering new data, enter the backslash ( \ ) character. The
license checksum is a required field for the REGISTER and AMEND
options.
```

```
Checksum []:
```

Enter the correct checksum at the prompt.

16. After the license is successfully registered, the procedure asks if you want to load the license on the current node, as follows:

Do you want to LOAD this license on this system? [YES]:

If you registered the PAK on a standalone system and you want to make the software available (active) immediately, enter Y. If you registered the license in an OpenVMS Cluster environment but do not want to make it available (active) on the current node, enter N. After you exit this procedure, you can enter the LICENSE LOAD command to load the license on the desired node.



If you enter Y and the license is successfully loaded, the procedure displays the following informational message and prompt:

```
%LICENSE-I-LOADED, DEC CRYPTICALMENT was successfully loaded with 460 units
```

```
VMS License Management Utility Options:
```

1. REGISTER a Product Authorization Key
2. AMEND an existing Product Authorization Key
3. CANCEL an existing Product Authorization Key
4. LIST the Product Authorization Keys
5. MODIFY an existing Product Authorization Key
6. DISABLE an existing Product Authorization Key
7. DELETE an existing Product Authorization Key
8. COPY an existing Product Authorization Key
9. MOVE an existing Product Authorization Key
10. ENABLE an existing Product Authorization Key
11. SHOW the licenses loaded on this node
12. SHOW the unit requirements for this node
  
99. EXIT this procedure

```
Type '?' at any prompt for a description of the information requested. Press Ctrl/Z at any prompt to return to the main menu.
```

```
Enter one of the above choices [1]
```

17. To register another PAK, enter 1. Then respond to the questions, again entering information from a license PAK.
18. Enter 99 to exit the procedure. You have registered the license for this product. The system may display an error message when the procedure attempts to load the license. This does not affect the license registration. Exit the procedure, and read the sections of this manual that describe loading a license. For example, read the LICENSE LOAD, LICENSE UNLOAD, and LICENSE MODIFY command descriptions.

#### 5.4.4.1 Using Data Files with VMSLICENSE.COM

You can use VMSLICENSE.COM in batch mode as well as interactively. Instead of entering the PAK data interactively from your terminal or workstation, you can create a VMSLICENSE data file with an editor and enter data obtained from your product PAKs (Example 5–2 shows sample PAK data). You can then invoke VMSLICENSE.COM, specifying the name of the new VMSLICENSE data file as a parameter on the same command line. The procedure displays the license data and performs the requested operation or operations using data only from the VMSLICENSE data file.

You can create a file that registers multiple licenses. VMSLICENSE.COM exits only when it reaches the end of the VMSLICENSE data file.

#### 5.4.4.2 Invoking VMSLICENSE.COM with a VMSLICENSE Data File

To pass a VMSLICENSE data file to VMSLICENSE.COM, use the following format:

```
$ @VMSLICENSE [license-option] vmslicense-data-file [license-database]
```

When you invoke VMSLICENSE with a data file, you must specify REGISTER as the *license-option*. You cannot use data files with any of the other options that are available when using VMSLICENSE interactively.

You can specify a License Database on the command line or in the data file. Any License Database you specify in the data file overrides a License Database you specify on the command line.

## Using LMF

### 5.4 Registering Licenses

For example, you can pass the data in the sample VMSLICENSE data file, shown in Example 5–2, with the following command line:

```
$ @VMSLICENSE REGISTER NODES_A_AND_B_VMS_LICENSE.DAT
```

#### Example 5–2 Sample VMSLICENSE Data File

```
!  
! License for COBOL on NODEA  
!  
DATABASE_FILE = SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB  
ISSUER        = DEC  
PRODUCT_NAME  = COBOL  
AUTHORIZATION = USA00326  
UNITS         = 1200  
VERSION       = 7.3  
AVAILABILITY  = A  
OPTIONS       = MOD_UNITS,NO_SHARE  
CHECKSUM      = 1-DELM-EAHF-ONIH-MBAH  
INCLUDE_NODE  = NODEA  
!  
<NEXT>  
!  
! License for COBOL on NODEB  
!  
AUTHORIZATION = USA00327  
UNITS         = 800  
CHECKSUM      = 1-PATC-IDOH-EPOF-MOHG  
INCLUDE_NODE  = NODEB  
!
```

#### 5.4.4.3 Creating VMSLICENSE Data Files

When you create a VMSLICENSE data file to be processed by VMSLICENSE.COM, you must follow these rules (refer to Example 5–2):

- Specify all PAK data as parameters in the form of DCL assignment statements. Use the following format:

```
parameter = parameter-value ! comment
```

Table 5–1 lists the parameters allowed in the VMSLICENSE data file. You must use the exact parameter names, or VMSLICENSE ignores the line in the VMSLICENSE data file and returns an error message.

- Separate the end of one license registration from the beginning of another with the following special license data separator:  
<NEXT>
- Precede each comment with an exclamation point; VMSLICENSE ignores everything to the right of an exclamation point when processing the line.
- Separate words in the VMSLICENSE data file with any number of spaces or tabs.
- List parameters in any order in the VMSLICENSE data file.

**Table 5–1 Parameters Used with VMSLICENSE.COM**

Parameter	Description
ACTIVITY	License unit code that corresponds to a License Unit Requirement Table (LURT) or to a constant value.
AUTHORIZATION	String that, when used with the PAK issuer string, identifies the license you want to register.
AVAILABILITY	License unit code that corresponds to a LURT value or to a constant value.
CHECKSUM	Verification string of 17 characters created by the PAK issuer for each PAK with this format: <i>n-cccc-cccc-cccc-cccc</i> where: <i>n</i> is an integer and <i>c</i> is an alphabetic character from A through P.
DATABASE_FILE	Location of the License Database to be used. The default file specification is defined by the logical LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system.
DATE	Product date specifying that the license authorizes use of all product versions released on or before the date.
HARDWARE_ID	Identification number of the hardware on which the product is licensed. On Integrity server systems, this field is used to specify the number of SOCKETS. If your PAK does not list SOCKETS= <i>n</i> in the HARDWARE_ID field, then your license is unlimited.
INCLUDE_NODE	Nodes in an OpenVMS Cluster environment that can access the licensed product. INCLUDE_NODE can specify one or more nodes in a clustered Galaxy.
ISSUER	Company that issued the PAK for this product.
OPTIONS	List of license options from a PAK.
PRODUCER	Company that owns this product.
PRODUCT_NAME	Product with a license to register. Use the product name exactly as it appears on the PAK.
RESERVE_LIST	Users (or activities) allowed to access the product license.
TERMINATION_DATE	Date on which the product license is no longer valid.
TOKEN	Product token string from a PAK.
UNITS	Number of license units from a PAK.
VERSION	Version limits from a PAK of the product for which you have a license.

#### 5.4.4.4 Using VMSLICENSE.COM Default Value Rules

If you do not specify a value for a parameter in a VMSLICENSE data file, VMSLICENSE substitutes default values. The current default values are ISSUER=DEC and PRODUCER=DEC on OpenVMS Alpha and VAX systems and HP on OpenVMS Integrity server systems. All other license parameters have null values until you specify a value.

After you specify a license parameter in the VMSLICENSE data file, the parameter becomes the new default value until the next occurrence of the parameter sets a new default value. To set or reset the value of a parameter to null, use a line of the following form:

```
parameter = " "
```

## Using LMF

### 5.4 Registering Licenses

For example, if the most recent PAK data set the INCLUDE\_NODE parameter to a specific node, reset the parameter to null for the current and subsequent PAKs by entering the following:

```
INCLUDE_NODE = ""
```

Example 5–2 shows how using default data can eliminate typing for the second PAK listed in the VMSLICENSE data file.

#### 5.4.5 Using the LICENSE REGISTER Command

You can directly register (and load) a license with LICENSE commands. For example:

```
$ LICENSE REGISTER CRYPTICALMENT /ISSUER=DEC -  
  $ /AUTHORIZATION=USA126087 /PRODUCER=DEC /UNITS=460 -  
  $ /VERSION=7.3 /TERMINATION DATE=31-DEC-2001 /AVAILABILITY=E -  
  $ /OPTIONS=(MOD_UNITS) /CHECKSUM=1-BGON-IAMA-GNOL-AIKO  
$ LICENSE LOAD CRYPTICALMENT  
LICENSE-I-LOADED, DEC CRYPTICALMENT was successfully loaded with 460 units  
$
```

After you successfully register a license in the License Database (the default file specification is SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB), you can enter the LICENSE LIST/FULL command to display the data in the database (see Example 5–3).

#### Example 5–3 Displaying License Database Information

```
$ LICENSE LIST /FULL /AUTHORIZATION=USA126087 CRYPTICALMENT
```

Use CTRL/Z to exit, PF3-PF4 for Previous-Next Screen and Arrow Keys to scroll.  
License Management Facility V1.2

```
License Database File:  SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB  
Created on:            28-AUG-2001  
Created by user:      GRAHAM  
Created by LMF Version: V1.2
```

```
-----  
Issuer:                DEC  
Authorization:         USA126087  
Product Name:         CRYPTICALMENT  
Producer:             DEC  
Units:                460  
Version:              7.3  
Release Date:         (none)  
PAK Termination Date: 31-DEC-2001  
Availability:         E (System Integrated Products)  
Activity:             0  
Options:              MOD_UNITS  
Hardware ID:  
  
Revision Level:       1  
Status:              Active  
Command:             REGISTER  
Modified by user:    GRAHAM  
Modified on:         23-SEP-2001 14:32:23.41
```

When a license is successfully loaded with VMSLICENSE.COM or the LICENSE LOAD command, LMF displays an informational message.

### 5.4.6 Displaying License Information

To display information from the License Database about specific licenses, enter the LICENSE LIST command.

To display information in memory about loaded licenses, enter the SHOW LICENSE command. This command displays licenses loaded on the current node, and displays any reservation list associated with each license.

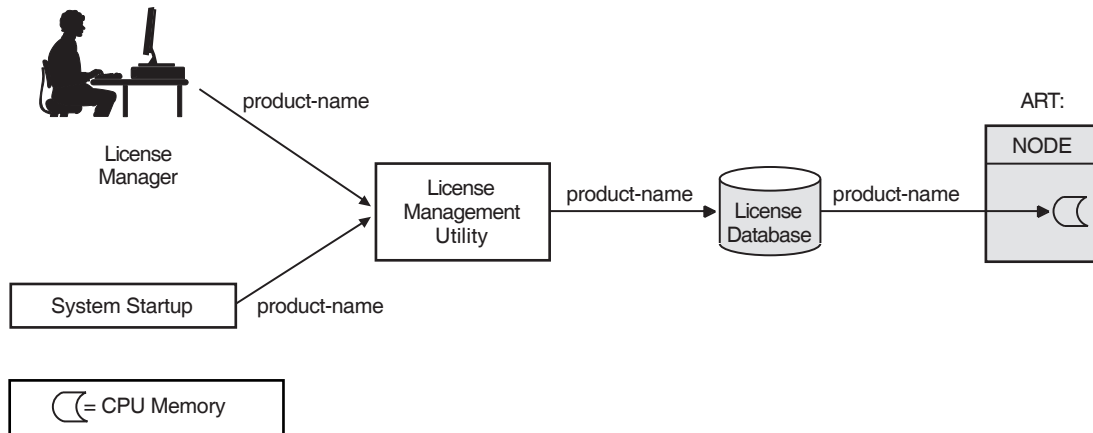
## 5.5 Loading a License

To allow users to access a product, you must load each registered license. Loading a license transfers data from the on-disk License Database into system memory. The following table shows the methods of license loading for LMF:

License Loading Options	
Status of Registered License	Loading Method
Registering with VMSLICENSE.COM	Confirm the load option.
Registered with the LICENSE REGISTER command	Use the LICENSE LOAD command.
Previously registered system starting up	System loads the license automatically.

Figure 5–4 illustrates the license loading process for a standalone system. Whether the license manager enters the LICENSE LOAD command or the system automatically loads all registered licenses during startup, the license information is transferred to the system memory of node ART.

Figure 5–4 Loading a License



ZK-7457-AI

### 5.5.1 Loading Licenses in an OpenVMS Cluster Environment

In an OpenVMS Cluster environment, multiple nodes load licenses from a single common database. LMF provides this capability to let you share licenses as well as control access to a product on a node-by-node basis (provided this is allowed by the terms and conditions of the license).

## Using LMF

### 5.5 Loading a License

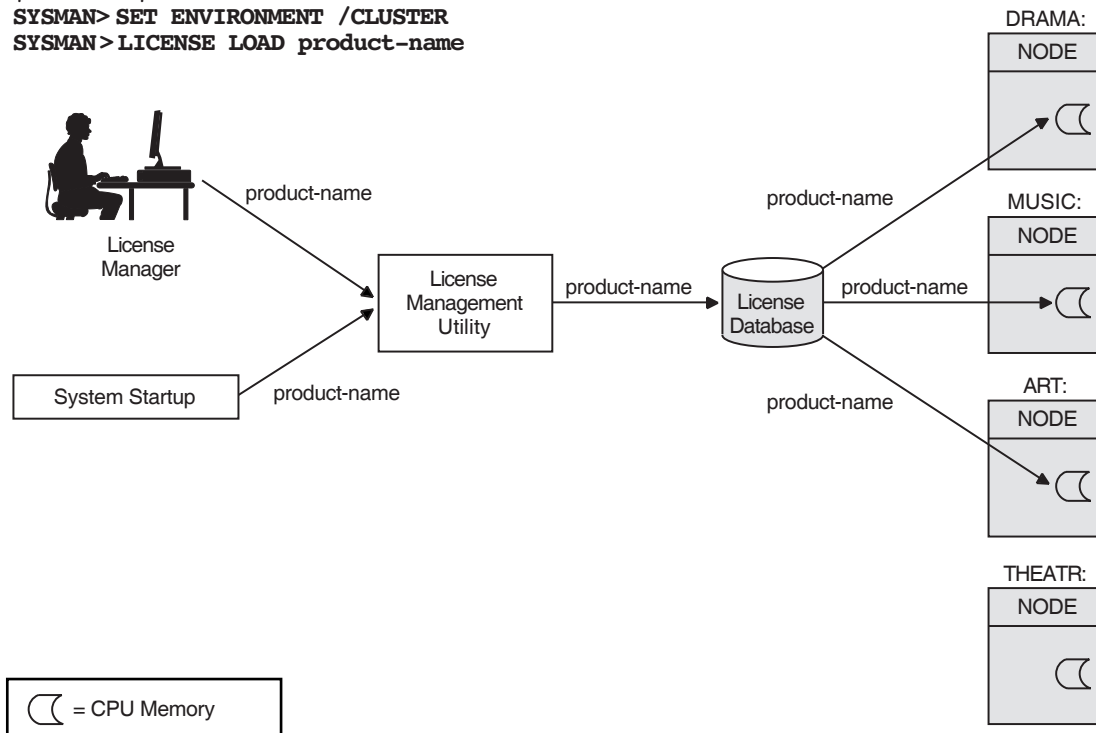
Typically, any node can load a license registered in the common License Database. For predictability, security, performance, or economic reasons, you can limit which nodes can access a product license intended to be shared among nodes. In an OpenVMS Cluster environment, you control access by modifying the license's **include list**. Use the LICENSE MODIFY command and either the /INCLUDE or /EXCLUDE qualifier to specify which cluster nodes can load the license. By changing the include list, you can create the effect of moving a product from node to node. Section 5.6.2 describes the process in detail. Note that you cannot share licenses that have the NO\_SHARE option. They must be assigned to a single node.

You can also control which users in an OpenVMS Cluster environment can access a product license. Control access by modifying the license's reservation list. Use the LICENSE MODIFY command with the /RESERVE qualifier to specify which users can access the license.

In an OpenVMS Cluster environment, license loading involves transferring, or loading, license information from the on-disk License Database into system memory of the current node. The number and combination of licenses for an OpenVMS Cluster environment can be complex, which in turn makes the loading process complex and sometimes confusing. The general rule is that nodes specified on include lists or not specified on exclude lists of the LICENSE MODIFY command can load a registered license. As each system starts up, it loads any licenses to which it has access. If you need to load a license for all assigned nodes of a running cluster, you can use the OpenVMS System Management utility (SYSMAN), which is described in the *HP OpenVMS System Manager's Manual*. Figure 5-5 illustrates the loading process in an OpenVMS Cluster environment.

Figure 5–5 Loading a License in an OpenVMS Cluster Environment

```
$ LICENSE MODIFY /EXCLUDE=THEATR product-name
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT /CLUSTER
SYSMAN> LICENSE LOAD product-name
```



ZK-7456-AI

See Section 5.6.2 for further details of license loading.

## 5.6 Managing Licenses After Registration

After you register licenses, you can manage them. While system managers can perform these tasks, managers of cluster environments and large multiple-user systems may also have an interest in managing products and product licenses.

For example, a cluster manager can use LMF to control which nodes can access a database product using important data (see Section 5.6.2). All system managers, however, may want to monitor license data using LICENSE commands.

VMSLICENSE.COM, LICENSE commands, and the DCL command SHOW LICENSE can perform most license management tasks, including:

- Listing registered licenses
- Showing loaded licenses
- Making loaded licenses unavailable — either permanently or temporarily
- Transferring licenses from one database to another
- Modifying licenses — such as termination dates or reservation lists

## Using LMF

### 5.6 Managing Licenses After Registration

#### 5.6.1 Restricting Product Use

LMF provides the following commands and qualifiers for controlling access to licensed products.

##### 5.6.1.1 Fastest Method

The quickest method to restrict access to products registered in the License Database is to unload the license with a LICENSE UNLOAD command. The product becomes unavailable to new users and unavailable to all users when the last process using the product finishes. The product remains inaccessible until either you or the system reloads the license.

Although LICENSE UNLOAD is fast, it is not permanent. Typically, at system startup, LMF loads all the licenses registered in the License Database. In addition, a license that is loaded remains loaded until either the system is shut down or you intervene.

##### 5.6.1.2 For Inactive Licenses

The following commands control license loading, which restricts product access. However, they do not unload loaded licenses or alter in-memory license data. For loaded licenses, use LICENSE UNLOAD.

- LICENSE DISABLE  
Prevents the specified license from being loaded until you enter a LICENSE ENABLE command.
- LICENSE MODIFY/INCLUDE, LICENSE MODIFY/EXCLUDE  
Makes the named nodes in an OpenVMS Cluster environment available or unavailable for license loading.
- LICENSE MODIFY/UNITS=*number*  
Limits access when you set the unit number to be less than that required by your processor or cluster.  
Use if your license has the MOD\_UNITS option. Not all licenses have a specific number of units. Some licenses provide zero units, which is equivalent to unlimited units.
- LICENSE MODIFY/RESERVE=*user-name*  
Makes the product available only to the users named in the reservation list. You must use this method if your license has the RESERVE\_UNITS option.
- LICENSE COPY  
Automatically disables a license as you reregister it in another License Database.  
This command is similar in effect to LICENSE DISABLE except that it also registers the license in the other database.
- LICENSE ISSUE  
Automatically disables a license so that you can then reregister it in another database.  
This command is similar in effect to LICENSE DISABLE except that it also produces a PAK replica to be registered elsewhere.



## 5.6 Managing Licenses After Registration

### 5.6.1.3 Permanent Methods

For more permanent access restrictions, issue one of the following commands:

- LICENSE DELETE

Permanently deletes one or more licenses from a database, disallowing further license use. Use this command to eliminate terminated or unused licenses.

- LICENSE MOVE

Automatically deletes a license as you reregister the license in another License Database.

This command is similar in effect to LICENSE DELETE except that it also registers the license in the other database.

### 5.6.2 Controlling Node Access to Licenses in Clusters

In an OpenVMS Cluster environment, you can control which nodes have access to a product. Some product licenses require you to control access by creating include or exclude lists with the LICENSE MODIFY command.

In an OpenVMS Cluster environment, license units are available to all nodes by default. If you need to control which nodes in a cluster have access to a product license, you must use the LICENSE MODIFY command with the /INCLUDE or /EXCLUDE qualifiers. These qualifiers take an argument list of System Communications Services (SCS) node names. SCS node names are system parameters set with the System Generation utility (SYSGEN). For example, if your cluster includes nodes MUSIC, ART, DANCE, and THEATR, you can specify that nodes MUSIC and ART have access to the software products registered in the License Database, while nodes DANCE and THEATR do not have access. The following command allows two nodes access to Pascal:

```
$ LICENSE MODIFY/INCLUDE=(MUSIC,ART) PASCAL
```

You can perform the same task by using the /EXCLUDE qualifier. The following command specifies the same product access as the previous command:

```
$ LICENSE MODIFY/EXCLUDE=(DANCE,THEATR) PASCAL
```

If a license does not provide enough license units for full availability across the cluster, use LICENSE MODIFY and one of these qualifiers. Otherwise, product availability is unpredictable. Products are authorized for use on nodes in the order in which they load the licenses. Unless you use an include list to control which nodes can load a product, the authorization to use a product can move from processor to processor during a series of system shutdowns and startups. When you shut down a system, LMF automatically unloads all loaded licenses on that system. If another cluster member boots before the first system reboots, the second system, referring to the common License Database, can automatically load the same license. Although this can be helpful, it may not be your intention.

You can use the /INCLUDE and /EXCLUDE qualifiers with the LICENSE MODIFY command to determine who has access to the pool of units created by LMF when it loads combinable licenses. However, note the following:

- LMF combines units from combinable licenses (licenses with the same product name and type) into a single pool from which all authorized nodes may draw. By default, all nodes are authorized. To restrict the nodes that are authorized, assign an include or exclude list to each license for the product.

## Using LMF

### 5.6 Managing Licenses After Registration

- Adding the NO\_SHARE option to PAKs prevents license units from being shared. You can use this option to reserve all the units on a PAK for a particular node in an OpenVMS Cluster.
- LMF unionizes all the include and exclude lists associated with combinable licenses. The resulting master list is all-inclusive; when combining licenses, a less restrictive list always overrides a more restrictive list.
- Assigning an include list to a license does not reserve the license units for the nodes in the list.

For example, if you assign an include list to four out of five combinable licenses, the default—all nodes are authorized—is in effect for the fifth license, and it overrides the lists on the other licenses. All nodes then have access to the product despite the include lists. Units for the product are allocated on a first-come, first-served basis as the nodes in the cluster are booted, until the pool of units is depleted.

To ensure access exactly as you intend it, assign the same include or exclude list to each license. Purchasing more license units to fulfill requirements to run across the cluster is another option.

#### 5.6.2.1 Effect of the NO\_SHARE Option

Some licenses, such as OpenVMS operating system licenses, have the NO\_SHARE option; they cannot be shared in a cluster environment. If you are using a common License Database, you must register a separate license for each cluster node and modify each to establish which node can load it.

To ensure that LMF can load a NO\_SHARE license in a cluster environment, you have two choices, as follows:

- When you register with VMSLICENSE.COM, you are prompted for a node name. Enter the correct SCS node name at the prompt.
- If you use LICENSE REGISTER, you must follow up by entering a LICENSE MODIFY/INCLUDE=*node-name* command.

A cluster environment typically uses a single License Database, which should include one OpenVMS license for each cluster node. You can change the association of license and node as long as the number, type, and size of the licenses match the processors present. For example, the cluster environment with nodes ART, MUSIC, DANCE, and THEATR should have four licenses, each one designated for a specific node. If you remove node THEATR and replace it with a node named CRAFTS, you must modify the license intended for THEATR to specify node CRAFTS.

Because an OpenVMS Cluster License Database often contains multiple licenses with the identical product name and producer, you should use the /AUTHORIZATION qualifier with LICENSE commands to uniquely identify a specific license. For example:

```
$ LICENSE MODIFY/INCLUDE=THEATR BASIC /AUTHORIZATION=USA123456
.
.
.
$ LICENSE MODIFY/INCLUDE=CRAFTS COBOL
%LICENSE-E-AMBIG, information provided was ambiguous;
multiple licenses were found
$ LICENSE MODIFY/INCLUDE=CRAFTS COBOL /AUTHORIZATION=USA123456
```

## 5.6 Managing Licenses After Registration

## 5.6.2.2 Editing Include and Exclude Lists

Each time you enter a LICENSE MODIFY command with an /INCLUDE or /EXCLUDE qualifier, LMF creates a new list. To edit an existing list, use the /ADD or /REMOVE qualifier in your command line. The following example illustrates the required syntax without using the /ADD or /REMOVE qualifier:

```
$ LICENSE MODIFY/INCLUDE=(ART,MUSIC,DANCE,THEATR) BASIC -
_ $ /AUTHORIZATION=USA123456
.
.
.
$ LICENSE MODIFY/INCLUDE=(ART,MUSIC,DANCE,CRAFTS) BASIC -
_ $ /AUTHORIZATION=USA123456
$ LICENSE UNLOAD BASIC
%LICENSE-I-LOADED, DEC BASIC was successfully loaded with 400 units
```

You can also use the following commands:

```
$ LICENSE MODIFY/INCLUDE=(ART,MUSIC,DANCE,THEATR) BASIC -
_ $ /AUTHORIZATION=USA123456
.
.
.
$ LICENSE MODIFY/REMOVE/INCLUDE=(THEATR) BASIC /AUTHORIZATION=USA123456
$ LICENSE MODIFY/ADD/INCLUDE=(CRAFTS) BASIC /AUTHORIZATION=USA123456
```

If your license uses the MOD\_UNITS option, you can also modify the size of a license in a cluster environment. To change the size of the license, enter a LICENSE MODIFY/UNITS=*number* command that specifies a number sufficient for your needs and allowed by your license agreement. For example, to change a license registered with 1000 license units to a 1500-unit license, enter:

```
$ LICENSE MODIFY/UNITS=1500 BASIC
$ LICENSE LOAD BASIC
```

---

**Note**


---

You can use the LICENSE MODIFY/UNITS command to increase the number of license units within the terms and conditions of your license agreement. If you need more license units than the number currently allowed by your license agreement, please contact your HP representative for assistance in upgrading your license.

---

## 5.6.3 Controlling User Access

To control which users have access to a product, use the LICENSE MODIFY command with the /RESERVE qualifier. This qualifier takes an argument list of user names called the reservation list. Although the definition of a user can differ from product to product, most products accept the user name that OpenVMS maintains for each account. This is the name you type at the Username prompt during login. See your product's Software Product Description (SPD) for details.

If your PAK specifies the RESERVE\_UNITS option, you *must* assign one or more users to a reservation list. The number of user names allowed per list depends on the number of activity units available. Calculate this number as you would for any activity license. For example, if a software product requires 50 license units per activity and your PAK provides 100 license units, you have a 2-activity license. If the PAK also specifies the RESERVE\_UNITS option, you

## Using LMF

### 5.6 Managing Licenses After Registration

have an unlimited activity, two-user license. For this license, you must create a reservation list with at least one, but no more than two, names.

**Example:** The following command assigns two users to a reservation list for the product called Terrapin:

```
$ LICENSE MODIFY /RESERVE=(R_HUNTER,J_BARLOW) TERRAPIN
```

Note that the LICENSE MODIFY command affects only data in the license database; it does not affect licenses already loaded. To change a loaded license, reload it with a LICENSE LOAD command. For example:

```
$ LICENSE MODIFY /RESERVE=(R_HUNTER,J_BARLOW) TERRAPIN
$ LICENSE LOAD TERRAPIN
%LICENSE-I-UNLOADED
LICENSE-I-LOADED, DEC TERRAPIN was successfully loaded with 200 units
```

To add more user names to the reservation list, use the /ADD qualifier and the /RESERVE qualifier, as follows:

```
$ LICENSE MODIFY /ADD /RESERVE=(P_LESH,M_HART) TERRAPIN
```

This adds new users P\_LESH and M\_HART to any list already established for the specified product. You can remove a user name with the /REMOVE qualifier.

---

#### Note

---

LMF does not restrict you from creating incorrect reservation lists. If a user on a reservation list is being denied access to a product, check the reservation list (or reservation lists with multiple licenses for the same product) for the following:

- Too many names. If you repeat a user name, LMF can reject a valid user name entry after reaching the allowed number of users for the license. LMF provides a warning when it loads a license with a list that is too long.
  - Incorrect spelling of user names. LMF simply compares and counts user names. If you misspell a name in the reservation list, LMF denies access to the user trying to access a product. LMF still counts each misspelled name as a potential user.
- 

You can have many Personal Use Licenses for the same product. For license loading, LMF combines all of the license units and determines the number of users according to the total number of units. Therefore, the total number of names on combined reservation lists for this product must not exceed the number LMF authorizes from the unit count, because LMF authorizes only the correct number from the lists and rejects extra names.

Although you can find extra or repeated names using one or more LICENSE LIST/FULL commands, you cannot easily predict which users LMF will reject. Do not assume that LMF denies access to the third name listed on the reservation list of a two-user license. The total number of names and total number of license units is the important calculation.

To load corrections to a reservation list you must enter the LICENSE LOAD command for the licenses. The following example includes the warning message for too many names:

## 5.6 Managing Licenses After Registration

```

$ LICENSE MODIFY/RESERVE=(R_HUNTER,J_BARLOW) TERRAPIN
$ LICENSE LOAD TERRAPIN
LICENSE-I-LOADED, DEC TERRAPIN was successfully loaded with 200 units
$ LICENSE MODIFY/ADD/RESERVE=(P_LESH,M_HART) TERRAPIN
$ LICENSE LOAD TERRAPIN
%LICENSE-I-UNLOADED
LICENSE-W-LONGLIST, reserve list for DEC TERRAPIN exceeds maximum of 2, 2 names removed
LICENSE-I-LOADED, DEC TERRAPIN was successfully loaded with 200 units

```

Because LMF combines the license units, you can assign one long reservation list to a single license; LMF simply adds the license units from the combinable licenses and counts the names in all reservation lists for those licenses. If you have three combinable licenses that authorize use to six users, you can modify one of them to have a 6-name reservation list. Note that this differs from the behavior of include and exclude lists with node names in an OpenVMS Cluster.

### 5.6.4 Controlling Loading Order

If you have many variations of licenses for a product (for example, with different versions, tokens, or hardware identifiers), and you think that you are not getting maximum use of the product, you can check the order of loading of the product licenses. By default, LMF assigns a selection weight to each license and loads each license in descending order of selection weight. The grant order is the order in which LMF checks licenses before granting one.

Loading is the process that LMF uses to store licenses in memory. Granting is the actual allocation of units to a user using a licensed product. Selection weights control the order in which LMF checks multiple licenses for a single product while attempting to perform a license grant for that product.

To determine grant order, enter the DCL command `SHOW LICENSE/FULL`. You can then enter the `LICENSE LIST` command with the `/SELECTION_WEIGHT` qualifier to display the selection weight. Modify selection weights of licenses as needed with the `/SELECTION_WEIGHT` qualifier to the `LICENSE MODIFY` command.

To change the selection weight, use `LICENSE MODIFY/SELECTION_WEIGHT`, and then load the changed licenses with `LICENSE LOAD`.

## 5.7 Logical Name LMF\$DISPLAY\_OPCOM\_MESSAGE

A previous version of this manual incorrectly stated that you must define the logical name `LMF$DISPLAY_OPCOM_MESSAGE` in order for messages to appear.

If you have already defined this logical name, you should delete the definition.

Define the `LMF$DISPLAY_OPCOM_MESSAGE` logical name only if you are explicitly directed by HP to do so (for debugging purposes). When defined, this logical name causes many “noise” messages to be displayed on the operator’s console. Some of these messages can be confusing and misleading to the point of suggesting that a product is not licensed when in fact it is.

To see if this logical name has been defined on your system, enter the following command:

```
$ SHOW LOGICAL LMF$DISPLAY_OPCOM_MESSAGE
```

To delete the logical name, enter the following command:

```
$ DEASSIGN LMF$DISPLAY_OPCOM_MESSAGE/EXEC/SYSTEM
```

## Using LMF

### 5.8 Troubleshooting Licensing Problems

### 5.8 Troubleshooting Licensing Problems

If you are having problems that appear to be related to reaching PAK limits or missing licenses, you can perform basic troubleshooting using the `LICENSE` and `SHOW LICENSE` commands.

First, try listing the PAKs using the following command:

```
$ LICENSE LIST /FULL
```

This command will list all the PAKs that are in the License Database (.LDB) file. These licenses may or may not have been loaded into the memory license database. Check to make sure that all your active licenses have been loaded and that any unused licenses are not being loaded into the License Database.

Next, use the `/HISTORY` qualifier to check licensing activity.

```
$ LICENSE LIST /HISTORY
```

This command shows you all the activity you have performed to the License Database. Make sure that the history matches what you think should be.

You can also use the `SHOW LICENSE` command to see if the number of licenses is correct. The `/UNIT_REQUIREMENTS` command displays the information in the License Unit Requirement Table.

```
$ SHOW LICENSE /UNIT_REQUIREMENTS
```

Use the `/CLUSTER` qualifier to display the license unit requirements for every node in an OpenVMS cluster.

Use the `SHOW LICENSE/USAGE` command to see how many license units are loaded and how many are allocated and available. `SHOW LICENSE/USAGE` also tells you the license type for each product on the system.

If you own multiple license types for a single product in an OpenVMS cluster, you can only view the usage information for the license type loaded on the node from which you issued the `SHOW LICENSE/USAGE` command. To find out the usage of another license type loaded on another node, issue the command on that node.

In an OpenVMS Cluster, usage information is limited to the local license type. For example, LMF considers VAX and Alpha availability licenses different license types. If you are running both VAX and Alpha systems in a cluster, usage information for availability licenses is limited to the local system type. For example, if you have HP C installed on all nodes in your OpenVMS Cluster, you can display HP C license allocation on all the VAX nodes in the cluster from any VAX node with HP C installed, but you cannot display the HP C license allocation on the Alpha nodes.

---

# OpenVMS Galaxy Licensing Information (Alpha Only)

The OpenVMS Galaxy Software Architecture on OpenVMS (OpenVMS Galaxy) is a system integrated product (SIP). That is, OpenVMS Galaxy code is integrated and delivered with the OpenVMS operating system.

The License Management Facility (LMF) Product Authorization Keys (PAKs) representing OpenVMS Galaxy licenses allow you to access and use OpenVMS Galaxy software.

## 6.1 OpenVMS Galaxy Licensing Requirements

The following list summarizes OpenVMS Galaxy licensing requirements:

- One OpenVMS Operating System License for a Galaxy system
- One SMP Extension License for each CPU after the first CPU
- One OpenVMS Galaxy License for each CPU in a Galaxy system
- No changes to how HP layered products are licensed:
  - One capacity license per system
  - One user license per use

The following sections describe these requirements in more detail.

### 6.1.1 OpenVMS Operating System License

When an AlphaServer system is configured as an OpenVMS Galaxy system, there are no changes in how a system is licensed for the OpenVMS operating system.

One OpenVMS Base License is required for the Galaxy system, plus one SMP Extension License for each CPU after the first CPU.

### 6.1.2 OpenVMS Galaxy License

To create and run multiple instances, one OpenVMS Galaxy License is required for each CPU in a Galaxy system.

License rights for running a single-instance Galaxy on any Alpha system are provided by the OpenVMS Base License.

## OpenVMS Galaxy Licensing Information (Alpha Only)

### 6.1 OpenVMS Galaxy Licensing Requirements

#### 6.1.3 OpenVMS Layered Products License

HP software layered products on OpenVMS Galaxy configurations continue to use standard license types: Traditional, Concurrent Use, and Personal Use.

- One Traditional Capacity License will continue to license the system, regardless of the number of instances. The license is based on the system class of the hardware system.
- Concurrent Use Licenses will continue to license one concurrent use of the product.
- Personal Use Licenses will continue to license one named user on the system.

### 6.2 Clustering OpenVMS Galaxy Instances

Instances in an OpenVMS Galaxy computing environment can be clustered with other instances in a single system, with instances in other Galaxy systems, or with non-Galaxy systems. Each type of clustering has different licensing requirements, as described in the following sections.

#### 6.2.1 Clustering in a Galaxy System

In an OpenVMS Galaxy computing environment, instances can be clustered with other instances within a Galaxy system. Clustered instances use the shared-memory cluster interconnect to communicate with each other.

The licensing and functionality for clustering within a Galaxy system is provided under the OpenVMS Galaxy License.

#### 6.2.2 Clustering Outside a Galaxy System

Instances in an OpenVMS Galaxy computing environment can be clustered with instances in another OpenVMS Galaxy system or with cluster nodes in non-Galaxy systems. Instances clustered outside of a Galaxy system use traditional cluster interconnects.

Each system that is clustered with another system must be licensed for OpenVMS Cluster Software. Clustering outside the OpenVMS Galaxy system is not covered by the OpenVMS Galaxy License.

### 6.3 License Databases

When an OpenVMS Galaxy system is configured with more than one instance, a license database must be set up for each independent instance or cluster of instances. The PAKs representing the licenses on the OpenVMS Galaxy configuration can be loaded on multiple license databases, as follows:

- OpenVMS Base, SMP Extensions, and OpenVMS Galaxy Licenses  
For full use of the OpenVMS Galaxy functionality, the PAKs representing your OpenVMS Base, SMP Extensions, and OpenVMS Galaxy Licenses must be loaded on each license database within the OpenVMS Galaxy on a single system.
- HP Layered Product Traditional Capacity Licenses  
The PAKs representing these licenses can be loaded on each license database within the OpenVMS Galaxy on a single system.
- HP Concurrent Use Licenses



## OpenVMS Galaxy Licensing Information (Alpha Only)

### 6.3 License Databases

The PAKs representing these licenses can be loaded on the license database where the license will be used.

For PAKs representing multiple concurrent uses (for example, a 50-use license) the PAK can be loaded on multiple license databases within the OpenVMS Galaxy on a single system. However, the PAK must be modified down on each license database so that the total number of license units does not exceed the total license units on the License PAK.

- HP Personal Use Licenses

The PAKs representing these licenses can be loaded on each license database within the OpenVMS Galaxy on a single system; however, the number of actual uses must not exceed the licensed number of personal uses.

### 6.4 OpenVMS Galaxy License PAKs and LMF

OpenVMS Galaxy PAK names are as follows:

- OpenVMS Operating System Base PAK: OPENVMS-ALPHA (base license)
- OpenVMS Operating System User PAK: OPENVMS-ALPHA-USER
- OpenVMS Galaxy PAK: OPENVMS-GALAXY

OpenVMS Galaxy customers must have at least one OPENVMS-ALPHA PAK, plus one additional OPENVMS-ALPHA PAK for each additional processor (CPU) after the first CPU (which is included in the Base Operating System License).

The OPENVMS-ALPHA and OPENVMS-ALPHA-USER PAKs can now be shared by multiple Galaxy instances. To implement this in the License Management Facility (LMF), include all OpenVMS Galaxy instance names in the PAK INCLUDE list.

For example, suppose that a customer has a system named ANDA1A in an OpenVMS Cluster. The OPENVMS-ALPHA license PAK currently has an INCLUDE list on it that has SCS node name ANDA1A in it. If that system is changed to an OpenVMS Galaxy running three instances named ANDA1A, ANDA2A, and ANDA3A, the OPENVMS-ALPHA license PAK must be modified so that all instances can share the NO\_SHARE OPENVMS-ALPHA license.

The command to modify the OPENVMS-ALPHA license PAK is:

```
$ LICENSE MODIFY OPENVMS-ALPHA/AUTHORIZATION=xxxxx -  
_ $ /INCLUDE=(ANDA1A,ANDA2A,ANDA3A)
```

Because this example assumes that ANDA1A was already in a cluster, the authorization number is required to identify the one PAK of many OPENVMS-ALPHA license PAKs in the License Database (.LDB) file.



# A

---

## Command Reference

This appendix describes the syntax of the following License Management utility (LICENSE) and SHOW LICENSE commands:

- LICENSE COPY
- LICENSE CREATE
- LICENSE DELETE
- LICENSE DISABLE
- LICENSE ENABLE
- LICENSE ISSUE
- LICENSE LIST
- LICENSE LOAD
- LICENSE MODIFY
- LICENSE MOVE
- LICENSE REGISTER
- LICENSE START
- LICENSE UNLOAD
- SHOW LICENSE

## LICENSE

---

### LICENSE COPY

Copies licenses from one License Database to another. When you use LICENSE COPY, LMF disables the source license and registers a copy in the destination License Database as if it were a new license. If the terms and conditions of your license contract allow it, you can re-enable the source database license by using LICENSE ENABLE.

LICENSE COPY cannot be used to create a copy of a license in the same database as the source of the copy.

### Format

```
LICENSE COPY product-name[...] output-database
```

### Parameters

#### **product-name[,...]**

Name or names of products with a license to be copied to the output License Database.

#### **output-database**

File specification of the License Database to which the license or licenses should be copied. This database must have been created previously using LICENSE CREATE.

If you enter a partial file specification (for example, specifying only a directory), LMF\$LICENSE is the default file name, and .LDB is the default file type. If you do not specify a device or directory, the current default device and directory are used.

### Qualifiers

#### **/ALL**

Positional qualifier.

Specifies that all licenses with the given product name should be copied. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/AUTHORIZATION=string**

Positional qualifier.

Specifies a string that helps identify the license you want to copy. You must enter the authorization string exactly as it appears on your PAK. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/DATABASE=filespec**

Specifies the location of the License Database from which the license should be copied. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default License Database name and location.

**/ISSUER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that issued the PAK for the product. Use this optional qualifier only if you need to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

**/LOG**

**/NOLOG (default)**

Controls whether LICENSE COPY displays the name of each license that it copies.

**/PRODUCER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that owns the product for which you have a license. Use this optional qualifier only if you need to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

## Description

To copy a license from one database to another, use LICENSE COPY. The following conditions apply to a LICENSE COPY transaction:

- The status of the source database license changes to disabled.
- Network copies are supported within the limits of remote FAL access. If you use access control strings, such as “USERNAME password” within the file specification, the actual password string is not stored.
- LICENSE COPY does not transfer any user-supplied data such as reservation lists, modified termination dates, modified units, include or exclude node lists, or comments.

## Examples

1. `$ LICENSE COPY FORTRAN BACKUP_DATA:BACKUP.LDB`

This command copies the Fortran license in the default License Database to the BACKUP\_DATA:BACKUP.LDB License Database. This command fails if there is more than one Fortran license in the default database.

2. `$ LICENSE COPY FORTRAN /DATABASE=BACKUP_DATA:BACKUP.LDB -  
_ $ BACKUP_DATA2:BACKUP2.LDB`

This command copies the Fortran license in the source License Database to the BACKUP\_DATA2:BACKUP2.LDB License Database. This command fails if there is more than one Fortran license in the source database.

3. `$ LICENSE COPY FORTRAN /ALL BACKUP_DATA:BACKUP.LDB`

This command copies all Fortran licenses in the default License Database to the BACKUP\_DATA:BACKUP.LDB License Database.

## LICENSE LICENSE COPY

4. \$ LICENSE COPY FOR\* BACKUP\_DATA:BACKUP.LDB

This command copies all licenses whose product names begin with the string "FOR" from the default License Database to the BACKUP\_DATA:BACKUP.LDB License Database. In this case, using the wildcard character (\*) implies the use of /ALL.

5. \$ LICENSE COPY \* BACKUP\_DATA:BACKUP.LDB

This command copies all licenses from the default License Database to the BACKUP\_DATA:BACKUP.LDB License Database. In this case, using the wildcard character (\*) implies the use of /ALL.

6. \$ LICENSE COPY \* /PRODUCER=DEC BACKUP\_DATA:BACKUP.LDB

This command copies all licenses with the producer name DEC from the default License Database to the BACKUP\_DATA:BACKUP.LDB License Database. In this case, using the wildcard character (\*) implies the use of /ALL.

7. \$ LICENSE COPY D%% BACKUP\_DATA:BACKUP.LDB

This command copies all licenses beginning with a "D" followed by exactly two characters from the default License Database to the BACKUP\_DATA:BACKUP.LDB License Database. In this case, using the wildcard character (%) implies the use of /ALL.

---

## LICENSE CREATE

Creates a License Database with no license records. Because LMF provides a default License Database in `SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB` when OpenVMS is installed, you do not typically need to use this command.

To use LMF, you must have a License Database file and the appropriate number of units for your system. On OpenVMS Alpha and VAX, the units are located in the License Unit Requirement Table (LURT) file (`SYS$COMMON:[SYSEXE]LMF$LURT.DAT`), which comes installed with OpenVMS. On OpenVMS Integrity servers, the units are based on the number of processor cores and the class of the machine specified in the PAK you receive with the license.

### Format

LICENSE CREATE

### Parameters

None.

### Qualifier

#### **/DATABASE=filespec**

Specifies the location of the License Database. The default file specification is defined by the logical name `LMF$LICENSE`, which points to `SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB` on an unmodified OpenVMS system.

### Example

```
$ LICENSE CREATE /DATABASE=SYS$MANAGER:LMF$LICENSE.LDB
```

This command creates the License Database named `LMF$LICENSE.LDB` in the directory `SYS$MANAGER`.

## LICENSE LICENSE DELETE

---

### LICENSE DELETE

Deletes one or more licenses and all history information for those licenses from the License Database.

#### Format

```
LICENSE DELETE product-name[,...]
```

#### Parameter

##### **product-name[,...]**

Name or names of products with a license to be removed from the License Database. You can delete only licenses that have been registered.

#### Qualifiers

##### **/ALL**

Positional qualifier.

Specifies that all licenses with the given product name should be deleted. This qualifier affects only the product name that immediately precedes it in the command string.

##### **/AUTHORIZATION=string**

Positional qualifier.

Specifies a string that helps identify the license you want to delete. You must enter the authorization string exactly as it appears on your PAK. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

##### **/DATABASE=filespec**

Specifies the location of the License Database from which the license or licenses should be deleted. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default License Database name and location.

##### **/ISSUER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that issued the PAK for the product. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

##### **/LOG**

##### **/NOLOG (default)**

Controls whether LICENSE DELETE displays the name of each license that it deletes.



**/PRODUCER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that owns the product for which you have a license. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

**/STATUS=[(keyword)[,...]]**

Positional qualifier.

Selects licenses to be deleted according to the *product-name* parameter specified and one or more license status keywords from the following list:

- ALL (default)—Deletes all specified licenses in the database.
- ACTIVE—Deletes all specified enabled licenses in the database. *ACTIVE* status means that the registered license is enabled for loading. For backward compatibility, the LICENSE LIST command identifies enabled licenses as having a status of active.
- DISABLED—Deletes all specified disabled licenses in the database.
- EXTINCT—Purges specified license information by deleting all extinct license records in the database. Extinct records are history records retained after a license is modified.
- CANCELED—Deletes all specified canceled licenses in the database. Note that current versions of LMF do not set license status to canceled. Old licenses may have this status.

If you enter more than one keyword, separate them with commas, and enclose the list in parentheses. You can abbreviate each option to the minimum number of characters needed to uniquely identify it.

## Description

Use LICENSE DELETE to delete licenses from the License Database. To tailor your command, use options to the /STATUS qualifier and wildcard characters in product name strings.

File space is not released following LICENSE DELETE commands. For information on retrieving Record Management Services (RMS) file space, see the *OpenVMS Record Management Utilities Reference Manual*.

## Examples

1. \$ LICENSE DELETE FORTRAN

This command deletes the Fortran license from the default License Database.

2. \$ LICENSE DELETE FORTRAN, COBOL, PASCAL

This command deletes the Fortran, COBOL and Pascal licenses from the default License Database.

3. \$ LICENSE DELETE FORTRAN /DATABASE=MY\$DISK:MYDATA.LDB

This command deletes the Fortran license from the MY\$DISK:MYDATA.LDB License Database.

## LICENSE LICENSE DELETE

4. \$ LICENSE DELETE FORTRAN /ISSUER=XYLASOFT

This command deletes all licenses for the product named Fortran issued by XYLASOFT from the default License Database. If there are licenses for products named Fortran issued by companies other than XYLASOFT, they are not deleted.

5. \$ LICENSE DELETE \* /STATUS=(EXTINCT)

This command deletes all license records with a status of EXTINCT from the database. This is effectively a purge of all historical information.

---

## LICENSE DISABLE

Disables a license currently registered in the License Database.

### Format

LICENSE DISABLE product-name[,...]

### Parameter

#### **product-name[,...]**

Name or names of products with a license that you want to disable. You can disable only licenses that currently exist in the License Database. Enter the product name exactly as it appears on your Product Authorization Key (PAK).

### Qualifiers

#### **/ALL**

Positional qualifier.

Specifies that all licenses with the given product name should be disabled. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/AUTHORIZATION=string**

Positional qualifier.

Specifies a string that helps identify the license you want to disable. You must enter the authorization string exactly as it appears on your PAK. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/DATABASE=filespec**

Specifies the location of the License Database. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default License Database name and location.

#### **/ISSUER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that issued the PAK for the product. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/LOG**

#### **/NOLOG (default)**

Controls whether LICENSE DISABLE displays the name of each license that it disables.

## LICENSE LICENSE DISABLE

**/PRODUCER=string**  
Positional qualifier.

Specifies the name of the company (for example, DEC) that owns the product for which you have a license. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

### Description

LICENSE DISABLE does not immediately affect loaded licenses. To affect a loaded license, you must first enter a LICENSE UNLOAD command, which unloads the license, but allows current processes to finish using the product. Note that to immediately disable all loaded licenses, you must shut down the system.

You cannot use LICENSE LOAD to activate a disabled license; you must first use LICENSE ENABLE.

LMF does not display error messages when either you or the system attempts to unload a disabled license.

### Example

```
$ LICENSE DISABLE ABCD /PRODUCER=DEC
```

This command disables the license for ABCD software produced by HP. Because no database is specified, LMF uses the default database.

---

## LICENSE ENABLE

Enables an existing license in the License Database so that you can load it with LICENSE LOAD. This command cancels the effect of LICENSE DISABLE, LICENSE COPY, and LICENSE ISSUE, which leave the license disabled. Newly registered licenses are enabled by default.

### Format

LICENSE ENABLE product-name[,...]

### Parameter

**product-name[,...]**

Name or names of products with a license to enable. You can enable only licenses that currently exist in the License Database. Enter the product name exactly as it appears on your PAK.

### Qualifiers

**/ALL**

Positional qualifier.

Specifies that all licenses with the given product name should be enabled. This qualifier affects only the product name that immediately precedes it in the command string.

**/AUTHORIZATION=string**

Positional qualifier.

Specifies a string that helps identify the license you want to enable. You must enter the authorization string exactly as it appears on your PAK. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

**/DATABASE=filespec**

Specifies the location of the License Database. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default License Database name and location.

**/ISSUER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that issued the PAK for the product. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

**/LOG**

**/NOLOG (default)**

Controls whether LICENSE ENABLE displays the name of each license that it enables.

## LICENSE LICENSE ENABLE

**/PRODUCER=string**  
Positional qualifier.

Specifies the name of the company (for example, HP) that owns the product for which you have a license. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

### Description

Use LICENSE ENABLE to reestablish disabled licenses as available for loading with a LICENSE LOAD command.

Enabled licenses can combine with other licenses when loaded for use. Do not enable a license that has expired, and be sure that all include, exclude, and reservation lists are up to date.

Use LICENSE LIST to inspect each license before you enable it. Use LICENSE MODIFY to change include, exclude, and reservation lists.

Because errors do not occur until enabled licenses are loaded, consider entering LICENSE LOAD immediately to load each newly enabled license on each appropriate node in an OpenVMS Cluster. If another combinable license for the same product is already loaded, first unload it with LICENSE UNLOAD. Use the DCL command SHOW LICENSE to see which licenses are currently active on your system. After you unload the other license, enter LICENSE LOAD to load the combination of the newly enabled license and the previously active license.

### Example

```
$ LICENSE ENABLE DECSET /PRODUCER=DEC
```

This command enables the license for DECset software. Because no database is specified, LMF uses the default database. Next, load the license with LICENSE LOAD.

---

## LICENSE ISSUE

Produces a replica of a Product Authorization Key (PAK) that is sent to a file or displayed on your terminal (the default). If the terms and conditions of your license contract allow it, you can then enter this PAK replica in the License Database of another processor. When you enter LICENSE ISSUE, LMF disables the license in the current License Database and marks the license DISABLED. To enable a license that has been marked ISSUED, enter LICENSE ENABLE.

For License Databases connected to a network, consider using LICENSE MOVE.

### Format

LICENSE ISSUE product-name[,...]

### Parameter

#### **product-name[,...]**

Name or names of products with a license to be issued. You can issue only licenses that currently exist in the License Database. Enter the product name exactly as it appears on your PAK.

### Qualifiers

#### **/ALL**

Positional qualifier.

Specifies that all licenses with the given product name should be issued. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/AUTHORIZATION=string**

Positional qualifier.

Specifies a string that helps identify the license you want to issue. You must enter the authorization string exactly as it appears on your PAK. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/DATABASE=filespec**

Specifies the location of the License Database. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default License Database name and location.

#### **/ISSUER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that issued the PAK for the product. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

## LICENSE LICENSE ISSUE

### **/LOG**

#### **/NOLOG (default)**

Controls whether LICENSE ISSUE displays the name of each license that it issues.

### **/OUTPUT[=filespec]**

Specifies the name of the file to which your PAK replica is written. If you do not specify the /OUTPUT qualifier, or if you do not supply a file specification with this qualifier, the output is sent to SYS\$OUTPUT.

If you specify a file name that already exists, this command creates a new version of the file. If you specify a complete file name and version that already exists, this command appends the PAK replica to the existing file.

### **/PROCEDURE**

#### **/NOPROCEDURE (default)**

Specifies that the PAK replica is to be written in the form of a DCL command procedure. Use /PROCEDURE with the /OUTPUT qualifier to create a command procedure in a file. Then you can invoke the procedure to register the PAK replica in the License Database of another processor.

If you do not specify the /OUTPUT qualifier with /PROCEDURE, or if you do not supply a file specification with the /OUTPUT qualifier, the procedure is sent to SYS\$OUTPUT.

### **/PRODUCER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that owns the product for which you have a license. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

## Description

If your license contract allows it, use LICENSE ISSUE to move a license from a License Database on one processor (or OpenVMS Cluster environment) to a License Database on another processor. To move a license, enter LICENSE ISSUE, including enough PAK information to clearly identify the license. LICENSE ISSUE automatically disables the current license but does not immediately unload it; LMF does not terminate any active processes. To unload the license, enter a LICENSE UNLOAD command.

After you issue the PAK replica, read the information, and register it on the new processor as you would any PAK, or, if you used the /PROCEDURE qualifier with the /OUTPUT qualifier, invoke the new DCL command procedure to register the license.

Note that the PAK replica includes only PAK information registered with a LICENSE REGISTER command. The replica does not include any changes made with other LICENSE commands.



## Examples

1. `$ LICENSE ISSUE /OUTPUT=SYS$MANAGER:FORTTRAN.PAK -  
_ $ /PRODUCER=DEC FORTRAN`

This command issues a PAK replica, which you can use to register the Fortran license on a new processor (or OpenVMS Cluster environment), and puts it into the file named FORTTRAN.PAK. The next step is to print the file, read the information, and, using a LICENSE REGISTER command or VMSLICENSE.COM, enter the correct information in the License Database of the new processor. The Fortran license in the current License Database is marked ISSUED and is disabled.

2. `$ LICENSE ISSUE /PRODUCER=DEC VOLSHAD`

This command displays, at the current terminal, a PAK replica with the information from the VOLSHAD (Volume Shadowing) license. This display is reproduced below. The license registered in the current License Database is marked ISSUED and is disabled. You can register the data from this replica of a PAK in the License Database of another processor using either VMSLICENSE.COM or LICENSE REGISTER.

