Educational Services

Guidelines for License Management

EY-D633E-SG-0001

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227–7013.

The following are trademarks of Digital Equipment Corporation:

ALL-IN-1
DEC
DECwindows
LAT

ULTRIX
VAX
VAX BASIC
VAX COBOL

VAX FORTRAN
VAX-11/780
VAXcluster
VMS
digital™

™ Display PostScript® is a registered trademark of Adobe Systems, Incorporated.

This document was prepared and published by Educational Services Development and Publishing, Digital Equipment Corporation.

Contents

Figures

Tables

About This Course

Course Description

This course is based on *A Business Guide to Digital Equipment Corporation's Software Business Technologies—The License Management Facility*.

Course Format

In general, this course is designed with background information on the **LEFT** page and course content on the **RIGHT** page. There is extra space for note taking.

Intended Audience

This training is designed to provide instruction and supporting by case studies on license management. This course is for you if you are a Business Manager (with a strong software development background) at an application development company.

Purpose and Design of the Course

Each case study will:

1. Point out a problem.
2. Define the components of LMF that can help solve it.
3. Provide an opportunity for you to propose possible solutions.

Goals

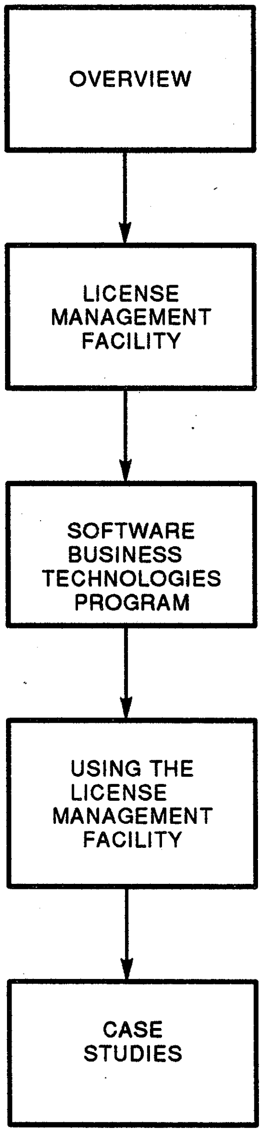
When you complete this course, you should be able to:

- Make decisions regarding the use of the License Management Facility (LMF) as the technology for license management.
- Manage the introduction of Product Authorization Keys (PAKs) and license management to customers.
- Set up a process to handle generation of PAKs.
- Handle potential problems with large quantity of PAK data that needs to be handled by customer.
- Modify the delivery cycle to include the shipment of a PAK upon a customer's purchase of license.
- Use DDSLA/LMF to provide loans, demos, and special offers to customers on both VMS and ULTRIX operating systems.
- Deliver PAKs by paper, phone, media, or other alternatives.

Non-Goals

This course is **not** designed to provide instruction on the Digital pricing schemes currently available or how to select a license and pricing policy.

Course Map


OVERVIEW
LICENSE MANAGEMENT FACILITY
SOFTWARE BUSINESS TECHNOLOGIES PROGRAM
USING THE LICENSE MANAGEMENT FACILITY
CASE STUDIES

Overview

1.1 Objectives and Resources

Objectives

When you complete this section, you should be able to:

- Define licensing
- Define a software license
- Define license styles

Resources include:

- *DEC VMS/LMF System Services Reference Manual (AA-NL45A-TE)*
- *DEC LMF PAK Generator for VMS Installation Guide (AA-NL42A-TE)*
- *DEC LMF PAK Generator for VMS Routine Reference Manual (AA-NL44A-TE)*
- *DEC LMF PAK Generator for VMS Application User's Guide (AA-NZ26A-TE)*
- *VMS License Management Utility Manual (AA-LA33A-TE)*
- *A Practical Guide to Managing Software Licenses (EJ-31899-76)*
- *The License Management Facility (AA-NZ27A-TE)*

Background Information

This section will cover these topics.

1.2 Overview

- Digital Distributed Software Licensing Architecture (DDSLA)
- Enforcement
- License management
- Types of licensing

Background Information

This course contains information about a new software tool offered by Digital Equipment Corporation—the License Management Facility (LMF). If you produce software, either for internal or external use, you may want to use this tool.

If you decide to use this tool, you may find that it opens new ways of licensing, pricing, and distributing software.

This course explains the LMF and illustrates how you might use it. It shows how the LMF may raise some business opportunities for you, and talks about the types of business decisions you have to make prior to using the LMF.

On the back of a PAK, terms and conditions for that PAK can be stated, for example:

SOFTWARE LICENSE PRODUCT AUTHORIZATION KEY

Use of the Software License Product Authorization Key, (PAK), the alphanumeric sequence appearing on the front of this document, is authorized only pursuant to a valid software license granted by Digital Equipment Corporation (DEC) to the individual or entity identified by the PAK ("the licensee").

This software does not constitute a software license.

The Software PAK is an integral part of the licensed software product identified on the front of this document ("the software") and is provided subject to the license terms applicable to the software.

The Software PAK is proprietary to and embodies the confidential technology of Digital. Possession, use, duplication or dissemination of the Software PAK is improper and unlawful unless pursuant to a valid software license from Digital.

Licensee shall maintain the Software PAK as confidential, and not transfer, disclose or otherwise make it available to any third party.

Any copies of the software PAK must be limited to internal use by licensee only.

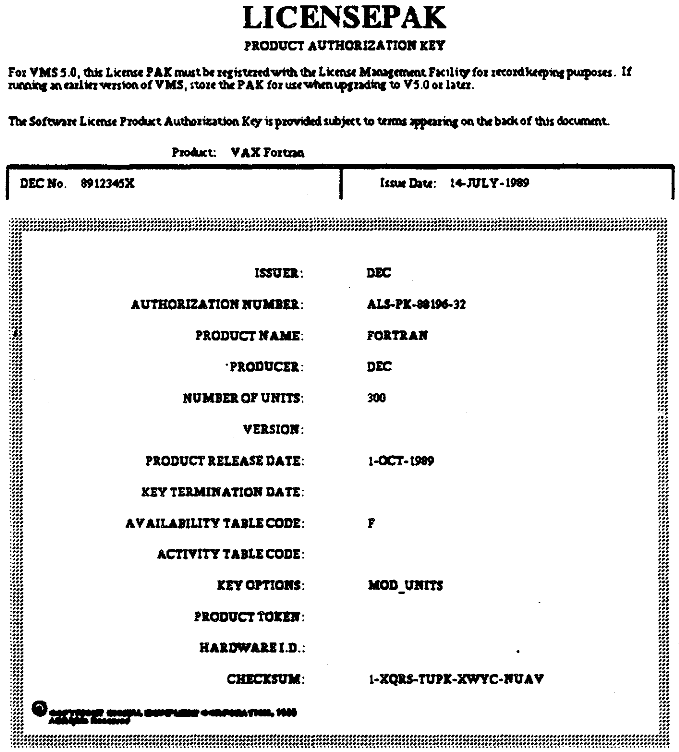
The Software PAK must be used to install the software on the Licensed Processor.

1.3 Digital Distributed Software Licensing Architecture

This course examines ways of incorporating the Digital Distributed Software Licensing Architecture (DDSLA)/License Management Facility (LMF) into your business. As a Digital customer, you may know that a Product Authorization Key (PAK), the list of license information that the user needs to enter into the License database, is an important component of LMF.

In this course, you'll learn about how your company's business policies are related to the values supplied on a PAK.

Figure 1–1 is a sample Product Authorization Key.



LICENSEPAK

PRODUCT AUTHORIZATION KEY

For VMS 5.0, this License PAK must be registered with the License Management Facility for record keeping purposes. If running an earlier version of VMS, store the PAK for use when upgrading to V5.0 or later.

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

Product: VAX Fortran

DEC No. 8912345X	Issue Date: 14-JULY-1989

ISSUER:	DEC
AUTHORIZATION NUMBER:	ALS-PK-88196-32
PRODUCT NAME:	FORTRAN
·PRODUCER:	DEC
NUMBER OF UNITS:	300
VERSION:	
PRODUCT RELEASE DATE:	1-OCT-1989
KEY TERMINATION DATE:	
AVAILABILITY TABLE CODE:	F
ACTIVITY TABLE CODE:	
KEY OPTIONS:	MOD_UNITS
PRODUCT TOKEN:	
HARDWARE I.D.:	
CHECKSUM:	1-XQRS-TUPK-XWYC-NUAV

© COPYRIGHT DIGITAL EQUIPMENT CORPORATION, [illegible]
All Rights Reserved

Figure 1–1 A Sample Product Authorization Key

Background Information

The LMF cannot ensure compliance with the license terms and conditions, but it helps to prevent accidental non-compliance. Non-compliance with the license terms and conditions requires an overt act, such as copying a PAK, knowingly registering it on more than one system, tampering with a software product, or patching the system.

LMF is like the sawhorse and bright yellow tape that police often use to cordon off the scene of an accident. These things do not prevent anyone from entering the area, but if they wish to do so, they must intentionally bypass these devices.

The philosophy behind LMF is that most users are honest and will comply with the license terms and conditions if they can. The LMF makes it easier for users to comply.

1.4 Enforcement

- The LMF is absolutely not an enforcement tool.
- It is a license management and tracking tool.

Background Information

The LMF is part of the SBT program and is viewed, within Digital at least, as a prerequisite to the consolidated distribution of software on CDROM. But the LMF has inherent benefits as well.

A license to use software is an intangible thing—it is an agreement between two parties. Many customers have asked for something they can touch, feel, and file. (In some parts of the world this is a legal requirement.) A PAK certificate is just that—a physical representation of a license. And the LMF, using the License database, is a built-in way to manage that information. It provides a central repository for software license information, and it gives system managers a way to examine this information online. gives system managers a way to examine this information on line,

Moreover, even without the LMF, PAK certificates are useful. In most cases there is nothing beyond the vendor's invoice that clearly establishes that someone has a right to be using a software product. A PAK certificate is tangible proof of licensing, something that is easy to file and reference.

Internal software is software that you produce and use within your own company. Unless you are affiliated with the MIS department, there may be more of this type of software being used within your company than you are aware of.

Internal software has a habit of acquiring a "life of its own." It is very easy for the group that developed the software to quickly lose track of who their users are. The users don't know who to contact for support, or enhancement. If access to the software were managed by using the LMF, this opens up new possibilities for the producers of the software. The producers could charge a fee for a PAK (an internal transfer cost) that might result in popular applications funding their own support and enhancements.

1.5 License Management

- A license is an agreement between two parties.
- A PAK is a document that you may use as a physical representation of a license. The key provides a way to manage access to demos, temporary software uses, promotions, and other business opportunities.
- LMF can be used to manage internal software product licenses. You may already charge a transfer fee for maintenance and support of internal products.

Background Information

We'll be discussing the use of LMF-supported license types. Are there other licensing styles that your business uses?

- Personal use
- Network
- Metering
- Site

1.6 Types of Licensing

The LMF can support three license formulas:

- An **availability** license is one that makes software available to anyone on the system.
- An **activity** license is based on the activity of the software such as the number of users concurrently accessing it.
- A **compound** license is one that charges license units both by the system and by the user.

Background Information

You should review these topics.

1.7 Summary

- Digital Distributed Software Licensing Architecture (DDSLA)
- Enforcement
- License management
- Types of licensing

License Management Facility

2.1 Objectives and Resources

Objectives

When you complete this module, you should be able to:

- Define the components of the License Management Facility.
  1. The License Unit Rating Table (LURT)
  2. The License Database
  3. The Product Authorization Key (PAK)
  4. A utility to maintain the License Database
  5. Two system service calls
  6. DEC LMF PAK Generator for VMS Software

Resources include:

- *DEC VMS/LMF System Services Reference Manual (AA-NL45A-TE)*
- *DEC LMF PAK Generator for VMS Installation Guide (AA-NL42A-TE)*
- *DEC LMF PAK Generator for VMS Routine Reference Manual (AA-NL44A-TE)*
- *DEC LMF PAK Generator for VMS Application User's Guide (AA-NZ26A-TE)*
- *VMS License Management Utility Manual (AA-LA33A-TE)*
- *A Practical Guide to Managing Software Licenses (EJ-31899-76)*
- *The License Management Facility (AA-NZ27A-TE)*

Background Information

This section will cover these topics.

2.2 License Management Facility

- Legal protection of software
- Digital's experience with licensing
- LMF components
  - License Unit Rating Table (LURT)
  - Product Authorization Key (PAK)
  - LICENSE Utility and Database (LDB)
  - System Services
  - PAK Generator Software

Background Information

Software development is an expensive, labor-intensive business. After spending the money to develop a software product, you are interested in protecting your investment.