```
Software Product Authorization Key Replica
Issued by CASPER
Issued on 24-FEB-2001 14:23
-----
Issuer:          DEC
Authorization:   ALS-WM-93166-5573
Product Name:    VOLSHAD
Producer:        DEC
Units:           460
Version:         5.4
PAK Termination Date: 31-DEC-2001
Availability:    E
Options:         MOD UNITS
Checksum:        1-ADEB-DOCJ-NENC-KDBM
```

3. `$ LICENSE ISSUE /PROCEDURE /OUTPUT=FORTTRAN-USA10.COM -  
_ $FORTTRAN /AUTHORIZATION=USA-10`

This command generates a DCL command procedure such as the following to be used for registering the specified license in a License Database:

```
$! Software Product Authorization Key Replica
$! Issued by CASPER
$! Issued on 23-Oct-2001 14:23
$ LICENSE REGISTER FORTTRAN -
  /ISSUER=DEC -
  /PRODUCER=DEC -
  /AUTHORIZATION=USA-10 -
  /UNITS=400 -
  /VERSION=5.4 -
  /AVAILABILITY=F -
  /CHECKSUM=1-HIDN-INDA-COMP-DAHH
```

## LICENSE LICENSE LIST

---

### LICENSE LIST

Displays information from the License Database on disk about the specified license or licenses. Use one or more qualifiers to control the form, content, and location of information displayed.

The SHOW LICENSE command, described in the *HP OpenVMS DCL Dictionary* and in this appendix, displays information from the License Database in memory.

#### Format

```
LICENSE LIST [product-name[,...]]
```

#### Parameter

##### **product-name[,...]**

Name or names of products with a license that you want to list. You can list only licenses that currently exist on disk in the License Database. You can specify one product name or use wildcard characters to display licenses. The *product-name* parameter is optional; the default is to display all of the licenses.

#### Qualifiers

##### **/AUTHORIZATION=string**

Positional qualifier.

Specifies a string that helps identify the license you want to list. You must enter the authorization string exactly as it appears on your PAK. Use this optional qualifier only if you need it to identify the license.

This qualifier affects only the product name that immediately precedes it in the command string.

##### **/BEFORE**

Used with /TERMINATION\_DATE and /RELEASE\_DATE, selects only those licenses whose times are before the time specified with the other qualifiers.

The /BEFORE qualifier cannot be used with the /SINCE qualifier.

##### **/BRIEF (default)**

Specifies a listing from the License Database that includes only the license product and producer names.

##### **/DATABASE=filespec**

Specifies the location of the License Database. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default License Database name and location.

##### **/FULL**

Specifies a listing from the License Database that includes a full display of the specified license or licenses.

##### **/HISTORY**

Specifies a listing from the License Database that includes the history records in the License Database for the specified license or licenses.

**/ISSUER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that issued the PAK for the product. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

**/OUTPUT[=filespec]**

Specifies the name of the file to which your list is written. If you do not specify the /OUTPUT qualifier, or if you do not supply a file specification with this qualifier, the output is sent to SYS\$OUTPUT.

**/PRODUCER=string**

Positional qualifier.

Specifies the name of the company (for example, HP) that owns the product for which you have a license. Use this optional qualifier only if you need it to identify the license.

This qualifier affects only the product name that immediately precedes it in the command string.

**/RELEASE\_DATE=date**

Used with /BEFORE or /SINCE, specifies a listing from the License Database that includes only licenses with a release date on or after the date specified. The date must be presented in the standard OpenVMS format: *dd-mmm-yyyy*. The default value is /SINCE /RELEASE\_DATE=TODAY.

**/SELECTION\_WEIGHT**

Produces a full display that includes the current selection weights assigned to individual PAKs.

**/SINCE**

Used with /TERMINATION\_DATE and /RELEASE\_DATE, selects only those licenses whose times are after the time specified with the other qualifiers.

/SINCE cannot be used with /BEFORE.

**/TERMINATION\_DATE[=date]**

Used with /BEFORE or /SINCE, specifies a listing from the License Database that includes only licenses with a termination date on or after the date specified. The date must be presented in the standard OpenVMS format: *dd-mmm-yyyy*. The default value is /SINCE /TERMINATION\_DATE=TODAY.

**/VERSION=nn.nn**

Positional qualifier.

Specifies the version number of the product for which you have a license. Versions use the format *integer.integer*. You can specify wildcard syntax as *\*.\** but not *\** alone. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

# LICENSE LICENSE LIST

## Description

LICENSE LIST displays license records as they appear on disk in the License Database. LICENSE LIST /BRIEF does not produce a display with history records. You can control the displays as follows:

- After you enter LICENSE LIST with the /BRIEF qualifier, you can scroll through the display with the arrow keys on your keyboard.
- After you enter LICENSE LIST with the /FULL or /HISTORY qualifier, which displays the first LICENSE record, you can see the other records one at a time by pressing Return. You can also scroll through the license records using the Previous Screen key (or PF3) and the Next Screen key (or PF4).

For any LICENSE LIST display, use the arrow keys to scroll vertically or horizontally one line at a time. Press Ctrl/Z to exit from the display.

Note that a LICENSE LIST command may display the status of a registered license as *Active*. This means the registered license is enabled for loading; it has not been disabled. It does not necessarily mean the license was loaded with a LICENSE LOAD command. The LICENSE LIST command displays only information on disk in the License Database; enter SHOW LICENSE to determine all active licenses on the current system.

You can also list licenses using the VMSLICENSE.COM command procedure.

## Examples

1. \$ LICENSE LIST /FULL

This example displays a list of the names of product licenses in the License Database on an OpenVMS Alpha system. Note that the LMF Version shown refers to the software that created the database.

Use CTRL/Z to exit, PF3-PF4 for Previous-Next Screen, Arrow Keys to scroll.

```
License Management Facility  V1.2
License Database File:      WORK2:[BACKUP]LMF$LICENSE.LDB;1
Created on:                 20-JUL-2000
Created by user:           USER_1
Created by LMF Version:    V1.1
-----
Issuer:                     DEC
Authorization:
Product Name:              OPENVMS-ALPHA
Producer:                  DEC
Units:                     500
Version:                   0.0
Release Date:              4-MAY-2001
PAK Termination Date:     (none)
Availability:              0
Activity:                   000000100
Options:                   MOD_UNITS
Product Token:
Hardware ID:

Revision Level:            1
Status:                   Active
Command:                   REGISTER
```

# LICENSE LICENSE LIST

## 2. \$ LICENSE LIST /FULL

This example displays a list of the names of product licenses in the License Database on an OpenVMS Integrity server system.

Use CTRL/Z to exit, PF3-PF4 for Previous-Next Screen, Arrow Keys to scroll.

License Management Facility V2.0

License Database File: SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB;1  
Created on: 11-MAR-2010  
Created by user: SYSTEM  
Created by LMF Version: V2.0

-----  
Issuer: HP  
Authorization: USA-1715  
Product Name: C  
Producer: HP  
Units: 3  
Version: 0.0  
Release Date: (none)  
PAK Termination Date: 31-DEC-2012  
Availability: 0  
Activity: 000000001  
Options: IA64\_ALPHA  
Product Token:  
Hardware ID:  
  
Revision Level: 1  
Status: Active  
Command: REGISTER  
Modified by user: SYSTEM  
Modified on: 11-MAR-2010 12:18:59.95

-----  
Issuer: HP  
Authorization: I64-AB-001  
Product Name: DVNETEXT  
Producer: HP  
Units: 20  
Modified Units: 4  
Version: 0.0  
Release Date: (none)  
PAK Termination Date: 31-DEC-2012  
Options: IA64, PCL  
Product Token:  
Hardware ID:  
  
Revision Level: 2  
Status: Active  
Command: MODIFY  
Modified by user: SYSTEM  
Modified on: 11-MAR-2010 12:29:42.18

## LICENSE LICENSE LIST

```
-----  
Issuer:                HP  
Authorization:         I64-AB-004  
Product Name:          OPENVMS-I64-BOE  
Producer:              HP  
Units:                 2  
Version:               0.0  
Release Date:          (none)  
PAK Termination Date: 31-DEC-2012  
Options:               IA64, PCL  
Product Token:  
Hardware ID:  
  
Revision Level:        1  
Status:                Active  
Command:               REGISTER  
Modified by user:      SYSTEM  
Modified on:           11-MAR-2010 12:33:26.19
```

### 3. \$ LICENSE LIST /HISTORY FORTRAN

This example displays a listing with full information of a current Fortran license issued by HP.

The first screen, shown here, displays the most recent license record for the Fortran license. To see the history records one screen at a time, press Return. The revision level of the displayed record is 2, and the status is Active. The next screen would display the previous license record with a revision level of 1 and a status of Extinct.

Use CTRL/Z to exit, PF3-PF4 for Previous-Next Screen, Arrow Keys to scroll.

```
License Management Facility V1.2  
License Database File:  ART::SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB  
  
Created on:             17-AUG-2000  
Created by user:        USER_2  
Created by LMF Version: V1.2
```

```
-----  
Issuer:                DEC  
Authorization:         USA-2468  
Product Name:          FORTRAN  
Producer:              DEC  
Units:                 0  
Modified Units:        2000  
Version:               5.4  
Release Date:          (none)  
PAK Termination Date: 31-DEC-2000  
Modified Termination Date: 30-NOV-2001  
Availability:          F (Layered Products)  
Activity:              0  
Options:               MOD_UNITS  
Hardware ID:  
  
Revision Level:        2  
Status:                Active  
Command:               MODIFY  
Modified by user:      DEGAS  
Modified on:           19-AUG-2000 14:32:23.41  
Include:               ART
```

# LICENSE LICENSE LIST

## 4. \$ LICENSE LIST /FULL

This example displays the full listing of a license that has been modified using the /VIRTUAL qualifier, for an OpenVMS guest system.

License Management Facility V2.0

License Database File: SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB;1  
Created on: 1-MAR-2010  
Created by user: SYSTEM  
Created by LMF Version: V2.0

-----  
Issuer: HP  
Authorization: I64-AB-001  
Product Name: OPENVMS-I64-HAOE  
Producer: HP  
Units: 4  
Version: 0.0  
Release Date: (none)  
PAK Termination Date: 10-MAR-2011  
Options: IA64, PCL, VIRTUAL  
Product Token:  
Hardware ID:  
  
Revision Level: 2  
Status: Active  
Command: MODIFY  
Modified by user: SYSTEM  
Modified on: 8-MAR-2010 16:45:00.32

## LICENSE LICENSE LOAD

---

### LICENSE LOAD

Loads licenses, making them available for product authorization on the current node. The product licenses must be registered and current in the License Database. That is, they must not have been disabled or issued.

If the license is already loaded, LMF returns an informational message, unloads the license, and then loads the license.

To use this command, you need CMKRNL, SYSNAM, and SYSPRV privileges.

#### Format

```
LICENSE LOAD [product-name][,...]
```

#### Parameter

##### **[product-name][,...]**

Name or names of products with a license to be loaded. You can load only licenses that are currently registered and enabled in the License Database. Enter the product name exactly as it appears on your Product Authorization Key (PAK). If you do not specify a product name, LICENSE LOAD loads all of the products that are registered and enabled.

You cannot use wildcard characters for *product-name*.

#### Qualifiers

##### **/AUTHORIZATION=string**

Positional qualifier.

Specifies a string that helps identify the license you want to register. You must enter the authorization string exactly as it appears on your PAK. This qualifier affects only the product name that immediately precedes it in the command string.

##### **/DATABASE=filespec**

Location of the License Database. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default License Database name and location.

##### **/ISSUER=string**

Positional qualifier.

Name of the company (for example, DEC) that issued the PAK for the product. Use this optional qualifier only if you need it to identify the license.

This qualifier affects only the product name that immediately precedes it in the command string.

##### **/LOG (default)**

##### **/NOLOG**

Controls whether or not LICENSE LOAD displays a message to acknowledge the loading of each license.



**/OEDB - Integrity servers only**

Using this qualifier refreshes the contents of the OE database. The contents of the OE database are described in a datafile (LMF\$OE.DAT). If new variants of operating environments become available, HP will provide a new datafile with information on the new or changed operating environments. Using LICENSE LOAD/OEDB updates your OE database without having to reboot the system.

**/PRODUCER=string**

Positional qualifier.

Name of the company that owns the product for which you have a license. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

**/UNLOAD (default)**

**/NOUNLOAD**

When requested to load a license that is currently loaded, LMF first automatically unloads it and then loads the latest license.

You can specify /NOUNLOAD to verify whether or not there is already a license loaded; LMF issues the warning LICENSE-W-ALREADYLOADED and does not load the license. To then load the license, follow these steps:

1. Manually unload the current license with the LICENSE UNLOAD command.
2. Reissue the LICENSE LOAD command.

## Description

The LICENSE LOAD command loads licenses registered in the License Database. To use a licensed product, ensure that the system loads the registered license. When you register a license with VMSLICENSE.COM, you can confirm an option to load the license, whereas if you register a license with LICENSE REGISTER, you must also load it with LICENSE LOAD.

Use LICENSE LOAD only after you register a new license; LMF automatically loads all registered licenses at each subsequent system startup. You can enter LICENSE LOAD at other times to load modifications made with other LICENSE commands.

You can enter one LICENSE LOAD command without *product-name* to load all the available registered licenses.

---

**Note**

---

Registered licenses are enabled for loading by default. You can, however, disable a registered license to prevent loading.

---

A LICENSE START command entered interactively or when the system reboots also loads all licenses that are registered and enabled.

If you register multiple licenses for a single product, LICENSE LOAD loads all of the matching licenses. You do not typically load individual licenses, and you cannot unload individual licenses for a product. The Availability, Activity, Personal Use, and User license units of the multiple licenses work in concert to provide more product availability.

## LICENSE LICENSE LOAD

In an OpenVMS Cluster environment, each system loads licenses when it reboots. If you need to load a license for all assigned nodes of a running cluster, you can do one of the following:

- Log in to each OpenVMS Cluster node, and enter LICENSE LOAD.
- Invoke the OpenVMS SYSMAN utility to execute the LICENSE LOAD command on the desired OpenVMS Cluster nodes. See the *HP OpenVMS System Manager's Manual* for details on defining your management environment and executing commands on a list of nodes.

A LICENSE LOAD command can fail, sending a message to the operator communication manager (OPCOM) for any of the following reasons:

- Insufficient license units are registered for the current node.
- The current date is later than the license termination date.
- A license checksum does not match the rest of the license data. Check for data corruption in the License Database.

If you attempt to load a disabled license or a license modified to exclude the current node in an OpenVMS Cluster environment, OPCOM does not display an error message.

If licenses for more than one product are being loaded, LICENSE LOAD continues with the next license following a failure.

### Examples

1. 

```
$ LICENSE MODIFY /INCLUDE=MUSIC FORTRAN
$ LICENSE LOAD FORTRAN
```

The commands in this example illustrate a situation in which you enter a LICENSE LOAD command interactively. LICENSE LOAD loads the product Fortran on the node MUSIC. Data in the License Database determines whether the license is successfully loaded on the specified node.

2. 

```
$ LICENSE LOAD BASIC
%LICENSE-W-NOLOAD, license was not loaded for BASIC
-LICENSE-F-EXCEEDED, attempted usage exceeds active license limits
```

This command attempts to load the product BASIC, but LICENSE LOAD fails because too few license units are registered to authorize use on the current processor.

---

## LICENSE MODIFY

Modifies a license for system management and license-sharing purposes. Immediately changes data in the License Database, but your modifications do not affect the running system until you load the modified license.

### Format

LICENSE MODIFY *qualifier*[,...] *product-name*[,...]

### Parameter

#### **product-name**[,...]

Name or names of products with a license to be modified. You can modify only licenses that currently exist in the License Database.

### Qualifiers

#### **/ADD**

Used with the **/INCLUDE** or **/EXCLUDE** qualifier, specifies that the node names provided are to be added to the previously established include or exclude lists.

Used with the **/RESERVE** qualifier, specifies that the user names provided are to be added to the previously established reservation lists.

When you use **/ADD**, you do not need to retype the entire list to add a new node name or user name.

#### **/ALL**

Positional qualifier.

Modifies all the licenses with the given product name. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/AUTHORIZATION=string**

Positional qualifier.

Specifies a string that helps identify the license you want to modify. You must enter the authorization string exactly as it appears on your PAK. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/COMBINE**

#### **/NOCOMBINE**

Modifies a PAK by adding or removing the **COMBINE** option. If the PAKs are combinable, LMF combines them during license loading.

#### **/COMMENT=string**

Specifies a string of text. Use this comment field of up to 63 characters to associate information for this transaction with the license. History records for the license retain this license information. If you specify more than one word, enclose the text in quotation marks (" "). This qualifier is optional.

The text in the comment field is replaced only when you enter new comments with another **LICENSE MODIFY** command. At this point the old comment text is available as a history record.

## LICENSE LICENSE MODIFY

### **/DATABASE=filespec**

Specifies the location of the License Database. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default License Database name and location.

### **/EXCLUDE=(node-name[,node-name,...])**

Specifies that the named node or nodes in an OpenVMS Cluster environment cannot access the licensed product. The excluded nodes cannot load (with a LICENSE LOAD or LICENSE START command) the license registered in the License Database. Each **node-name** argument must be a System Communications Services (SCS) node name or a system parameter set with the System Generation utility (SYSGEN). The node name might not be the same as the DECnet node name. If you specify more than one node name, separate them with commas, and enclose the list in parentheses. This qualifier is optional.

To modify previously defined lists without having to retype all of the node names, use the /ADD or /REMOVE qualifiers with /EXCLUDE.

You can control license access to nodes with /EXCLUDE and control user access with /RESERVE, but you cannot use these qualifiers on the same command line. To use both types of control with the same license, you must enter separate LICENSE MODIFY commands.

### **/INCLUDE=(node-name[,node-name,...])**

Specifies that the named node or nodes in an OpenVMS Cluster environment can access the licensed product. Only the included nodes can load (with a LICENSE LOAD or LICENSE START command) the license registered in the License Database. Each **node-name** argument must be an SCS node name, or a system parameter set with SYSGEN. The node name might not be the same as the DECnet node name.

Licenses for the OpenVMS operating system usually specify the NO\_SHARE option on their PAKs. In a cluster environment you must restrict each of these OpenVMS licenses to a single node. If you did not do this when registering with VMSLICENSE.COM, enter LICENSE MODIFY/INCLUDE=*node-name*, specifying one SCS node name for each OpenVMS license.

To specify more than one SCS node name for a license that does not specify NO\_SHARE, separate the names with commas, and enclose the list in parentheses. This qualifier is optional.

To modify previously defined lists without having to retype all of the node names, use the /ADD or /REMOVE qualifiers with /INCLUDE.

You can control license access to nodes with /INCLUDE and control user access with /RESERVE, but you cannot use these qualifiers on the same command line. To use both types of control with the same license, you must enter separate LICENSE MODIFY commands.

### **/ISSUER=string**

Positional qualifier.

Specifies the name of the company (for example, DEC) that issued the PAK for the product. Use this qualifier only if it is required to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

**/LOG**

**/NOLOG (default)**

Controls whether LICENSE MODIFY displays the name of each license that it modifies.

**/NO\_SHARE**

**/NONO\_SHARE**

Specifies whether to add or subsequently remove /NO\_SHARE from a PAK.

Adding /NO\_SHARE prevents the sharing of the PAK units with other cluster nodes.

PAKs with /NO\_SHARE require you to provide the SCS node name of the cluster node that will be using this particular license. See the /INCLUDE qualifier for more information.

Note that if /NO\_SHARE is present on your PAK when you register it, you cannot remove the option using /NONO\_SHARE. Only if you add /NO\_SHARE with the MODIFY command, can you subsequently remove it.

**/PRODUCER=string**

Positional qualifier.

Specifies the name of the company (for example, DEC) that owns the product for which you have a license. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

**/REMOVE**

Used with the /INCLUDE or /EXCLUDE qualifier, specifies that the node names provided are to be removed from the previously established include or exclude lists.

Used with the /RESERVE qualifier, specifies that the user names provided are to be removed from the previously established reservation lists.

When you use /REMOVE, you do not need to retype the entire list to remove a node name or user name.

**/RESERVE=(user-name[,user-name,...])**

Specifies that the license or licenses are to be reserved for use by the users listed in the *user-name* parameter. Users not listed are denied access to the product. The value applied to *user-name* differs from product to product. See your Software Product Description (SPD) for details.

Most products define **user-name** to be the user name OpenVMS maintains for each account. This is the name you type at the Username prompt during login.

If your PAK specifies the RESERVE\_UNITS option, you *must* assign one or more users to a reservation list. On OpenVMS Alpha and VAX systems, the number of user names allowed per list depends on the number of activity units available and a constant value or the License Unit Requirement Tables (LURTs). Calculate this number as you would for any Activity License. For example, a 200-unit license with a constant value of 100 is a two-user license. On OpenVMS Integrity server systems, units are expressed in single units that directly correlate to the constant value listed.

## LICENSE LICENSE MODIFY

You can also create and modify a reservation list for Availability and regular Activity Licenses that do not specify the RESERVE\_UNITS option. Because these licenses do not limit the number of names on the list, you can assign as many names as you like to the reservation list. All users not on the list are denied access.

Although you can control license access to nodes with /INCLUDE and /EXCLUDE qualifiers and control user access with the /RESERVE qualifier, you cannot use these qualifiers on the same command line. If you want to use both types of control with the same license, you must enter separate LICENSE MODIFY commands.

Use the /ADD and /REMOVE qualifiers for further control in modifying previously established reservation lists.

### **/SELECTION\_WEIGHT=number**

Modifies the selection weight. Selection-weight values determine the order in which LMF checks multiple licenses when a product makes a license grant request. LMF checks higher-weighted licenses before lower-weighted ones. Specify arbitrary numbers between 1 and 1000.

---

#### **Note**

---

You cannot modify selection weights for Availability Licenses.

---

To restore the selection weight of a PAK to the default value, enter the LICENSE MODIFY command with /SELECTION\_WEIGHT=0. For example, you can use either of the following commands:

```
$ LICENSE MODIFY FORTRAN /SELECTION_WEIGHT=0  
$ LICENSE MODIFY FORTRAN /NOSELECTION_WEIGHT
```

### **/TERMINATION\_DATE=date**

Date at which the product license is to be terminated. If your PAK supplied a license termination date, LMF uses the earliest date to determine the termination date. The date must be presented in the standard OpenVMS format: *dd-mmm-yyyy*. If you want to restrict a product from further use today, enter yesterday's date; LMF terminates the license at the end of the day specified.

### **/UNITS=n**

Number of license units you want on a license that includes the MOD\_UNITS option. If your PAK allows you to modify the license units, use this qualifier to change the value in the License Database.

### **/VIRTUAL**

### **/NOVIRTUAL**

Specifies that the modified license must only be loaded on OpenVMS guests. Virtual licenses are ignored during physical machine LICENSE LOAD command processing. To see the licenses that are ignored, use the /LOG qualifier with the LICENSE LOAD command. Use the /NOVIRTUAL qualifier to remove the virtual option from the license.

The /VIRTUAL qualifier is valid for licenses with the IA64 and PCL options.

Licenses that you intend to load on OpenVMS guest cluster members must be modified with the /VIRTUAL qualifier. It is optional, but recommended to modify licenses for standalone OpenVMS guest systems with the /VIRTUAL qualifier.

HP recommends that you use `/INCLUDE` or `/EXCLUDE` lists on your virtual machine host licenses to define all the OpenVMS guest cluster members from a host that must load the license. In future, HP may limit the usage of a Virtual license to one host, which is the first host that loads the license.

## Description

Use the `LICENSE MODIFY` command to modify a license. To control which nodes in a cluster environment have access to what software, use `LICENSE MODIFY` with the `/INCLUDE` or `/EXCLUDE` qualifier. For example, you can load licenses for products used less often or requiring limited access on one node.

If you do not specify which nodes can load a license (with a `LICENSE LOAD` or `LICENSE START` command), LMF loads a license on a first-come, first-served basis. When your license has insufficient license units for full cluster environment use, control product access with an include list.

Because most OpenVMS PAKs use the `/NO_SHARE` option, in a cluster environment you must restrict these operating system licenses to one node. Enter `LICENSE MODIFY/INCLUDE=node-name`, specifying only one SCS node name for each OpenVMS license.

To control which users have access to a product, use `LICENSE MODIFY` with the `/RESERVE` qualifier. You can create and modify a reservation list for any kind of license. Only users on the reservation lists are allowed access to the product.

If your PAK specifies the `RESERVE_UNITS` option, you *must* assign one or more users to a reservation list. The number of user names allowed per list depends on the number of activity units available and a constant value or the License Unit Requirement Tables (LURTs). Calculate this number as you would for any Activity License. For example, a 200-unit license with a constant value of 100 is a two-user license.

Use the `/ADD` and `/REMOVE` qualifiers in conjunction with the `/INCLUDE`, `/EXCLUDE`, and `/RESERVE` qualifiers when you modify existing include, exclude, and reservation lists.

To add comments about a license in the License Database, use `LICENSE MODIFY` with the `/COMMENT` qualifier.

If your PAK includes the `MOD_UNITS` option, you can use the `/UNITS` qualifier to specify the number of license units you want for your registered license.

Use the other `LICENSE MODIFY` command qualifiers only as needed to identify the correct license.

You can also modify a license record using the `VMSLICENSE.COM` command procedure.

### List Size Restrictions

Two restrictions apply to the size of lists (reservation lists, include lists, or exclude lists). These restrictions apply to PAKs of all license types.

- On any single PAK, the sum of characters contained in all lists must not exceed 5000 characters.

Because the length of names varies and some overhead is used for each name, this 5000-character limit cannot be expressed as an exact number of permissible names. However, HP guarantees that at least 400 names, in

## LICENSE LICENSE MODIFY

total, can be specified in the various types of lists. For example, each of the following represents the minimally guaranteed number of names:

- Reservation list with up to 400 user names
- Reservation list with up to 200 user names plus an include list with up to 200 node names (totaling up to 400)
- Reservation list with up to 200 user names plus an exclude list with up to 200 node names (totaling up to 400)
- Include list with up to 400 node names
- Exclude list with up to 400 node names

---

### Note

---

If you enter more names than are permitted, LICENSE LIST might not be able to display all names entered. In this case, you receive the error message LICENSE-F-CORRUP. However, the License Database is not actually corrupt, and the PAKs can still be loaded into memory (though the names are not displayed).

---

- The LICENSE LOAD and LICENSE START commands can load into memory a reservation list with no more than 30,000 characters. (Include and exclude lists, which are not loaded into memory, are irrelevant to the 30,000-character limit.)

Because the length of names varies and some overhead is used for each name, this 30,000-character limit cannot be expressed as an exact number of permissible names. But HP guarantees that, for each product, at least 2000 user names can appear on reservation lists. In the case of an OpenVMS Cluster, this is a per-node limit.

Note that, because 2000 user names is a per-product limit and because there can be more than one PAK per product, the number of user names on a per-product basis is the *sum of* the user names specified on each PAK.

For example, if three activity PAKs for the DECwrite product were registered on a system and each PAK specified a reservation list with 200 user names, the total number of user names for that product is 600. This is safely below the 30,000-character (2000 user name) limit and below the 5000-character (400 user name) limit.

## Examples

1. 

```
$ LICENSE MODIFY /EXCLUDE=(DANCE,THEATR) -  
_ $ /COMMENT="Modified to exclude nodes DANCE & THEATR 10/23/04" -  
_ $ FORTRAN
```

This command modifies the Fortran license in the License Database so that users cannot access Fortran from the nodes named DANCE and THEATR. A comment is added to the database record for future reference.



## LICENSE LICENSE MODIFY

2. \$ LICENSE MODIFY /ADD /INCLUDE=(DRAMA) -  
\_ \$ /COMMENT="Modified to add node named DRAMA 10/23/04" -  
\_ \$ FORTRAN

This command modifies the Fortran license in the License Database so that users can access Fortran from the node DRAMA in addition to any nodes previously named in the license include list.

3. \$ LICENSE MODIFY /UNITS=1200 FORTRAN  
\$ LICENSE LOAD FORTRAN

This command changes the license units on a license with the MOD\_UNITS option.

4. \$ LICENSE MODIFY /TERMINATION\_DATE=1-JAN-2005 FORTRAN

Unless an earlier termination date exists, this command sets a new termination date of 1-JAN-2005 for the Fortran license.

5. \$ LICENSE MODIFY /EXCLUDE="" FORTRAN

This command removes all nodes from the previously established exclude list. All nodes now have access to the Fortran license.

## LICENSE LICENSE MOVE

---

### LICENSE MOVE

Moves one or more licenses from one License Database to another. When you use LICENSE MOVE, LMF deletes those licenses from the source License Database.

For License Databases not connected to a network, consider using the LICENSE ISSUE /PROCEDURE command.

### Format

```
LICENSE MOVE product-name[,...] output-database
```

### Parameters

#### **product-name[,...]**

Name or names of products with a license to be moved to the output License Database.

#### **output-database**

File specification of the License Database to which the license or licenses should be moved. This database must have been previously created using LICENSE CREATE.

If you enter a partial file specification (for example, specifying only a directory), LMF\$LICENSE is the default file name, and .LDB is the default file type. If you do not specify a device or directory, the current default device and directory are used.

### Qualifiers

#### **/ALL**

Positional qualifier.

Specifies that all licenses with the given product name should be moved. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/AUTHORIZATION=string**

Positional qualifier.

Specifies a string that helps identify the license you want to modify. You must enter the authorization string exactly as it appears on your PAK. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

#### **/DATABASE=filespec**

Specifies the location of the License Database from which the license or licenses should be moved. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default License Database name and location.

**/ISSUER=string**

Positional qualifier.

Specifies the name of the company (for example, DEC) that issued the PAK for the product. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

**/LOG**

**/NOLOG (default)**

Controls whether LICENSE MOVE displays the name of each license that it moves.

**/PRODUCER=string**

Positional qualifier.

Specifies the name of the company (for example, DEC) that owns the product for which you have a license. Use this optional qualifier only if you need it to identify the license. This qualifier affects only the product name that immediately precedes it in the command string.

## Description

If your license contract allows it, use LICENSE MOVE to move a license from one License Database to another. To move a license, enter LICENSE MOVE, including enough PAK information to clearly identify the license. LICENSE MOVE automatically deletes the license from the source License Database.

Note that the moved license includes only the general PAK information normally provided by LICENSE REGISTER. LICENSE MOVE does not transfer any user-supplied data such as reservation lists, modified termination dates, modified units, include or exclude node lists, or comments.

## Examples

1. `$ LICENSE MOVE FORTRAN ALT_SYS2:LMF$LICENSE.LDB`

This command moves the Fortran license in the default License Database to the ALT\_SYS2:LMF\$LICENSE.LDB output License Database. This command fails if the default database contains more than one Fortran license.

2. `$ LICENSE MOVE FORTRAN /DATABASE=LMFDATA:LMF$LICENSE.LDB -  
_ $ ALT_SYS:LMF$LICENSE.LDB`

This command moves the Fortran license in the source License Database, LMFDATA:LMF\$LICENSE.LDB, to the destination License Database, ALT\_SYS:LMF\$LICENSE.LDB. This command fails if the source License Database contains more than one Fortran license.

3. `$ LICENSE MOVE FORTRAN /ALL ALT_SYS2:LMF$LICENSE.LDB`

This command moves all Fortran licenses in the default License Database to the output License Database, ALT\_SYS2:LMF\$LICENSE.LDB.

4. `$ LICENSE MOVE * ALT_SYS2:LMF$LICENSE.LDB`

This command merges two databases by moving all licenses in the default License Database to the output License Database, ALT\_SYS2:LMF\$LICENSE.LDB.

## LICENSE LICENSE REGISTER

---

### LICENSE REGISTER

Adds a new license to the License Database. A Product Authorization Key (PAK) provides the product name and information you need to register the license. You must enter all information provided by your PAK exactly as it appears.

You can also register a new product license with the command procedure SYS\$UPDATE:VMSLICENSE.COM, which provides a prompt-based interface to the LICENSE REGISTER command.

#### Format

```
LICENSE REGISTER product-name
```

#### Parameter

##### **product-name**

Name of the product with a license to register. You can register only licenses that do not currently exist in the License Database. You can register multiple licenses for the same product when they have different authorization numbers. Enter the product name exactly as it appears on your PAK.

You cannot use wildcard characters for the *product-name* parameter with this command.

#### Qualifiers

##### **/ACTIVITY=code | CONSTANT=integer**

Specifies a license unit code that corresponds to a License Unit Requirement Table (LURT) or to a constant value. If your PAK supplies an activity code, you must enter the code exactly as it appears. The current codes are A, B, C, D, E, F, G, H, and I. If your PAK specifies the keyword CONSTANT, then you must also specify the integer value. This denotes a constant requirement for all System Marketing Models (SMMs) equal to the value given. If your PAK specifies the decimal value 0, then the license has no requirement for that license type. PAK issuers determine the value for this element.

##### **/AUTHORIZATION=string**

Specifies a string that helps identify the license you want to register. You must enter the authorization string exactly as it appears on your PAK.

##### **/AVAILABILITY= code | CONSTANT=integer**

Specifies a license unit code that corresponds to a License Unit Requirement Table (LURT) or to a constant value. If your PAK supplies an availability code, you must enter the code exactly as it appears. The current codes are A, B, C, D, E, F, G, H, and I. If your PAK specifies the keyword CONSTANT, then you must also specify the integer value from your PAK. PAK issuers determine the value for this element.

##### **/CHECKSUM=string**

Specifies a 17-character verification string created by the PAK issuer for each PAK. The checksum string is presented in the format *n-ccc-ccc-ccc-ccc*, where *n* is an integer and *c* is an alphabetic character from A through P. A PAK presents the checksum string with hyphen (-) characters for readability. Because LMF does not count hyphens for authorization, you do not have to enter them.

Otherwise, you must enter the checksum string exactly as it appears on your PAK.

**/DATABASE=filespec**

Specifies the location of the License Database. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB in an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default database.

**/HARDWARE\_ID=string**

Specifies the identification number of the hardware on which the product is licensed. If your PAK supplies a hardware identification number, you must enter the information exactly as it appears. On Integrity server systems, the HARDWARE\_ID string is in the format SOCKETS=*n*.

**/ISSUER=string**

Specifies the name of the company (for example, DEC) that issued the PAK for the product. Note that the PAK issuer is often the same as the producer. You must enter the information exactly as it appears on your PAK.

**/OPTIONS=[(keyword[,...])]**

Specifies LICENSE REGISTER options. If your PAK supplies any license options, you must enter this information exactly as it appears.

Table A-1 describes the available keywords.

**Table A-1 LICENSE REGISTER /OPTIONS Keywords**

Keyword	Meaning
ALPHA	Identifies Availability Licenses for Alpha systems.
HARD_COMPLIANCE	Identifies a license that will enforce compliance to license terms.
IA64	Identifies Licenses for Integrity server systems.
IA64_ALPHA	Identifies Activity Licenses that are valid for OpenVMS Integrity servers and OpenVMS Alpha systems.
IA64_ALPHA_VAX	Identifies Activity Licenses that are valid for OpenVMS Integrity servers, OpenVMS Alpha, and OpenVMS VAX systems.
MOD_UNITS	You can modify the number of license units.
NO_SHARE	You cannot use the license on more than one processor in an OpenVMS Cluster environment.
	To use this license in a cluster, designate it for one node. Issue LICENSE MODIFY with the /INCLUDE qualifier.
PCL	Designates a Per Core License on an OpenVMS Integrity server system.
RESERVE_UNITS	The license must be assigned to one or more users. Reserve the license using LICENSE MODIFY with the /RESERVE qualifier.

(continued on next page)

## LICENSE LICENSE REGISTER

Table A-1 (Cont.) LICENSE REGISTER /OPTIONS Keywords

Keyword	Meaning
USER	Designates a User License.
VAX_ALPHA	Identifies Availability Licenses that are valid for both OpenVMS VAX and OpenVMS Alpha systems.

If you enter more than one keyword, separate them with commas, and enclose the list in parentheses. You can abbreviate each option to the minimum number of characters needed to uniquely identify it.

### **/PRODUCER=string**

Specifies the name of the company (for example, HP) that owns the product for which you have a license. You must enter the information exactly as it appears on your PAK.

### **/RELEASE\_DATE=date**

Specifies a product release date such that the license authorizes use of all product versions released on or before the date. If your PAK supplies a product release date, you must enter the information exactly as it appears. The date must be presented in the standard OpenVMS format: *dd-mmm-yyyy*.

### **/TERMINATION\_DATE=date**

Specifies the date on which the product license terminates. If your PAK supplies a license termination date, you must enter it exactly as it appears. The date must be presented in the standard OpenVMS format: *dd-mmm-yyyy*.

### **/TOKEN=string**

Specifies a string of information associated with some products. This option can enable or disable certain product features. See your product documentation for details. If your PAK provides token information, you must enter it exactly as it appears.

### **/UNITS=number**

Specifies the number of license units for your license. You must enter the number exactly as it appears on your PAK even if your PAK specifies the MOD\_UNITS option.

### **/VERSION=nn.nn**

Limits the version number of the product for which you have a license. Use the format *integer.integer*. If your PAK supplies version information, you must enter it exactly as it appears.

## Description

LICENSE REGISTER is the primary LICENSE command. Before you enter a LICENSE REGISTER command, you need a PAK that supplies the information required to enter a license in the License Database.

You can register additional licenses for products that already exist in the License Database. If you register another combinable license in the License Database, LMF combines the license units during a LICENSE LOAD or LICENSE START command. This allows more product availability or activity for the same product.

## LICENSE LICENSE REGISTER

The checksum number supplied with your PAK is calculated from the other information supplied with the PAK. Thus, you must enter each qualifier necessary to supply information from your particular PAK. If you enter LICENSE REGISTER without a required qualifier, LMF returns a checksum error.

### Examples

1. 