Copyrights, trade secrets, patents, and trademarks are legal mechanisms for declaring ownership of property in general, and intellectual property such as software products in particular. The importance of declaring ownership is that you as the owner then have a number of rights with respect to that property, including the right to set the terms and conditions for its use.

Contracts are the legal mechanisms for controlling the use of the software, and a license is one type of contract. A license grants the right to use your software.

2.3 Legal Protection of Software

Your software has two levels of legal protection:

- You declare ownership of the software product
- You control the use of that software product

Background Information

Within this broad definition of a license, there are many ways to actually design and implement one. In other words, there are many ways to write the terms and conditions. When you write your terms and conditions, you must keep your users in mind, and write something they can comply with. It would be difficult, for example, for even a well-intentioned customer to comply with a license that grants the right to use software between the hours of 9 a.m. and 5 p.m. on alternate Wednesdays.

Digital Equipment Corporation faced one such situation with licensing, and it had to do with the VAXcluster system environment.

2.4 Digital's Experience with Licensing

- According to the terms and conditions of the license agreement for software running in a VAXcluster system environment, each system had to be licensed to run the software.
- Unless extraordinary system management measures were taken, once the software was installed on a single system in the cluster, it was available to all users on the cluster.
- When customers complied with the terms and conditions and licensed each system in the cluster, they paid more to run the software in a cluster than on a single machine with similar power.
- A motivating factor in the development of the License Management Facility was the desire to correct this situation.

Background Information

One of the first steps taken was to develop a licensing architecture. This became known as the Digital Distributed Software Licensing Architecture (DDSLA).

At the heart of the architecture is the concept of **license units**. This is a new way of thinking about licensing, and is intended to help break the bond between a license and a particular piece of hardware.

To make this discussion more interesting, let's pretend that you are the president of Gregg Associates, Inc. (GAI), a small software company. The two products we'll be using in our examples are your XYZ compiler, and your Accounts Receivable application.

Currently you license your XYZ compiler to run on a specific system. The DDSLA allows you instead to license a number of units of XYZ, and then make rules about how the user can "spend" those units. This idea has opened the way to new methods of licensing software, and has led to the creation of several new tools and structures. Rather than be abstract, let's look at how the DDSLA was implemented in the License Management Facility (LMF).

2.5 LMF Components

There are several parts to the LMF:

Since the introduction of VMS Version 5.0, Digital has provided:

1. The License Unit Rating Table (LURT)

   The VMS file is SYS$COMMON:[SYSEXE]LMF$LURT.DAT

2. The License Database (LDB)

   The VMS file is SYS$COMMON:[SYSEXE]LMF$LICENSE.LDB

3. The LICENSE utility to maintain the License database

   To enter PAK information into the License database, a VMS system manager may use either tool:

   - SYS$UPDATE:VMSLICENSE.COM command procedure
   - LICENSE REGISTER command

   To allow users to access a licensed product, a VMS system manager must activate each registered license (transferring or loading data from the on-disk License database into system memory) by using the LICENSE LOAD command.

4. The Product Authorization Key (PAK)

   A PAK contains licensing information that a customer must register in the License database.

Now, using the following components, you can create LMF-compliant products:

1. Two system service calls (SYS$GRANT_LICENSE and SYS$RELEASE_LICENSE)
2. PAK Generator Software

Background Information

Who's responsible for providing information to the customer?

A vendor provides an LMF-compliant product and needed PAK to the customer. The customer uses LMF components (part of the VMS operating system) to manage license data. The system manager uses the LICENSE utility to:

- Register the PAK information into the License database
- Load each license

2.5.1 Responsibilities

From a customer's point of view, Digital and a vendor share responsibility in providing PAK information and tools for gaining access to software products.

Figure 2–1 illustrates these responsibilities.

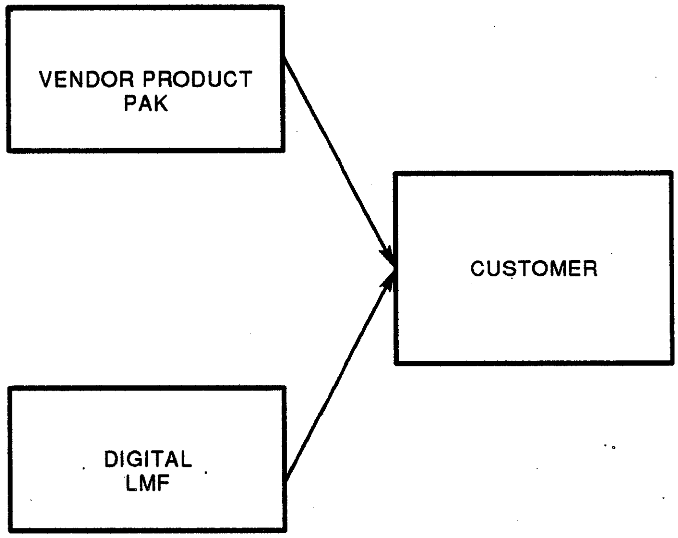


Figure 2–1 Responsibilities

Background Information

The LMF was first implemented as part of the VMS operating system, and contains special processing for VAXcluster systems. In fact, a "system" to the VMS/LMF is either a stand-alone system, or a VAXcluster system.

A method was created to rate all the available systems according to how many units it would cost to run various types of software on them. This notion led to the creation of a License Unit Rating Table (LURT). The table is a simple one, made up of rows and columns. There is one row for each System Marketing Model (which is usually the nameplate on the front of the system). The columns are labeled A, B, C, and so on. They represent different types of software. Several are reserved for the operating system. One is for system integrated products (such as Volume Shadowing). One is for layered products.

Here is an extract from the LURT showing some of the entries from the layered products column, which is column F. (See your most recent operating system release notes for a complete and up-to-date LURT.)

2.5.2 License Unit Rating Table (LURT)

Table 2–1 shows that we need 300 license units of a software product (such as your XYZ compiler) to run it on a VAX 6210, and 1200 license units to run it on a VAX 6240 **or** a VAX 8800.

Table 2–1 License Unit Rating Table

System Marketing Model	F
VAX 11/780	100
VAX 6210	300
VAX 6220	600
VAX 6230	900
VAX 6240	1200
VAX 8800	1200
VAX 8840	2400

Background Information

Customers get the information to enter into the License database from a Product Authorization Key, or PAK, that you send to them. A PAK is simply a list of license information that the user needs to enter into the License database.

A PAK consists of 13 pieces of information, plus a verification mechanism called a **checksum** that makes sure the information is properly entered into the database.

Remember that a PAK is simply a list of information about a software license. It doesn't need to be printed out; it could be distributed as an electronic file that the License Utility could process and enter into the License Database. When it does get printed out, the piece of paper is referred to as a PAK document.

Don't confuse a PAK Certificate with a license—they're not the same thing. A license is intangible; an agreement between two parties. A PAK may have some encoding of some of the license agreement's terms and conditions, but it is not itself a license. A PAK may be other things as well – there are several different types of PAKs, as we shall see.

If it is printed out, a PAK may serve as a **proof** of license.

You should examine each of these fields to determine the business decisions you need to make involving each entry.

2.5.3 Product Authorization Keys (PAKs)

Thirteen data elements can appear on a PAK that are used together to calculate the **checksum** (an encoded number calculated from the other information supplied on a PAK). The elements are:

- ISSUER
- AUTHORIZATION NUMBER
- PRODUCT NAME
- PRODUCER
- NUMBER OF UNITS
- VERSION
- PRODUCT RELEASE DATE
- KEY TERMINATION DATE
- AVAILABILITY TABLE CODE
- ACTIVITY TABLE CODE
- KEY OPTIONS
- PRODUCT TOKEN
- HARDWARE ID

Background Information

Now, let's look at each element in detail and answer the following questions:

- What is the data that can appear in the field?
- How is it used?

NOTE
The PAK Generator may limit the number of characters for each field. Refer to the *DEC LMF PAK Generator for VMS Routines Reference Manual* for more information.

2.5.3.1 Issuer

The Issuer field contains the name of the legal entity that issues the license.

The rules for creating your Issuer name are:

- The name cannot exceed 31 characters.
- The name should not be case sensitive.
- No blanks can be used. Use the dash (-) character instead of a blank.
- Besides letters, numbers, and the dash character, you may only use the period (.) character in your name.

The Producer name and the Issuer name are usually the same. The names may differ in the case of resellers, if one party is authorized to license another party's software product.

Background Information

Description of PAK information continues.

2.5.3.2 Authorization Number

The Authorization Number field is a text field. The combination of Issuer and Authorization Number is intended to uniquely identify a particular software license.

The rules for creating your Authorization Number are:

- The name cannot exceed 31 characters.
- The name should not be case sensitive.
- No blanks can be used. Use the dash (-) character instead of a blank.
- Besides letters, numbers, and the dash character, you may only use the period (.) character in your name.

Digital's Authorization Numbers encode certain information about the PAK type, the location that generated the PAK, and the date on which it was generated. You may follow a similar scheme, or another of your own creation.

For example, you may have the number AB-DK-89287 for a demo PAK generated at facility AB on the fourteenth day of October (Day 287 of 1989).

The Authorization Number might include a purchase order number as well.

Background Information

Description of PAK information continues.

2.5.3.3 Product Name

The Product Name is a (possibly) abbreviated form of your product's name because the LMF limits you to 24 characters. The combination of Product Name and Producer Name will uniquely identify a software product.

Thus, within the scope of the LMF, naming your product FORTRAN is not a problem although Digital has a product called VAX FORTRAN. Because the two Producer Names are different, the two products can be distinguished.

Your goal is to create a straight-forward, easy to remember Product Name.

The rules for creating your product name are:

- The name cannot exceed 24 characters.
- The name should not be case sensitive.
- No blanks can be used. Use the dash (-) character instead of a blank.
- Besides letters, numbers, and the dash character, you may only use the period (.) character in your name.

The spelling of the Product Name must be identical within the call and on the PAK.

Background Information

Description of PAK information continues.

2.5.3.4 Producer

The Producer name field obtains the name of the legal owner of the software, your company. It is an agreed-upon representation of your company's name, and it is verified by the PAK Generator software.

You have to register your Producer Name with Digital before receiving the PAK Generator software.

Also, your Producer Name must be unique across all other companies who are using the PAK Generator software. If you are a public company, you might consider using your stock exchange name for your Producer Name.

The rules for creating your Producer Name are:

- The name cannot exceed 24 characters.
- The name should not be case sensitive.
- No blanks can be used. Use the dash (-) character instead of a blank.
- Besides letters, numbers, and the dash, you may only use the period (.) character in your name.

The spelling of the Producer Name within the system service call must exactly match the spelling in the PAK.

Background Information

Description of PAK continues.

In the case of a product like a message router (the processor it's installed on is unrelated to its pricing or licensing), you might set static number of units to be associated with the product.

2.5.3.5 Number of Units

A unit is an arbitrary measure. The Number of Units field contains the maximum value to grant for this PAK.

It must be a value in the range of 0 through 2 billion. A value of 0 here has a special meaning: infinity. (A PAK with 0 units specified is an unlimited-use PAK.)

Background Information

Description of PAK information continues.

The Producer Name and Product Name will generally not change from release to release of the product.

However, in each release of the software, the engineer must encode the version number, and the date on which the software is to be released. Both of these pieces of information should be included, regardless of how the product is currently licensed. In this way, license policy is separated from software engineering. It is then possible to change one without affecting the other.

The PAK is supposed to embody the license policy, and the contents of these fields can reflect that.

If you license strictly by version number, include a Version Number on the PAK. At the time the product executes, the LMF will compare the contents of this field on the PAK to the information passed in the system service. If there is a Version Number specified on the PAK, and if the version in the call is less than or equal to the version on the PAK, it is authorized.

2.5.3.6 Version

The Version field is optional. The field represents the highest version number of your software product that the PAK will authorize.

The Version Number must be in the form nn.nn (where n is a number between 0 and 9).

Note that the "." is not a decimal point. If 4.10 is in the field, not only is 4.10 authorized, but also Version 4.9, Version 4.8, Version 4.7, and all earlier versions.

Background Information

Description of PAK information continues.

If you license by **date**, include a Product release Date on the PAK. At the time the product executes, the LMF will compare the contents of this field on the PAK to the information passed in the system service. If there is a Product Release Date specified on the PAK, and if the date in the call is earlier than or equal to the date on the PAK, it is authorized.

You cannot have both a Version and a Product Release Date on a PAK. But you can leave them both blank. What happens if you do? Barring an entry in the Key Termination Date field, your software product is authorized to run forever, for any version. If a user acquires the most recent version of your software product, a PAK with blank Version and Product Release Date fields will permit the use of the product.

2.5.3.7 Product Release Date

The Product Release Date is optional. If it is used, it represents the latest release date of your software product that the PAK will authorize.