```
$ LICENSE REGISTER FORTRAN /ISSUER=DEC /AUTHORIZATION=USA-10 -
_ $ /PRODUCER=DEC /UNITS=400 /VERSION=5.4 -
_ $ /AVAILABILITY=F /CHECKSUM=1-HIDN-INDA-COMP-DAHH
```

This command adds the license for the product Fortran to the default License Database. Fortran becomes licensed using the availability formula with 400 license units available.

2. 

```
$ LICENSE REGISTER DVNETRTG /ISSUER=DEC /AUTHORIZATION=USA-15 -
_ $ /PRODUCER=DEC /UNITS=1000 /VERSION=4.0 -
_ $ /AVAILABILITY=E/CHECKSUM=1-COOD-AGON-EFIC-HING
```

This command adds the license for the product DVNETRTG (DECnet for OpenVMS Routing) to the default License Database. In the example, DVNETRTG is licensed using the availability formula with 1000 license units.

## LICENSE LICENSE START

---

### LICENSE START

This command loads all licenses that are registered and enabled in the License Database into memory. On OpenVMS Alpha and VAX systems, it sets up the License Unit Requirement Table (LURT) for your system. On OpenVMS Integrity server systems, it loads the operating environment table and all per core licenses into memory. Because the OpenVMS operating system issues a LICENSE START command during system startup, you should need this command only if system startup fails.

To use this command, you need CMKRNL, SYSNAM, and SYSPRV privileges on OpenVMS Alpha systems. In addition to those three, you also need SYSLCK privilege on OpenVMS Integrity server systems.

To load the licenses in the License Database of a system with LMF already started, use LICENSE LOAD.

#### Format

```
LICENSE START
```

#### Parameters

None.

#### Qualifiers

##### **/DATABASE=filespec**

Specifies the location of the License Database. The default file specification is defined by the logical name LMF\$LICENSE, which points to SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB on an unmodified OpenVMS system. Use this optional qualifier only if you do not use the default database.

##### **/LOG (default)**

##### **/NOLOG**

Controls whether LICENSE START displays a message to acknowledge the loading of each product license.

#### Example

```
$ LICENSE START
```

On OpenVMS Alpha and VAX systems, this command sets up the LURT for your system and loads all the licenses that are registered and enabled in the License Database. On OpenVMS Integrity server systems, this command loads the operating environment table and all PCL licenses into memory.



---

## LICENSE UNLOAD

Unloads a license, making the product unavailable from the current node. The product license or licenses must be registered in the License Database and must have been previously loaded with an interactive or automatic LICENSE LOAD command. Running processes are allowed to continue to completion.

To use this command, you need CMKRNL, SYSNAM, and SYSPRV privileges.

### Format

```
LICENSE UNLOAD product-name[,...]
```

### Parameter

**product-name[,...]**

Name of the product to be unloaded. You can unload only licenses that have been loaded. Enter each product name exactly as it appears on its Product Authorization Key (PAK). You cannot use wildcard characters for **product-name**.

### Qualifiers

**/LOG**

**/NOLOG (default)**

Controls whether LICENSE UNLOAD lists the name of each unloaded license.

**/PRODUCER=string**

Positional qualifier.

Specifies the name of the company (for example, DEC) that owns the product for which you have a license. The default string for this qualifier on Alpha and VAX is *DEC*. On Integrity servers, the default string is HP. If DEC or HP is not the producer of the product, you must use this qualifier to identify the product. This qualifier affects only the product name that immediately precedes it in the command string. Wildcard characters are not allowed.

### Description

LICENSE UNLOAD affects all units for a single product even if the loaded units are combined from multiple licenses. In such a case, you cannot unload the units from a single license. You must unload all the units.

You are not required to unload a loaded license before modifying data in the License Database. To maximize product availability, modify licenses in the License Database first, and then load the changes by entering a LICENSE UNLOAD command followed by a LICENSE LOAD command.

### Examples

1. \$ LICENSE UNLOAD /PRODUCER=DEC FORTRAN

This command unloads the HP Fortran license on the node from which it is entered.

## LICENSE LICENSE UNLOAD

2. \$ LICENSE UNLOAD PASCAL,FORTRAN

This command unloads the HP Pascal and HP Fortran licenses on the node from which it is entered.

---

## SHOW LICENSE

Displays software product licenses active on the current node and lists the names attached to a license (known as the RESERVE list). The SHOW LICENSE command displays the license database information currently in your system's memory. Use the License Management utility command, LICENSE LIST, when you want to view the license database information that is on disk.

### Format

SHOW LICENSE [product-name [...]]

### Parameter

#### **product-name**

Specifies the name or names of activated software product licenses to display. The asterisk (\*) and the percent sign (%) wildcard characters are allowed. If you do not specify a product name, information is displayed about all active product name licenses. The *product-name* parameter is incompatible with the /UNIT\_REQUIREMENTS qualifier.

### Description

The DCL command SHOW LICENSE displays software product licenses active on the current node. An active license is one that has been registered in the LICENSE database *and* has been loaded into system memory. To register and activate software product licenses, use the License Management utility (LICENSE) or VMSLICENSE.COM. Some licenses are registered automatically during product installation.

To display licenses registered in the LICENSE database, use the LICENSE LIST command.

### Qualifiers

#### **/BEFORE**

Use with /TERMINATION\_DATE and /RELEASE\_DATE qualifiers. Selects only those licenses whose times are before the time specified with the other qualifiers.

The /BEFORE qualifier cannot be used with the /SINCE qualifier.

#### **/BRIEF (default)**

Displays a summary of information about the specified active product licenses. Use the /FULL qualifier to obtain a complete product license listing.

#### **/CHARGE\_TABLE**

Synonym for the /UNIT\_REQUIREMENTS qualifier.

#### **/CLUSTER**

Use with the /UNIT\_REQUIREMENTS qualifier to display the license unit requirements for every node in an OpenVMS Cluster.

#### **/EXACT**

Use with the /PAGE=SAVE and /SEARCH qualifiers to specify a search string that must match the search string exactly and must be enclosed with quotation marks (" ").

## LICENSE

### SHOW LICENSE

If you specify the `/EXACT` qualifier without the `/SEARCH` qualifier, exact search mode is enabled when you set the search string with the Find (E1) key.

#### **/FULL**

Displays a summary of information about the specified active product licenses, including Product Authorization Key (PAK) options and the reserve list (if any). On Integrity server systems, lists the licenses for OEs currently active on the system.

#### **/HIERARCHY - Integrity servers only**

Displays the hierarchy of licenses for operating environments active on the current node.

#### **/HIGHLIGHT[=keyword]**

Use with the `/PAGE=SAVE` and `/SEARCH` qualifiers to specify the type of highlighting you want when a search string is found. When a string is found, the entire line is highlighted. You can use the following keywords: BOLD, BLINK, REVERSE, and UNDERLINE. BOLD is the default highlighting.

#### **/OE[=OE name] - Integrity servers only**

When an OE name is specified, displays the contents of the named operating environment. Currently, valid OE names are BOE and HA-OE. When no OE name is specified, displays the operating environment currently active on the node.

#### **/OUTPUT[=filespec]**

##### **/NOOUTPUT**

Controls where the output of the SHOW LICENSE command is sent. By default, the output of the SHOW LICENSE command is sent to the current `SYS$OUTPUT` device (usually your terminal). To send the output to a file, use the `/OUTPUT` qualifier followed by a file specification.

The asterisk (\*) and the percent sign (%) wildcard characters are not allowed in the file specification. If you enter a partial file specification (for example, specifying only a directory), SHOW is the default file name and .LIS is the default file type.

If you enter the `/NOOUTPUT` qualifier, output is suppressed.

#### **/PAGE[=keyword]**

##### **/NOPAGE (default)**

Controls the display of license information on the screen.

You can use the following keywords with the `/PAGE` qualifier:

<code>CLEAR_SCREEN</code>	Clears the screen before each page is displayed.
<code>SCROLL</code>	Displays information one line at a time.
<code>SAVE[=n]</code>	Enables screen navigation of information, where <i>n</i> is the number of pages to store.

The `/PAGE=SAVE` qualifier allows you to navigate through screens of information. The `/PAGE=SAVE` qualifier stores up to 5 screens of up to 255 columns of information. When you use the `/PAGE=SAVE` qualifier, you can use the following keys to navigate through the information:

Key Sequence	Description
Up arrow key, Ctrl/B	Scroll up one line.
Down arrow key	Scroll down one line.
Left arrow key	Scroll left one column.
Right arrow key	Scroll right one column.
Find (E1)	Specify a string to find when the information is displayed.
Insert Here (E2)	Scroll right one half screen.
Remove (E3)	Scroll left one half screen.
Select (E4)	Toggle 80/132 column mode.
Prev Screen (E5)	Get the previous page of information.
Next Screen (E6), Return, Enter, Space	Get the next page of information.
F10, Ctrl/Z	Exit. (Some utilities define these differently.)
Help (F15)	Display utility help text.
Do (F16)	Toggle the display to oldest/newest page.
Ctrl/W	Refresh the display.

The /PAGE qualifier is not compatible with the /OUTPUT qualifier.

**/PRODUCER=producer-name**

Displays software product licenses active on the current node and supplied by the specified producer. The asterisk (\*) and the percent sign (%) wildcard characters are allowed for the *producer-name* parameter. You cannot use the /PRODUCER qualifier with the /UNIT\_REQUIREMENTS qualifier.

**For HP Products Only**

On OpenVMS VAX and Alpha systems, the producer is shown as DEC. On OpenVMS Integrity server systems, the producer is shown as HP.

**/RELEASE\_DATE=[date\_time]**

Allows listing licenses using release dates as selection criteria.

**/SEARCH="string"**

Use with the /PAGE=SAVE qualifier to specify a string that you want to find in the information being displayed. Quotation marks are required for the /SEARCH qualifier, if you include spaces in the text string.

You can also dynamically change the search string by pressing the Find key (E1) while the information is being displayed. Quotation marks are not required for a dynamic search.

**/SINCE(default)**

Use with the /TERMINATION\_DATE and /RELEASE\_DATE qualifiers. Selects only those licenses whose times are on or after the time specified with the other qualifiers.

The /SINCE qualifier cannot be used with the /BEFORE qualifier.

## LICENSE

### SHOW LICENSE

#### **/TERMINATION\_DATE=date\_time**

Allows listing licenses using termination dates as selection criteria.

#### **/UNIT\_REQUIREMENTS**

On Alpha and VAX systems, displays information in the License Unit Requirement Table (LURT). On Integrity server systems, displays information about the type of system, the number of CPUs active, and the number of sockets. The **/UNIT\_REQUIREMENTS** qualifier is incompatible with the *product-name* parameter and with the **/BRIEF** and **/PRODUCER** qualifiers.

#### **/USAGE**

Tells you how many license units are loaded, how many are currently allocated, and how many are currently available, as well as the license type for each product on the system. Use with the **/FULL** qualifier to display complete information—including the PID, process name, node, or user name—for each instance of use of the product. You need group privilege to see the list of users in your group who have allocated license units; you need world privilege to see the list of users in all groups.

In an OpenVMS Cluster, if you own multiple license types for a single product, you are limited to viewing the usage information for the license type loaded on the node from which you are executing the **SHOW LICENSE/USAGE** command. To find out the usage of the other license type loaded on another node, issue the command on that node. You can also use the System Management (SYSMAN) utility to do this.

In an OpenVMS Cluster, usage information is limited to the local license type. For example, VAX and Alpha availability licenses are considered by LMF to be different license types. If you are running both VAX and Alpha systems in a cluster, usage information for availability licenses is limited to the local system type. For example, if you have DEC C installed on all nodes in your OpenVMS Cluster, you can display DEC C license allocation on all the VAX nodes in the cluster from any VAX node with DEC C installed, but you cannot display the DEC C license allocation on the Alpha nodes.

Usage information is not available for unlimited licenses (a license with 0 units). Clusterwide usage information is not available for personal use or **NO\_SHARE** licenses. Refer to the *HP OpenVMS License Management Utility Manual* for more information on license types.

#### **/WARNING\_INTERVAL=n**

#### **NOWARNING\_INTERVAL**

Displays a warning stating the number of licenses that will terminate in *n* days. The default is 30 days.

#### **/WRAP**

#### **/NOWRAP (default)**

Use with the **/PAGE=SAVE** qualifier to limit the number of columns to the width of the screen and to wrap lines that extend beyond the width of the screen to the next line.

The **/NOWRAP** qualifier extends lines beyond the width of the screen and can be seen when you use the scrolling (left and right) features provided by the **/PAGE=SAVE** qualifier.

## Examples

1. \$ SHOW LICENSE/FULL  
Active licenses on node WTPOOH:  
DVNETEND  
    Producer: DEC  
    Units: 0  
    Version: 0.0  
    Release Date: (none)  
    Termination Date: 31-DEC-2012  
    Availability: 0  
    Activity: 100  
    MOD\_UNITS  
    Product Token:  
  
OPENVMS-ALPHA  
    Producer: DEC  
    Units: 0  
    Version: 0.0  
    Release Date: (none)  
    Termination Date: 31-DEC-2012  
    Availability: 0  
    Activity: 100  
    MOD\_UNITS  
    Product Token:

The SHOW LICENSE command in this example displays all the active licenses on the current Alpha node, WTPOOH.

2. \$ SHOW LICENSE/FULL  
Active licenses on node MACCHU:  
C  
    Producer: HP  
    Units: 3  
    Version: 0.0  
    Release Date: (none)  
    Termination Date: 31-DEC-2012  
    Availability: 0  
    Activity: 1  
    MOD\_UNITS  
    IA64 ALPHA  
    Product Token:  
  
DVNETEXT  
    Producer: HP  
    Units: 4  
    Version: 0.0  
    Release Date: (none)  
    Termination Date: 31-DEC-2012  
    Per Core License  
    Activity: 0  
    IA64  
    Product Token:

## LICENSE

### SHOW LICENSE

```
OPENVMS-I64-BOE
Producer: HP
Units: 4
Version: 0.0
Release Date: (none)
Termination Date: 31-DEC-2012
Per Core License
Activity: 0
IA64
Product Token:
```

The SHOW LICENSE command in this example displays all the active licenses on the current Integrity server node, MACCHU.

#### 3. \$ SHOW LICENSE/BRIEF

Active licenses on node WTPOOH:

Product	Producer	Units Avail	Rating	Activ	Version	Release	Termination
DVNETEND	DEC	0 0	----	100	0.0	(none)	(none)
VAX-VMS	DEC	0 0	----	100	0.0	(none)	(none)

The SHOW LICENSE command in this example displays a summary of all the active licenses on the current VAX node, WTPOOH.

#### 4. \$ SHOW LICENSE/OUTPUT=SYS\$LOGIN:ACTIVE\_LICENSES\_OCT30.DAT

The SHOW LICENSE command in this example writes all the active licenses to the file named SYS\$LOGIN:ACTIVE\_LICENSES\_OCT30.DAT.

#### 5. \$ SHOW LICENSE/FULL PERSONAL

Active licenses on node PICCHU:

```
PERSONAL
Producer: DEC
Units: 100
Version: 0.0
Release Date: (none)
Termination Date: (none)
Availability: 0
Activity: 100
RESERVE_UNITS
Reserve: RANCE
```

The SHOW LICENSE command in this example displays information about the product PERSONAL, as well as the name RANCE attached to the product license (known as the RESERVE list).

#### 6. \$ SHOW LICENSE/TERM=10-JAN-2014 test0%

Active licenses on node PICCHU:

Product	Producer	Units Avail	Rating	Activ	Version	Release	Termination
TEST01	DEC	0 A	----	0	0.0	(none)	(none)
TEST02	DEC	0 B	----	0	0.0	10-JAN-2014	12-NOV-2014
TEST03	DEC	0 C	----	0	0.0	30-DEC-2014	(none)
TEST04	DEC	0 D	----	0	0.0	(none)	25-AUG-2015
TEST05	DEC	0 E	----	0	0.0	14-NOV-2016	14-AUG-2016

```
$ SHOW LICENSE/RELEASE=10-JAN-2014/SINCE test0%
```

Active licenses on node PICCHU:



## LICENSE SHOW LICENSE

```

--- Product ID ----          ---- Rating ----- -- Version --
Product      Producer      Units Avail Activ Version Release Termination
TEST02       DEC                0   B    0    0.0 10-JAN-2014 12-NOV-2014
TEST03       DEC                0   C    0    0.0 30-DEC-2014 (none)
TEST05       DEC                0   E    0    0.0 14-NOV-2016 14-AUG-2016

```

\$ SHOW LICENSE/RELEASE=10-JAN-2014/BEFORE test0%

Active licenses on node PICCHU:

```

--- Product ID ----          ---- Rating ----- -- Version --
Product      Producer      Units Avail Activ Version Release Termination
TEST01       DEC                0   A    0    0.0 (none)      (none)
TEST04       DEC                0   D    0    0.0 (none)      25-AUG-2015

```

In these examples, the SHOW LICENSE command uses the /TERM, /RELEASE, /SINCE and /BEFORE qualifiers.

### 7. \$ SHOW LICENSE/UNIT\_REQUIREMENTS

VMS/LMF Charge Information for node PICCHU

This is a AlphaServer 8400 5/440, hardware model type 1567

```

Type: A, Units Required: 2700 (VAX/VMS Capacity or OpenVMS Unlimited or Base)
Type: B, * Not Permitted * (VAX/VMS F&A Server)
Type: C, * Not Permitted * (VAX/VMS Concurrent User)
Type: D, * Not Permitted * (VAX/VMS Workstation)
Type: E, * Not Permitted * (VAX/VMS System Integrated Products)
Type: F, * Not Permitted * (VAX Layered Products)
Type: G, * Not Permitted * (Reserved)
Type: H, Units Required: 1150 (Alpha Layered Products)
Type: I, Units Required: 1150 (Layered Products)

```

In this example, the /UNIT\_REQUIREMENTS qualifier displays information in the License Unit Requirement Table (LURT) for the Alpha node PICCHU.

### 8. \$ SHOW LICENSE/CHARGE TABLE

OpenVMS I64/LMF Charge Information for node MACCHU

This is an HP rx2600(900MHz/1.5MB), with 2 CPUs active, 2 socket(s)

Type: PPL, Units Required: 2 (I64 Per Processor)

Type: PCL, Units Required: 2 (I64 Per Core)

This example displays the CHARGE\_TABLE information for an Integrity server node MACCHU with two active processor cores.

### 9. \$ SHOW LICENSE/CHAR/CLUSTER

VMS/LMF Cluster License Unit Requirements Information

14-MAR-2010 06:39:41.54

Node	A	B	C	D	E	F	G	H	I	PCL
FISH	20	-	-	-	-	-	-	1050	1050	-
SWORD	15	-	-	-	-	-	-	1050	1050	-
SALMON	12	-	-	-	-	-	-	1050	1050	-
MONGER	12	-	-	-	-	-	-	1050	1050	-
GORDON	15	-	-	-	-	-	-	1050	1050	-
ARTIST	-	-	-	-	-	-	-	-	-	2
PAINTS	-	-	-	-	-	-	-	-	-	1

# LICENSE

## SHOW LICENSE