If 1-JAN-1991 appears in this field, all versions of your software product that were released on or before that date are authorized. This is useful if your licensing policy includes an update policy whereby the user is entitled to all releases of your product that occur within a given length of time, such as a year.

This date must be in the form dd-mmm-yyyy.

You cannot have both a Version and a Product Release Date on a PAK since each reflects a separate licensing practice.

Background Information

Description of PAK information continues.

If there is a Version Number specified as well, the user may use any version of the software up to and including the Version Number specified, until the termination date.

If there is a Product Release Date specified, the user may use any version of the software released on or before the Product Release Date specified, until the termination date.

And if there is no Version Number or Product Release Date specified, the user may use any version of your software product until the termination date.

2.5.3.8 Key Termination Date

The Key Termination Date field contains a date on which authorization to use the software product is terminated. It can be used for rental agreements, trials, and field test versions of software.

As with Product Release Date, the Termination Date must be in the form dd-mmm-yyyy.

If you choose to use this field, you might want to remind a customer by mail that the PAK is due to expire.

Background Information

Description of PAK information continues.

The sliding scale makes use of the License Units Rating Table (LURT) included with the operating system. See the current operating system release notes for an up-to-date listing of the LURT, and see the PAK Generator release notes for a listing of what columns in the table are available to third parties.

After the charge has been determined, the LMF will compare the charge against available units. If there are enough units available, the LMF will authorize the execution of the product. If not, the LMF will indicate that authorization has not been granted.

For capacity-based licenses for layered products, a PAK would have the letter F in the Availability Table Code field, nothing in the Activity Table Code field, and the number of units required in the Number of Units field (for example, 300).

2.5.3.9 Availability Table Code

The Availability Table Code field is one of two fields (Activity Table Code is the other) used to specify how many units are needed to execute the software product.

An availability license grants use of the product to any number of users. It is available to all.

Two methods determine how many units to require for executing software:

- **A sliding scale based on the type of hardware**

  If there is a column entry in this field (such as F0, the charge for the product is determined at execution time, based on the type of hardware.

- **A fixed charge**

  To charge a flat rate for use of the software product, enter the number of units to charge in the form CONSTANT=N where N is a number in the range of 0 through 2 billion. If you specify CONSTANT=0, there is no unit-charge for your product; it effectively grants unlimited use in any hardware configuration.

Background Information

Description of PAK information continues.

2.5.3.10 Activity Table Code

The Activity Table Code field is used to determine how many license units to reserve (charge) for the use of the product. The difference is that this charge is per user. Use the Activity Table Code to grant use of your product to a specific number of executions of your product.

You can have your charge determined by the hardware environment, by using the LURT table.

Alternately, you can specify a flat-charge per user by entering the number of units. Use the form CONSTANT=N (where N is a number in the range of 1 through 2 billion). CONSTANT=0 is not allowed because it is meaningless. To specify that there is no unit-charge for the product, enter CONSTANT=0 in the Availability Table field.

Background Information

Description of PAK information continues.

For ULTRIX systems, p_family is also an option (but not currently supported by the PAK Generator).

2.5.3.11 Key Options

As its name implies, this field is optional. It allows you to specify some rules applied by the LMF at the operating system level. Your product does not have to include any extra code. The current options are:

1. **MOD_UNITS**

   If MOD_UNITS is specified as an option, the user will be able to use the LICENSE MODIFY command to change the number of units on a PAK.

   This option enables a requirement that is often in a license's Terms and Conditions: granting the user the ability to run the software on a backup machine in case of an emergency. If necessary, the system manager uses the LICENSE MODIFY command to increase the units to permit the software product to run on a backup machine, which may exact a larger charge than the one licensed. The expectation is that when the primary machine is running again, the backup PAK must be disabled.

2. **NO_SHARE**

   If NO_SHARE is specified as an option, the user is restricted to using the software on a specific single node. If on a cluster, you can specify the node by using the INCLUDE list, which is discussed in *The VMS License Management Utility Manual.*

   NOTE
   It is possible to have both MOD_UNITS and NO_SHARE specified on a PAK.

Background Information

Description of PAK information continues.

You might use the Product Token Field to contain a country name, a customer name and address, or specific information your product uses.

For example, if your software is a program used to promote political candidates, you might use the Product Token field to contain the person's name. Then, when your program is invoked, it appears customized with the candidate's name throughout.

You could also use the Product Token field to contain the Key Termination Date. Your product could display that date and remind customers that the PAK will expire.

2.5.3.12 Product Token

This field is one of two fields (along with Hardware ID) that allows your software product to modify its behavior.

The Product Token field can contain a string of up to 31 characters, that can be returned to your software product when it issues its service call. What your product then does with that string is limited only by the ingenuity of your engineering staff.

For example, the Product Token field might contain a decryption key that would be used by your product to decrypt portions of itself. Another example is the PAK Generator, which uses the contents of the Token field (your company name on the PAK that authorizes its use) to verify the Producer and Issuer Names on every PAK it generates.

Background Information

Description of PAK information continues.

Since you can use this field as a second token, you might consider passing the Key Termination Date in this field. That way, your product can examine the contents of this field and output a message to your user when the Key Termination Date is near.

2.5.3.13 Hardware ID

Another optional field, the Hardware ID field is intended to contain the serial number of the CPU authorized to run the software. Since the DDSLA does not support this feature, individual software products must implement it if so desired. The system service call returns the contents of this field to your product, which then must compare it to the serial number of the local CPU.

The Hardware ID field can contain a string of up to 31 characters that can be returned to your software product when it issues its service call.

You must use a capital X as the first character in the Hardware ID field, and then strip it from your program. This character may have special meaning to the LMF should CPU serial-number checking ever be supported. But, for the time being, this field is functionally equivalent to the Token field.

Background Information

Using PAK information.

For example, a telephone switching product might charge a flat fee for access and then an additional fee based on the number of times the product is accessed.

2.5.3.14 Compound Licensing

In some cases, you may want an availability unit charge plus a per-user charge. To accomplish this, enter information into both Availability and Activity Table Code fields. The LMF will make a one-time charge of what is entered into the Availability Table Code field, and then charge per-user based on what is in the Activity Table Code field.

> NOTE
> You must enter a value into either the Availability or Activity Table Code fields; they may not both be blank. However, as we have seen with Compound Licensing, they may both contain values.

Background Information

The license terms and conditions define what right is actually being granted in a license agreement. When you use the LMF you encode some of these terms and conditions in a simple way such that they can be stored as records in a database. We call these records **license records**. The database that holds these license records is part of the operating system.

Under the architecture, the units of like license records are combined before they are allocated, or "spent". So if you had two separate license records for XYZ, one with 500 units and one with 400 units, the LMF would combine them into 900 units which would then be allocated as necessary. After combining them, those license units could then be shared among multiple systems in a cluster.

A 900-unit license could be split among nine VAX-11/780 systems, or between a VAX 6210 system and a VAX 6220 system, or they could be allocated to a VAX 6230 system alone.

Because license units can be combined and shared, the License Architecture effectively disconnects licensing from a particular system.

The LICENSE utility to manage the License database is also part of the operating system, and is documented in the *VMS License Management Utility Manual*, which is part of the VMS documentation set.

2.5.4 License Utility and Database (LDB)

One element of a license record is the number of units. License units have some interesting properties:

- **License units can be combined**

  For license combination:

  - If Version number is specified (must be part of both PAKs), the fields must match exactly.
  - If Product release Date is specified (must be part of both PAKs), the earliest of the dates is used.
  - If a Key Termination Date is specified, the earliest date is used.

- **License units can be shared**

Background Information

There is one more missing piece of the puzzle. We've discussed the LURT, a mechanism for determining how much it "costs" to run a product on a system. And we've discussed PAKs and the License database. But how does your XYZ compiler know about any of this?

2.5.5 System Services

To use LMF, you must make a small change to the product itself to include one or two system service calls to the LMF.

- One call checks the authorization of the product.
- Another call releases any units used.

Background Information

You should review this example.

The LMF is passive. A software product must explicitly ask to use it. So, you insert an LMF call into your XYZ compiler.

When a customer begins a compilation, the LMF returns an indication as to whether or not the compiler is authorizes to run. Depending on what you decide, the compiler then may:

- Execute
- Refuse use
- Issue a message
- Log a message

Assuming the customer has purchased enough license units to run their environment, he is never aware of the LMF again.

Besides inserting the LMF call, you must produce a PAK that you issue to your customer when they purchase a license for your XYZ compiler.

If you want to restrict the use of your XYZ compiler to specific nodes within a VAXcluster system, you can include that in your license agreement. A Key Option on the PAK (NO_SHARE) requires that the system manager specify the nodes by name using the LICENSE utility.

You can also permit the system manager to temporarily run the compiler on a backup machine which may require more license units than they have. Let's assume that they have ordered 100 units of your compiler. The system they normally use to run your product needs maintenance and they want to temporarily run your compiler on a backup machine which requires more units. If you want to allow them to do this, you can specify a Key Option on their PAK (MOD_UNITS) which will let the system manager temporarily increase the number of units they have when registering the PAK on the backup system. A history record of this transaction is kept in the License database.

2.5.5.1 An Example

The complete picture is this:

- You modify your XYZ compiler to include a call to the LMF.
- When you sell a license to an end-user you have a new deliverable: a PAK. (You have also modified your documentation to tell your user what to do with the PAK. Recognize that the user may be reluctant to change practices, especially if you've already changed several times recently!)
- You create a PAK (possibly using the PAK Generator).
- The user receives your compiler (or already has it as part of a consolidated distribution).
- The user receives your PAK.
- The user registers the PAK in the License database on their system.

  The LICENSE utility, described in the *VMS License Management Utility Manual*, can be used to enter commands into the License database, and to do other important system management functions:

  - Restrict the use of the XYZ compiler to specific nodes within a VAXcluster system
  - Temporarily run the XYZ compiler on a backup machine
- The user installs the compiler.

  When the user runs it, the system service calls (SYS$GRANT_LICENSE and SYS$RELEASE_LICENSE) in your compiler will cause the LMF to verify that enough license units exist on the system.

  If the number is sufficient, the compiler executes. If there are not enough units, the compiler will then take a predetermined action. For example, your product could refuse use, issue a message, write a message to an audit file, or anything else you wish.

These are some of the business decisions that you must make before implementation.

Background Information

You should review these topics.

2.6 Summary

- Legal protection of software
- Digital's experience with licensing
- LMF components
  - License Unit Rating Table (LURT)
  - Product Authorization Key (PAK)
  - License Database and Utility (LDB)
  - System Services
  - PAK Generator Software

Software Business Technologies Program

3.1 Objectives and Resources

Objectives

When you complete this section, you should be able to:

- Describe Software Access Management
- Describe Consolidated Distribution of Software
- Describe Online Documentation

Resources include:

- *DEC VMS/LMF System Services Reference Manual*
  (AA-NL45A-TE)
- *DEC LMF PAK Generator for VMS Installation Guide*
  (AA-NL42A-TE)
- *DEC LMF PAK Generator for VMS Routine Reference Manual*
  (AA-NL44A-TE)
- *DEC LMF PAK Generator for VMS Application User's Guide*
  (AA-NZ26A-TE)
- *VMS License Management Utility Manual*
  (AA-LA33A-TE)
- *A Practical Guide to Managing Software Licenses*
  (EJ-31899-76)
- *The License Management Facility*
  (AA-NZ27A-TE)

Background Information

This section will cover these topics.

3.2 Software Business Technologies Program

- Software access management
- Consolidated software distribution
- Online documentation

Background Information

Digital Equipment Corporation has created a program called the Software Business Technologies (SBT) Program, to make all software and documentation materials available at every desk, with demonstrations and browsing available to every user.

The first part that is being extended to Digital's third party software vendors is Software Access Management, using the LMF. The other pieces may be extended to third party software vendors in the future.

3.3 Components of the Software Business Technologies Program

The SBT Program has three parts:

1. Software access management
2. Consolidated distribution of software
3. Online documentation

Background Information

Rather than prevent users from copying the software, make it easy for them to do so—but manage the access to it.

3.3.1 Software Access Management

Digital's approach to Software Access Management is the opposite of the copy-protection schemes of the personal computer software market. It:

- Puts a demonstration of the software in front of thousands of users, and then make a full-function copy easily available
- Makes the full-function copy available, but only for a limited time, with an easy mechanism to switch over to a full license

Background Information

Distributing more than one software product can be complex. If you kit each product on a separate medium, your manufacturing costs soar. If you kit all products on a single medium, people may access your software without first obtaining a license. With more than 200 software products, this was a problem Digital had to face.

The solution: Consolidated Software Distribution. Put as many software products as possible on a single distribution medium, and then manage the access to those products using the LMF.