```

Total Cluster Unit Requirements
Type: A, Units Required: 74 (VAX/VMS Capacity or OpenVMS Unlimited or Base)
Type: B, * Not Permitted * (VAX/VMS F&A Server)
Type: C, * Not Permitted * (VAX/VMS Concurrent User)
Type: D, * Not Permitted * (VAX/VMS Workstation)
Type: E, * Not Permitted * (VAX/VMS System Integrated Products)
Type: F, * Not Permitted * (VAX Layered Products)
Type: G, * Not Permitted * (Reserved)
Type: H, Units Required: 5250 (Alpha Layered Products)
Type: I, Units Required: 5250 (Layered Products)

Type: PPL, Units Required: 3 (I64 Per Processor)
Type: PCL, Units Required: 3 (I64 Per Core)

```

In this example, the display shows how many license units are required for each license type (A, B, etc. on Alpha and VAX and PCL on Integrity servers) on each node in the cluster. If a row of three asterisks (\*\*\*) is displayed for a node, it means that the node is in the process of booting.

### 10. \$ SHOW LICENSE/OE

```

Current Operating Environment on node MACCHU at 8-MAR-2010 16:12:51.72

----- Operating Environment ----- Units -----
Name      Description      Type Level  Loaded  Total
HAOE      High Availability  H    5      4      4

```

This example shows the currently operating environment (HAOE) on an Integrity server node MACCHU.

### 11. \$ SHOW LICENSE/HIER/FULL

```

Operating Environment Hierarchy
----- Operating Environment ----- Units -----
Name      Description      Type Level  Loaded  Total
HAOE      High Availability  H    5      2      2
  GWLM
MCOE      Mission Critical  H    4      -      2
  RTR-SVR
  VMSCUSTER
  VMSCUSTER-CLIENT
EOE      Enterprise      H    3      -      2
  AVAIL-MAN
  RMSJNL
  VOLSHAD
BOE      Base            H    2      -      2
  DEGRAM
  OMS
FOE      Foundation      H    1      -      2
  OPENVMS-I64
  OPENVMS-USER
  DVNETEND
  DW-MOTIF
  UCX
  TDC
  X500-ADMIN-FACILITY
  X500-DIRECTORY-SERVER
  CIFS

```

This example displays information about the available operating environments, the hierarchy among them, and the products contained in each OE on an Integrity servers system.

## LICENSE SHOW LICENSE

12. \$ SHOW LICENSE/OE=BOE/FULL

```

----- Operating Environment ----- Units -----
Name      Description          Type Level  Loaded   Total
BOE       Base                       H       2       4       7
  DEGRAM
  OMS
  OPENVMS-I64
  OPENVMS-USER
  DVNETEND
  DW-MOTIF
  UCX
  TDC
  X500-ADMIN-FACILITY
  X500-DIRECTORY-SERVER
  CIFS

```

This example shows all the products included in the Base Operating Environment (BOE) on an Integrity server node.

13. \$ SHOW LICENSE OPENVMS-I64-HAOE

Active licenses on node MACCHU:

```

----- Product ID -----      ---- Rating ----- -- Version --
Product      Producer      Units PCL Activ Version Release  Termination
OPENVMS-I64-HAOE  HP              4   1   0       0.0 (none)  10-MAR-2011

```

This example shows licensing information for the HA-OE environment currently active on an Integrity server node MACCHU.

14. \$ SHOW LICENSE/WARNING\_INTERVAL=8000 test0%

Active licenses on node PICCHU:

```

--- Product ID ---      ---- Rating ----- -- Version --
Product      Producer      Units Avail Activ Version Release  Termination
TEST01       DEC              0   A   0       0.0 (none)  (none)
TEST02       DEC              0   B   0       0.0 10-JAN-2014 12-NOV-2014
TEST03       DEC              0   C   0       0.0 30-DEC-2014 (none)
TEST04       DEC              0   D   0       0.0 (none)  25-AUG-2015
TEST05       DEC              0   E   0       0.0 14-NOV-2016 14-AUG-2016
%SHOW-I-TERMIMM, 3 licenses will terminate in 8000 days

```

The /WARNING\_INTERVAL qualifier in this example displays three licenses that will terminate in 8000 days.

15. \$ SHOW LICENSE/USAGE/FULL DECWRITE-USER

View of loaded licenses from node SLTG24 29-DEC-2001 13:36:22.23

ACTIVITY license DECWRITE-USER usage information:

```

Pid          Process Name      Units  Username      Node
416000E6     MACAHAY           100    MACAHAY       SLTG24
416000E7     MACAHIGH          100    MACAHIGH      SLTG24
416000E8     ALICE              100    ALICE         SLTG24
416000E9     MORGEN            100    MORGEN        SLTG24
416000F1     ANGEL              100    ANGEL         SLTG24
416000F2     ANGEL_1           100    ANGEL         SLTG24

```

Units loaded: 2000 Units allocated: 600 Units available: 1400

The SHOW LICENSE command in this example lists the current users of the activity license for the product DECwrite. For each instance of use of the product, the process identification (PID), process name, node, and user name are identified. The units column shows the number of units allocated for each particular invocation of the product. The last line displays the units loaded

## LICENSE SHOW LICENSE

when the LICENSE LOAD command was given, the total number of units currently allocated, and the total of unused (available for others to use) units.

### 16. \$ SHOW LICENSE/USAGE/FULL TEST\_PER

View of loaded licenses from node: SLTG24 30-DEC-2001 15:45:59

PERSONAL USE license DEC TEST\_PER usage information:

Units	Reserved for:
100	UNCLE
100	AUNT
100	NEPHEW
100	NIECE

Units loaded: 600 Units reserved: 400 Units available: 200

This example shows a personal use license. The DEC TEST\_PER product has enough units for six reservations with 100 units for each reservation. The license database (LDB) only has a total of four names in the reserve list attached to this product. If the license administrator (usually the system manager) wants to take full advantage of this license and adds 2 more names to the reserve list, he should use the following commands to update the product information:

```
$ LICENSE MODIFY TEST_PER/RESERVE=(NAME, ANOTHER_NAME)/ADD
$ LICENSE UNLOAD TEST_PER
$ LICENSE LOAD TEST_PER
```

If this product is used in a cluster environment, you may use the SYSMAN utility to unload and load the license.

### 17. \$ SHOW LICENSE/USAGE/FULL TEST\_CAP

View of loaded licenses from node: SLTG24 30-DEC-2001 15:45:59

Availability license DEC TEST\_CAP usage information:

Units	Node
10	SLTG24
10	SLTG43
600	TORN80
600	LTNUP

Units loaded: 620 Units allocated: 1220 Units available: \*\*\*

In this example, the number of units allocated appears to be greater than the total units loaded and the units available value is three asterisks (\*\*\*).

When you see three asterisks (\*\*\*) as the number of units available, it is generally not a cause for alarm. This situation might arise when the license database (LDB) has been updated on disk, but the new information has not been propagated to the license database in memory on all nodes in the cluster. This node, SLTG24, happens to be one of the nodes that has not received the latest LDB information.

To update the information in the license database in memory for the TEST\_CAP product, enter the following commands:

```
$ LICENSE UNLOAD TEST_CAP
$ LICENSE LOAD TEST_CAP
```

The next time you issue the SHOW LICENSE/USAGE command the three asterisks (\*\*\*) in display should disappear. If, however, you are using multiple LDB files in a cluster, you should read the section on the license database in the *HP OpenVMS License Management Utility Manual*.

## LICENSE SHOW LICENSE

### 18. \$ SHOW LICENSE/UNIT\_REQUIREMENT/CLUSTER

```
VMS/LMF Cluster License Unit Requirements Information
                                24-DEC-2001 14:05:51.65

Node      A      B      C      D      E      F      G      H      I
KARBO     -      -      -     100     50     10     -      -     10
JENJON    -      -      -     100     50     10     -      -     10
HELENA    143    -      -      -     600    2400   -      -    2400
SHAKTI    -      -      -     100     50     10     -      -     10
Total Cluster Unit Requirements
Type: A, Units Required: 143      (VMS Capacity)
Type: B, * Not Permitted *       (VMS Server)
Type: C, * Not Permitted *       (VMS Concurrent User)
Type: D, Units Required: 300     (VMS Workstation)
Type: E, Units Required: 750     (System Integrated Products)
Type: F, Units Required: 2430    (Layered Products)
Type: G, * Not Permitted *       (VMS Reserved)
Type: H, * Not Permitted *       (Alpha Layered Products)
Type: I, Units Required: 2430    (Layered Products)
```

In this example, the display shows how many license units are required for each license type (A, B, etc.) on each node in the cluster. If a row of three asterisks (\*\*\*) is displayed for a node, it means that the node is in the process of booting.

### 19. \$ SHOW LICENSE/USAGE

```
View of loaded licenses from node REDSOX      8-MAR-2010 16:20:11.14

----- Product ID ---- ---- Unit usage information -----
Product      Producer Loaded Allocated Available Compliance
C             HP           250      0       250      Yes
DVNETEXT     HP           4         3        1       Yes
OPENVMS-I64-BOE HP         2         2        0       Yes
OPENVMS-I64-HAOE HP         20        8        12      Yes
VAXSET       HP           10        8         2       Yes
```

This example shows how many license units are loaded, how many are currently allocated, and how many are available on REDSOX, an Integrity servers system. The last column in the display shows that are products are in compliance with their license unit requirements.

### 20. \$ SHOW LICENSE/USAGE

```
View of loaded licenses from node HOVMS2      8-MAR-2010 08:38:17.13
----- Product ID ----- ---- Unit usage information -----
Product      Producer Loaded Allocated Available Compliance
OPENVMS-I64-HAOE HP      Virtual Machine guest, no usage information
```

Issuing the SHOW LICENSE/USAGE command from an OpenVMS guest cluster member displays the text "Virtual Machine guest, no usage information" for PCL licenses loaded on the system. There is essentially no usage charge against the license units for OpenVMS guest nodes since multiple guests can run on the same host using the same license units.



This appendix provides the following examples of license-related management tasks using the LICENSE commands:

- Registering a system integrated product (Section B.1)
- Issuing LICENSE LIST and SHOW LICENSE commands (Section B.2)
- Restricting product use (Section B.3)

### B.1 Registering a System Integrated Product

The following example provides a step-by-step procedure for registering a PAK for Volume Shadowing for OpenVMS, which is a System Integrated Product (SIP). This example uses command procedure VMSLICENSE.COM and the following PAK information:

```
ISSUER: DEC
AUTHORIZATION NUMBER: ALS-WM-45789-6666
PRODUCT NAME: VOLSHAD
PRODUCER: DEC
NUMBER OF UNITS: 400
VERSION: 7.3
PRODUCT RELEASE DATE:
KEY TERMINATION DATE: 31-DEC-2001
AVAILABILITY TABLE CODE: E
ACTIVITY TABLE CODE:
KEY OPTIONS: MOD_UNITS
PRODUCT TOKEN:
HARDWARE I.D.:
CHECKSUM: 2-EBID-GOOD-NIGH-OJGG
```

## Examples

### B.1 Registering a System Integrated Product

Execute the VMSLICENSE.COM command procedure to register the Volume Shadowing PAK as follows:

1. Log in to the system manager's account, SYSTEM. Enter the command:

```
$ @SYS$UPDATE:VMSLICENSE
```

The procedure displays the following menu:

```
VMS License Management Utility Options:
```

1. REGISTER a Product Authorization Key
2. AMEND an existing Product Authorization Key
3. CANCEL an existing Product Authorization Key
4. LIST the Product Authorization Keys
5. MODIFY an existing Product Authorization Key
6. DISABLE an existing Product Authorization Key
7. DELETE an existing Product Authorization Key
8. COPY an existing Product Authorization Key
9. MOVE an existing Product Authorization Key
10. ENABLE an existing Product Authorization Key
11. SHOW the licenses loaded on this node
12. SHOW the unit requirements for this node
99. EXIT this procedure

```
Type '?' at any prompt for a description of the information requested. Press Ctrl/Z at any prompt to return to the main menu.
```

```
Enter one of the above choices [1]
```

2. Enter 1. The procedure displays the following message:

```
* Do you have your Product Authorization Key? [YES]:
```

3. Enter Y. The procedure displays the following information and prompts:

```
Use the REGISTER option to add a new license to a license database. A Product Authorization Key (PAK) provides the product name and information you need to register the license. You must enter all the information provided by your PAK exactly as it appears.
```

```
Type '?' at any prompt for a description of the information requested. Press CTRL/Z at any prompt to return to the main menu.
```

```
Issuer [DEC]:
```

```
Authorization Number []:
```

If you do not have your PAK information at hand, you cannot continue at this point.

4. Press Return to specify DEC as the issuer.
5. Enter the authorization number from the PAK, ALS-WM-45789-6666. The procedure prompts for the following information:

```
Product Name []:
```

6. Enter the product name string, VOLSHAD from the PAK. The procedure prompts for the producer:

```
Producer [DEC]:
```

7. Press Return to specify DEC as the producer. The procedure prompts for the number of units:

```
Number of Units []:
```



**B.1 Registering a System Integrated Product**

8. Enter the number of units, 400. The procedure prompts for the version:

Version []:

9. Enter the version number from the PAK, 7.3. The procedure prompts for the key termination date:

Key Termination Date []:

10. Enter the key termination date, 31-DEC-2001. The procedure prompts for the following information:

Availability Table Code []:

Activity Table Code []:

11. Enter the availability table code, E. Press Return after the Activity Table Code prompt. The procedure prompts for the following information:

Key Options []:

Product Token []:

Hardware-Id []:

12. Enter the option MOD\_UNITS after the Key Options prompt. Press Return after the Product Token prompt and the Hardware-ID prompt. The procedure prompts for the checksum:

Checksum []:

13. Enter the checksum, 1-EBID-GOOD-NIGH-OJJG.

---

**Note**

---

The checksum string always begins with a number. The other 16 characters are always alphabetic characters from A through P.

---

The procedure displays the information you entered. For example:

Here is a list of the license information just entered:

```

      Issuer:  DEC
  Authorization:  ALS-WM-45789-6666
      Producer:  DEC
   Product Name:  VOLSHAD
         Units:  400
   Release Date:
      Version:   7.3
  Termination Date:  31-DEC-2001
     Availability:  E
       Activity:
         Options:  MOD_UNITS
           Token:
   Hardware ID:
     Checksum:   1-EBID-GOOD-NIGH-OJJG
  
```

Is that correct? [YES]:

14. Compare the information on the screen with the information on the PAK. If the information is correct, enter Y. Otherwise enter N.

## Examples

### B.1 Registering a System Integrated Product

---

#### Note

---

If you enter any of the information incorrectly, you receive an error message, and the license is not registered. A checksum error can result when you enter incorrect information for other items on the PAK. If you get an error, carefully check all the data that you entered.

---

15. When the procedure displays the following question, enter Y.
- Do you wish to make corrections? [YES]:
16. To make corrections, the procedure steps you through all of the questions again with the data you just entered as defaults for each data field. If the data is correct, press Return. To replace incorrect data, enter the new data. To cancel data without entering new data, enter the backslash ( \ ) character.

If you entered all the information correctly, the procedure displays the following message:

```
Registering VOLSHAD license in SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB...
```

17. After the license is successfully registered, the procedure asks if you want to load the license on the current node, as follows:

Do you want to LOAD this license on this system? [YES]:

- If you registered the PAK on a standalone system and want to make the software available (active) immediately, enter Y.
- If you registered the license in an OpenVMS Cluster environment but do not want to make it available (active) on the current node, enter N.

In this example, assume the license is being registered in an OpenVMS Cluster environment and that you do not want it loaded on the current node. Enter N to complete this license registration. Note that you must load the Volume Shadowing license before you can use the product. See Step 18.

The procedure returns you to the first menu and prompt as follows:

```
VMS License Management Utility Options:
```

1. REGISTER a Product Authorization Key
2. AMEND an existing Product Authorization Key
3. CANCEL an existing Product Authorization Key
4. LIST the Product Authorization Keys
5. MODIFY an existing Product Authorization Key
6. DISABLE an existing Product Authorization Key
7. DELETE an existing Product Authorization Key
8. COPY an existing Product Authorization Key
9. MOVE an existing Product Authorization Key
10. ENABLE an existing Product Authorization Key
11. SHOW the licenses loaded on this node
12. SHOW the unit requirements for this node
  
99. EXIT this procedure

```
Type '?' at any prompt for a description of the information requested. Press Ctrl/Z at any prompt to return to the main menu.
```

```
Enter one of the above choices [1]
```

18. To register another PAK, enter 1. Then respond to the questions, again entering information from a license PAK. For this example assume you are finished. Enter 99 to exit the procedure. You have registered the license for Volume Shadowing.

## B.1 Registering a System Integrated Product

19. Load the license on the desired node by logging in to that node and entering the LICENSE LOAD VOLSHAD command at the DCL prompt (\$).

## B.2 Difference Between LICENSE LIST and SHOW LICENSE

This example shows the differences between the LICENSE LIST command, which displays license information stored on disk, and the SHOW LICENSE command, which displays license information stored in memory. The first command registers a Fortran license, as follows:

```
$ LICENSE REGISTER FORTRAN /ISSUER=DEC /AUTHORIZATION=USA-10 -
_$ /PRODUCER=DEC /UNITS=400 /VERSION=6.0 -
_$ /AVAILABILITY=F /CHECKSUM=1-HIDN-INDA-COMP-DAHH
```

This command adds the license for the product Compaq Fortran to the default License Database. To see the information registered in the database, enter a LICENSE LIST command, as follows:

```
$ LICENSE LIST /FULL FORTRAN
Use Ctrl/Z to exit, PF3-PF4 for Previous-Next Screen and Arrow keys to Scroll.
```

```
License Management Facility V1.2
```

```
License Database File:      SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB
Created on:                 17-AUG-2000
Created by user:           GRAHAM
Created by LMF Version:    V1.2
```