CDs have another characteristic which can translate to a potentially tremendous savings in manufacturing costs: CDs are stamped, so the reproduction of 1 bit or 1 trillion bits takes the same time and, in volume, costs the same.

By the way, putting multiple software products on a single medium is a boon not only for the software producer. It also makes life much easier for system managers and, ultimately, for end-users, since it is more likely that the software they want is on the consolidated distribution. Plus, you can give the users demonstrations, product marketing information, and ordering information on the same disk.

3.3.2 Consolidated Software Distribution

Digital has selected Compact Disk, Read-Only Memory (CDROM) as the medium of choice for consolidated distribution.

Identical in size and physical format to audio CDs, CDROM offers the following:

- 600 megabytes of data
- Random access
- Good seek times (and getting better)
- An audio industry user base which keeps standards stable
- Technology which is actively improving
- Better security with Read-only access which makes it difficult to tamper with the data

Background Information

In addition to easy access to software at their desks, users want easy access to documentation.

In the case of both Consolidated Distribution and Online Documentation, you need a way to control access to the contents of the CDROM. This is **one** function of the License Management Facility.

3.3.3 Online Documentation

Digital provides Online Documentation, which includes:

- Documentation displayed on a workstation making use of the bit-mapped graphics technology
- CDROM storage containing thousands of pages of text and graphics
- Software designed to maximize the information retrieval capabilities
- Rapid access to documentation by all users
- Automatically updated manuals

Background Information

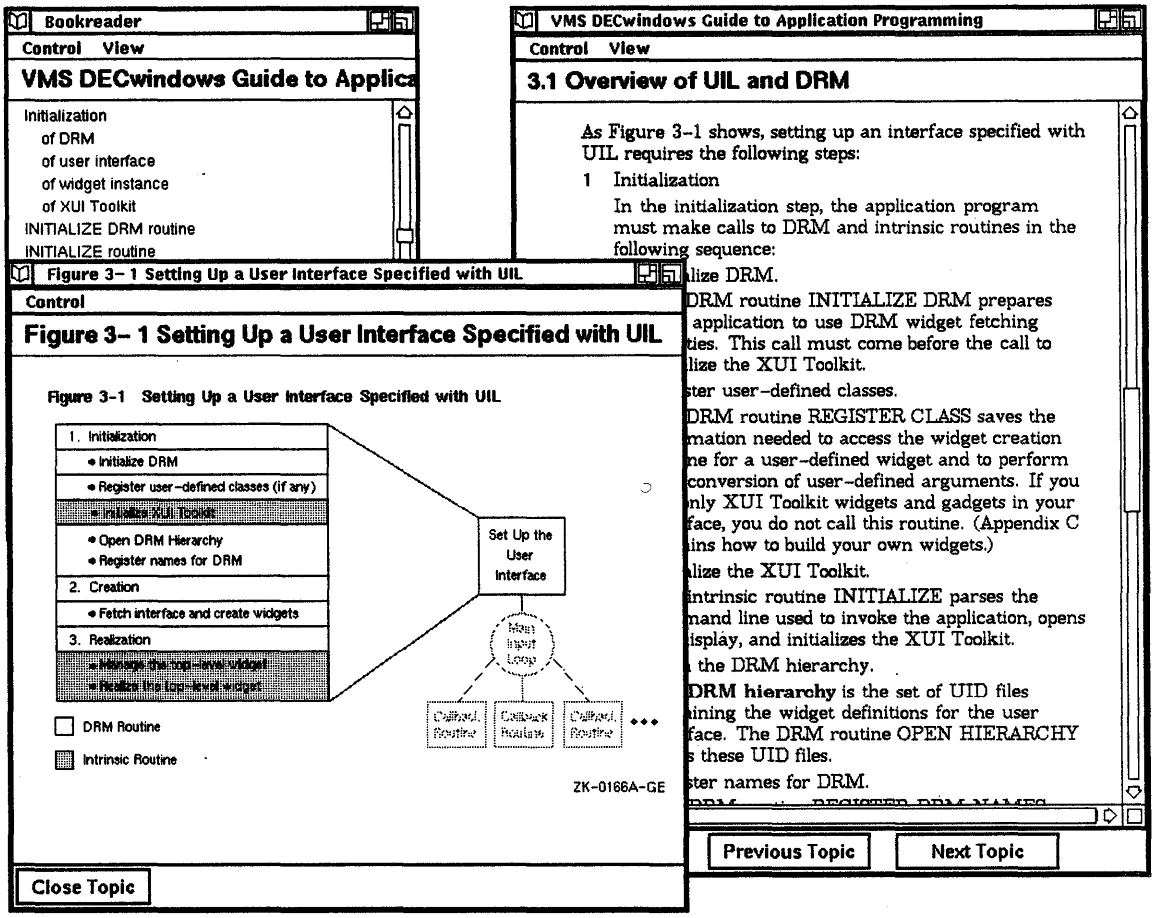
The Bookreader lets you read online documentation on your workstation screen. Bookreader contains two windows: a selection window and a topic window.

The selection window includes a library of bookshelves and books that are available from the Bookreader. When you choose a book from the selection window, Bookreader will open a topic window.

The topic window contains the book that you want to read. The table of contents is displayed in the selection window.

3.3.3.1 Bookreader

Figure 3–1 displays the Bookreader application.



Initialization
of DRM
of user interface
of widget instance
of XUI Toolkit
INITIALIZE DRM routine
INITIALIZE routine

Figure 3– 1 Setting Up a User Interface Specified with UIL

Figure 3-1 Setting Up a User Interface Specified with UIL

1. Initialization
   - Initialize DRM
   - Register user-defined classes (if any)
   - Initialize XUI Toolkit
   - Open DRM Hierarchy
   - Register names for DRM
2. Creation
   - Fetch interface and create widgets
3. Realization
   - Manage the top-level widget
   - Realize the top-level widget

□ DRM Routine

▨ Intrinsic Routine

3.1 Overview of UIL and DRM

As Figure 3-1 shows, setting up an interface specified with UIL requires the following steps:

1 Initialization

In the initialization step, the application program must make calls to DRM and intrinsic routines in the following sequence:

lize DRM.

DRM routine INITIALIZE DRM prepares application to use DRM widget fetching ties. This call must come before the call to lize the XUI Toolkit.

ster user-defined classes.

DRM routine REGISTER CLASS saves the mation needed to access the widget creation ne for a user-defined widget and to perform conversion of user-defined arguments. If you nly XUI Toolkit widgets and gadgets in your face, you do not call this routine. (Appendix C ins how to build your own widgets.)

lize the XUI Toolkit.

intrinsic routine INITIALIZE parses the nand line used to invoke the application, opens isplay, and initializes the XUI Toolkit.

the DRM hierarchy.