```
-----
Issuer:                     DEC
Authorization:              USA-10
Product Name:               FORTRAN
Producer:                   DEC
Units:                      400
Version:                    6.0
Release Date:               (none)
PAK Termination Date:      (none)
Availability:               F (Layered Products)
Activity:                   0
Options:
Hardware ID:

Revision Level:             1
Status:                     Active
Command:                    REGISTER
Modified by user:          LESH
Modified on:                21-AUG-2000 14:32:23.41
```

Notice that for status the LICENSE LIST command displays *Active*. This means the registered license is enabled for loading, that it has not been disabled or canceled. It does not necessarily mean the Fortran license was loaded with a LICENSE LOAD command. Because the LICENSE LIST command sees only the License Database on disk, you must enter a SHOW LICENSE command to refer to the License Database in memory to determine whether a license is active on the current system. For example:

```
$ SHOW LICENSE/FULL
Active licenses on node BIODTL:
```

## Examples

### B.2 Difference Between LICENSE LIST and SHOW LICENSE

```
CRYPTICALMENT
  Producer: DEC
  Units: 400
  Version: 7.3
  Release Date: (none)
  Termination Date: 31-DEC-2001
  Availability: E (System Integrated Products)
  Activity: 0
  MOD_UNITS
```

```
VAX-VMS
  Producer: DEC
  Units: 400
  Version: 6.0
  Release Date: (none)
  Termination Date: (none)
  Availability: A (VMS Capacity)
  Activity: 0
  MOD_UNITS
  NO_SHARE
```

The SHOW LICENSE command in this example displays all the active licensed products on the current node named BIODTL; the Fortran license has not yet been loaded.

After you load the Fortran LICENSE, the SHOW LICENSE command displays the license. For example:

```
$ LICENSE LOAD FORTRAN
LICENSE-I-LOADED, DEC FORTRAN was successfully loaded with 400 units
$ SHOW LICENSE /FULL
```

Active licenses on node BIODTL:

```
CRYPTICALMENT
  Producer: DEC
  Units: 400
  Version: 7.3
  Release Date: (none)
  Termination Date: 31-DEC-2001
  Availability: E (System Integrated Products)
  Activity: 0
  MOD_UNITS
```

```
FORTRAN
  Producer: DEC
  Units: 400
  Version: 6.0
  Release Date: (none)
  Termination Date: (none)
  Availability: F (Layered Products)
  Activity: 0
```

```
VAX-VMS
  Producer: DEC
  Units: 400
  Version: 6.0
  Release Date: (none)
  Termination Date: (none)
  Availability: A (VMS Capacity)
  Activity: 0
  MOD_UNITS
  NO_SHARE
```

## B.3 Restricting Product Use

This example illustrates how LMF restricts use of a product when insufficient license units are registered for it. In this example, the product DEC BASIC is installed and its license is registered with zero availability units and the MOD\_UNITS option. Zero-unit licenses provide authorization for that product's use on any processor. In the first LICENSE MODIFY command, however, the license is changed to a 1500-unit Availability License:

```
$ LICENSE UNLOAD BASIC
$ LICENSE MODIFY/UNITS=1500 BASIC
```

The next command attempts to load the registered license on a system:

```
$ LICENSE LOAD BASIC
%LICENSE-W-NOLOAD, license was not loaded for BASIC
-LICENSE-F-EXCEEDED, attempted usage exceeds active license limits
```

Because the VAX 6000 system in this example requires 2400 license units to authorize DEC BASIC, the LICENSE LOAD command fails. The next command attempts to invoke DEC BASIC despite the failed LICENSE LOAD command:

```
$ BASIC
%LICENSE-F-NOAUTH, DEC BASIC use is not authorized on this node
-LICENSE-F-NOLICENSE, no license is active for this software product
-LICENSE-I-SYSMGR, please see your system manager
```

Note that the attempt to invoke DEC BASIC fails. Because the LICENSE LOAD command failed, DEC BASIC use is unauthorized on the current node. The solution is to modify the license again using a value obtained by using the DCL command SHOW LICENSE /UNIT\_REQUIREMENTS.

```
$ SHOW LICENSE/UNIT_REQUIREMENTS
VMS/LMF Charge Information for node JANINA
This is a VAX 6000-540, hardware model type 188
Type: A, Units Required: 170 (VAX/VMS Capacity or OpenVMS Unlimited or Base)
Type: B, * Not Permitted * (VAX/VMS F&A Server)
Type: C, * Not Permitted * (VAX/VMS Concurrent User)
Type: D, * Not Permitted * (VAX/VMS Workstation)
Type: E, Units Required: 600 (VAX/VMS System Integrated Products)
Type: F, Units Required: 2400 (VAX Layered Products)
Type: G, * Not Permitted * (Reserved)
Type: H, * Not Permitted * (Alpha Layered Products)
Type: I, Units Required: 2400 (Layered Products)
```

Issuing the SHOW LICENSE/UNIT\_REQUIREMENTS shows that the required number of license units is 2400. Change the number of units with the LICENSE MODIFY command to provide sufficient units to allow a successful LICENSE LOAD command, which authorizes use of the product.

```
$ LICENSE MODIFY/UNITS=2400 BASIC
$ LICENSE LOAD BASIC
%LICENSE-I-LOADED, DEC BASIC was successfully loaded with 2400 units
```

After the license is loaded, the product can be invoked, as follows:

```
$ BASIC
DEC BASIC V3.2
Ready
```



---

# Glossary

This glossary defines the LMF-related terms used in the *HP OpenVMS License Management Utility Manual*.

## **active license**

A license that has been enabled. The term *active* appears in displays produced by LICENSE LIST and has been retained to prevent automated procedures from breaking.

## **Activity License**

A license that defines the allowed number of concurrent uses of a product. Each product defines an activity as either an interactive user, a running process, or a job. For example, a 4-Activity License may 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. LMF can activate a product when the number of license units on a license matches or exceeds the license unit rating for the current processor. Every System Marketing Model (SMM) has a series of license unit requirements, typically related to performance, that define how many license units are required to make a product available.

## **checksum**

An encoded number calculated from the other information supplied with a PAK. The checksum string always begins with the number 1, which is the only number in the string. The other 16 characters are always alphabetic characters from A through P.

## **core**

The actual data-processing engine within a cell-based processor. A single processor can contain multiple cores.

## **hardware identifier**

An optional string that identifies a particular hardware unit.

## **Integrated Software Business Technologies**

The name for HP's business plan that integrates consolidated software distribution, online documentation, and software access management. With this plan more products will be available on compact disc read-only memory (CD-ROM) where software access is authorized by PAKs and LMF.

### **license**

(In this manual) PAK information for a software product that is registered in the License Database.

### **license combination**

A method for using the license units from two or more combinable licenses to provide more product availability. Two licenses with 100 units each combine to equal a 200-unit license. You may use license combination, for example, when you add a new processor to a VAXcluster environment.

### **License Database**

A collection of interrelated data stored on a disk and accessed through LMF software. The default location for the database is SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB. Each record in the License Database corresponds to a license. Sometimes OpenVMS licenses are registered in a second License Database located in SYS\$SPECIFIC:[SYSEXE]LMF\$SYSTEM.LDB.

### **License Management Facility (LMF)**

See LMF.

### **license registration**

The entry in the License Database of a Product Authorization Key (PAK) to add a new license. To register a license, enter the LICENSE REGISTER command, or respond to prompts from the VMSLICENSE.COM command procedure.

### **license sharing**

A method to allow more than one processor to use the license units from a single license. In OpenVMS, this refers to sharing licenses among nodes in an OpenVMS Cluster environment. Licenses that specify the NO\_SHARE option cannot be shared.

### **license unit**

A basic measurement that HP uses to specify how much product use a license provides. HP 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 200-unit license, or a 700-unit license.

### **License Unit Requirement Table (LURT)**

See LURT.

### **LMF**

License Management Facility. A variety of system-level software components used to maintain software license information in the License Database of the OpenVMS operating system. LMF is a management tool; the terms and conditions of your product contract determine your legal use of software.



## **LURT**

License Unit Requirement Table. Online tables provided by HP that specify 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 unit requirements. The default file name for the LMF LURTs is `SYS$COMMON:[SYSEXE]LMF$LURT.DAT`.

## **Operating Environment (OE)**

A collection of products, including the OpenVMS operating system, that are bundled together under a single license. Operating environments (also known as OEs) are tiered in a hierarchy. Each higher-level OE contains everything in the lower tiers plus additional functionality.

## **PAK**

Product Authorization Key. License information that you must register in the License Database in order to use the product. It is produced by a PAK issuer and delivered to you by mail, electronic transfer, or by telephone.

### **PAK issuer**

The LMF name for 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.

## **Per Core License (PCL)**

Formerly Per Processor License (PPL). A per core license authorizes use of a product based on the number of active processor cores on the system. Each active processor core requires one PCL unit. A PCL is required to run an operating environment and many other products on OpenVMS Integrity server systems. LMF monitors active processor cores on a system for compliance.

## **Personal Use License**

A license that designates the names of specific users for unlimited use of a product. Each product defines a user as either an interactive user, a running process, or a job. LMF requires user names associated with this kind of license.

## **processor**

The component that plugs into a processor socket. The processor can contain more than one processor core.

## **processor module**

The packaging of one or more processors to connect into a single socket on a system bus.

## **processor socket**

The system board socket into which a processor attaches.

## **Product Authorization Key (PAK)**

See PAK.

**product identification**

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

**record**

A collection of data fields in the License Database that define a license at any one time.

**release level**

Uniquely identified by either a product release date or product release version. To authorize a product for use by license version number, the product release level (in the form *nn.nn*) must be less than or equal to the license version number. For example, license version number 4.4 allows operation of product release levels 4.3 and 4.4, but not 4.5.

**reservation list**

A list that contains the names of users with authorized access to a product that is registered with a Personal Use License.

**selection weight**

An arbitrary attribute of a license, assigned by LMF, to control the order in which different licenses for a product are loaded. You can modify the selection weight with the /SELECTION\_WEIGHT qualifier to LICENSE MODIFY.

**SMM**

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

**socket**

A receptacle into which a processor module can be installed. Each processor module can contain one or more processor cores. The number of sockets allowed by a license can be specified as an entry in the HARDWARE\_ID field on the PAK.

**software license**

A contract between a license issuer (HP) 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.

**software producer**

The company that owns the software being licensed. Software producers are usually the same as PAK issuers but can operate under agreement with the issuer.

**Software Product Description (SPD)**

See SPD.

**SPD**

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

**System Marketing Model (SMM)**

See SMM.

**termination date**

The date when a license contract is no longer valid, and when LMF no longer authorizes product use.

**token**

A text string specific to each product used to control additional licensing features. HP does not currently use tokens; however, LMF accepts them for use by certain third-party products.

**user license**

The number of users allowed unlimited use of a product. Each product defines a user as either an interactive user, a running process, or a job. LMF requires user names with this kind of license.



## A

---

activity license, 2-7, 3-3  
Adding OpenVMS Guest to a cluster, 4-1  
Availability License  
    restrictions for Alpha and VAX, 2-5

## C

---

Checksum error, avoiding, A-37  
Codes for license types, 2-2, 2-4, A-34  
Combination rules, 2-11  
    dates and version numbers, 2-12  
Commands  
    syntax descriptions, A-1 to B-1  
Controlling license loading order, 5-23  
COPY command, A-2 to A-4  
CREATE command, A-5

## D

---

DELETE command, A-6 to A-8  
DISABLE command, A-9 to A-10

## E

---

ENABLE command, A-11 to A-12

## G

---

Group License, 2-10

## H

---

History records  
    in the License Database, 1-3, 5-3

## I

---

Installing products, 5-1  
    sequence with license registration, 5-6  
ISSUE command, A-13 to A-15

## L

---

License  
    Activity  
        sharing units, 2-9  
    authorization  
        by user, 2-7  
    authorization time, 2-8  
    automatic registration, 5-1  
    combination, 2-6, 2-9, 2-11  
        /COMBINE qualifier, A-25  
        dates and version numbers, 2-12  
        /NOCOMBINE qualifier, A-25  
        NO\_SHARE option, 2-11  
    controlling loading, 5-16, 5-18  
    controlling user access to, 5-21  
    defined, 1-1  
    examples of registration, B-1  
    loading, 5-1, 5-15, A-22, A-38  
    loading in an OpenVMS Cluster environment,  
        5-15  
    managing after registration, 5-17  
    methods for registering, 5-7  
    modifying, 5-2, A-25  
    modifying to include a reservation list, 5-21  
    modifying to include SCS node name, 5-2,  
        5-19  
    MOD\_UNITS option, 5-18  
    multiples of, 5-20  
    multiples with LICENSE LOAD command,  
        A-23  
    multiples with LICENSE UNLOAD, A-39  
    NO\_SHARE option, 5-20  
    OpenVMS guest licensing, 4-1  
    order of checking  
        /SELECTION\_WEIGHT qualifier, A-28  
    product self-registration, 5-7  
    providing activity use, 2-8  
    providing availability in an OpenVMS Cluster  
        environment, 2-6  
    registering a System Integrated Product (SIP),  
        5-4, B-1  
    registration, A-34  
        best time, 5-5  
    registration and product installation, 5-6  
    RESERVE\_UNITS option, A-27

- License (cont'd)
  - restricting access to, 5-18
  - sharing in an OpenVMS Cluster environment, 2-4, 5-19
  - types, 2-4 to 2-10
    - activity, 2-7
    - Availability, 2-4
    - Group, 2-10
    - Personal Use, 2-9
  - unlimited units, 2-3, 5-18
  - unloading, A-39
    - on shutdown, 5-19
  - Virtual licensing, 4-1
  - with the NO\_SHARE option, 5-2
  - zero units, 2-3, 5-18
- License checking
  - specifying order of, A-28
- LICENSE commands
  - entering long, A-37
  - list of, A-1
- License database, 1-2, 5-1, 6-2
  - backing up, 5-2
  - common, with multiple system disks, 5-2
  - creating, A-5
  - defining a logical name for, 5-2
  - definition, 1-2
  - fields, 1-2
  - history records, 1-3, 5-3
  - location, 5-2
  - merging, A-33
  - versions, A-18
- License PAKs and LMF, 6-3
- Licenses
  - activity, 3-3
  - displaying active, A-41
  - per core, 3-3
- License types
  - codes for, 2-2
  - per core license, 3-3
  - RESERVE\_UNITS option, 2-9
  - User, 2-10
- License Unit Requirement Tables
  - See LURT
- License units, 2-1, 2-7
  - allocating, 2-4
  - combination, 2-6, 2-9, 2-11
  - CONSTANT value, 2-2
  - creating a pool of, 2-11
  - modifying, 5-21
  - providing enough, 2-5, 2-8
  - providing more, 2-5, 2-9
  - sharing of activity, 2-9
- Licensing
  - preparation, 5-1 to 5-7
- Licensing requirements
  - clustering in Galaxy system, 6-2
  - clustering outside a Galaxy system, 6-2
  - Galaxy system, 6-1

- Licensing requirements (cont'd)
  - layered products, 6-2
  - OpenVMS operating system, 6-1
- LIST command, A-16 to A-21
  - active status, B-5
  - difference from SHOW LICENSE command, A-16, B-5
  - displaying a license with, 5-14
  - example, 5-14
- List size restrictions, reservation list, A-29
- LMF\$DISPLAY\_OPCOM\_MESSAGE, 5-23
- LMF\$LICENSE database, 4-2
- LOAD command, A-22 to A-24
  - in an OpenVMS Cluster environment, A-24
- LURT (License Unit Requirement Table)
  - file name, 2-1
  - setting up, A-38

## M

---

- MODIFY command, A-25 to A-31
  - controlling reservation list size, 5-22
  - using the /ADD qualifier, 5-22
  - using the /AUTHORIZATION qualifier, 5-20
  - using the /EXCLUDE qualifier, 5-19
  - using the /INCLUDE qualifier, 5-19, 5-20
  - using the /RESERVE qualifier, 5-21
  - using the /UNITS qualifier, B-7
  - using the /VIRTUAL qualifier, A-28
- MOD\_UNITS option, 5-21
  - example, B-7
- MOVE command, A-32 to A-33

## N

---

- NO\_SHARE option, 5-2
- /NO\_SHARE qualifier
  - controlling license combination, 2-11

## O

---

- OpenVMS Cluster environments
  - controlling node access, 5-19
  - managing licenses in, 5-19
  - NO\_SHARE option, 5-2
  - PAK restrictions, 2-5
  - providing more availability in, 2-7
  - registering licenses for, 5-6
  - sharing Activity License units in, 2-9
- OpenVMS Clusters
  - NO\_SHARE, 5-20
  - OpenVMS guest licensing, 4-1
- OpenVMS guests
  - in an OpenVMS Cluster, 4-1
  - licensing, 4-1
  - SHOW LICENSE/USAGE command, 4-2
  - /VIRTUAL qualifier, 4-2

## P

---

/PAGE=SAVE qualifier navigation keys, A-42

PAKs (Product Authorization Keys), 5-1

getting, 5-3

issuers, 5-4

location, 6-1

registering, B-1

replica, A-13

transfer methods, 5-4

per core license, 3-3

Performance ratings, 2-1

Personal Use License, 2-9, A-27

Producer

difference between PAK issuer and, 5-4

Product

installing, 5-1, 5-6

matching release and license, 5-1

Product Authorization Keys

See PAKs

## R

---

REGISTER command, 5-1, A-34 to A-37

Reservation lists, A-27

access denied to users on, 5-22

matching user number to combined units, 5-22

replacing many with one, 5-23

restrictions, A-29

used with LICENSE MODIFY command, 5-21

RESERVE\_UNITS option, 2-9, A-27

## S

---

SCS node names, used with LICENSE MODIFY command, 5-2, 5-19

Selection weight

controlling license loading order, 5-23

/SELECTION\_WEIGHT, 5-23

SHOW LICENSE command

difference from LICENSE LIST command, A-16, B-5

examples, B-5

SMM (System Marketing Model), 2-3

START command, A-38

Syntax

of LICENSE commands

descriptions, A-1 to B-1

System disks

using multiple, 5-2

System Management utility (SYSMAN)

licenses

loading, 5-16, A-24

System memory, transferring from the License

Database, 5-16

## U

---

Unlimited units, 2-3, 5-18

UNLOAD command, A-39 to A-40

restricting access with, 5-18

User Licenses, 2-10

## V

---

Virtual licensing, 4-1, 4-2

VIRTUAL option, 4-1, 4-2

/VIRTUAL qualifier, 4-2

VMSLICENSE.COM, 5-1

batch processing with, 5-11

canceling default data with, 5-10

creating data files for, 5-12

default value rules used in data files, 5-13

examples, 5-7, B-2

parameters used in data files, 5-12

registering a license with, 5-7, B-2

using data files with, 5-11

## Z

---

Zero units, 2-3, 5-18