DRM hierarchy is the set of UID files ining the widget definitions for the user face. The DRM routine OPEN HIERARCHY s these UID files.

ster names for DRM.

Figure 3–1 Bookreader

Background Information

You should review these topics.

3.4 Summary

- Software access management
- Consolidated software distribution
- Online documentation

Using the License Management Facility

4.1 Objectives and Resources

Objectives

When you complete this section, you should be able to:

- Describe the steps for implementing the License Management Facility calls in a product.

Resources include:

- *DEC VMS/LMF System Services Reference Manual (AA-NL45A-TE)*
- *DEC LMF PAK Generator for VMS Installation Guide (AA-NL42A-TE)*
- *DEC LMF PAK Generator for VMS Routine Reference Manual (AA-NL44A-TE)*
- *DEC LMF PAK Generator for VMS Application User's Guide (AA-NZ26A-TE)*
- *VMS License Management Utility Manual (AA-LA33A-TE)*
- *A Practical Guide to Managing Software Licenses (EJ-31899-76)*
- *The License Management Facility (AA-NZ27A-TE)*

Background Information

This section will cover these topics.

The technical implementation of LMF support into your product is the easy part. Before the engineering occurs, you must make a number of business decisions.

4.2 Using the License Management Facility

- Deciding upon your License policy
- Matching your licensing style to the License Management Facility
    - Pricing
- Designing your company's PAK documents
- Distributing PAKs
- Deciding product behaviors
- Deciding business practices and administrative impact
- Modifying your product

Background Information

First, and most basic, you must decide exactly what is your licensing policy. The LMF is a tool to apply technology to a licensing policy; it is not a tool to create one.

If PAKs are issued in conjunction with your licensing policies to enable software then it is important that you first clarify all the details of your licensing policy. Otherwise, PAKs may not end up reflecting your company's true policies.

What if you license by module? You will have to consider the ramifications on your administrative process of supplying a PAK for **each** module.

Also, you may currently provide alternate license styles. In fact, you might provide for the possibility activity, availability, **or** compound licenses for on or more of your products. Will you provide separate part numbers for each type of license for each product?

You may find new markets for your product, depending on if you license depending on the activity or on on the processor.

4.3 Deciding Upon Your License Policy

- What are your current Terms and Conditions?
- Do you use tiered, flat, or some other method of pricing?
- How do you handle updates to the software?
- Are they included in a service contract?
- Can they be ordered à la carte, at any time, by existing customers?
- What is the benefit to your customers?
- Do you license field test software?
- What about demos or loans of your software?
- Do you license versions of software?

These questions will give you information about how to create PAKs for your customers.

— **By the version**

"You have the right to use this and all preceding versions of this product."

— **All versions released within a specified time period**

"You have the right to use this, all preceding versions, and all versions that will be released in the next year."

— **Any and all future versions**

Background Information

What is the similarity between the license types supported by the LMF (Availability, Activity, and Compound) and your current policy? If you don't see one, you may not want to use the LMF or you'll need to decide how you want to use the LMF.

If you license by personal use, you can use the activity-based license and modify your product to supply an additional layer of enforcement.

4.4 Matching Your Licensing Style to the License Management Facility

Do the license types supported by the LMF match your current policy?

- Availability
- Activity
- Compound

Background Information

Pricing is closely aligned with licensing. Generally speaking, you charge a fee for a license. Just as there are many ways to license software, there are many ways to price it. One method of pricing is value pricing, assessing the value of this software product in a particular environment. Of course, questions of competitiveness, strategic direction, and market demand all figure in pricing as well.

The License Management Facility opens the way to a new method of pricing software, by the unit. Using your business rules for software pricing and an awareness of the LURT that establishes a tier by system type, you can develop a pricing curve such that the more units your users buy, the less they pay per unit.

Here is an example of how such a scheme might look. Note that when crossing into the next price tier, the user pays a base price which is the "unit price times maximum quantity" of the previous tiers.

Such a pricing strategy will allow your customers to buy exactly the number of license units they need for their system configuration.

Starting from this relatively simple starting point, you can develop policies for license upgrades (permitting customers to trade-in license units). You can also develop discounting structures for non-clustered environments (either based on units alone or on targeted platforms such as workstations).

4.4.1 Pricing

Table 4–1 illustrates a sample pricing strategy.

Table 4–1 Pricing Example

Number of Units	Base	Plus$/Unit
0-100	$0	$10.00
101-1000	$1000	$ 1.50
1001+	$2350	$ 0.50

Background Information

Design all the PAKs you are going to produce. Think very seriously about the appearance of your PAKs.

The DEC PAK Generator software enables you to **generate** your own PAKs. Refer to the *DEC LMF PAK Generator for VMS Application User's Guide* for more information.

You might want to work with a lawyer to ensure your business policies are accurately reflected in PAKs.

4.5 Designing Your Company's PAK Documents

Decisions regarding PAK contents (whether created on paper or not:)

- Decide upon an acronym for your company that will appear in the Producer and Issuer fields on a PAK. This will be unique among all other recipients of the PAK Generator.
- How many PAK types will you have and what will be the standard data elements required for each?
- Will you require unique PAKs and, if so, at what level?
- For example, Digital's license PAKs are unique. Digital's Service Update PAKs are unique by product (all PAKs for the same product are the same.) Digital's Temporary Service PAKs are unique for each product within a month.

Decisions regarding PAK documents (when printed):

- What will your PAK documents look like?
- Digital prints some PAK documents (those that represent licenses) on Bank Note paper which is customized and registered. This makes it easy to identify an official PAK document and also ensures that we can differentiate between an original and a copy.
- Think about the wording of any accompanying letters, or wording that you may want to have appear on the back of your PAK documents.

  You may want a lawyer to help you in this process.

Background Information

You must decide how you are going to supply your newly LMF-compliant product and accompanying PAK to your existing customers. You must also decide how you are going to package your PAK Certificates. This may be the first time that a license is a deliverable, and may necessitate changes in your order fulfillment system.

4.6 Distributing PAKs

- Decide how your PAKs will be manufactured, if required.
- Decide how you are going to supply your newly LMF-compliant product and the needed PAKs to your existing customers.

  In some cases, your customers will already have the software and you'll just need to provide the PAK. Will you provide PAKs on paper, in machine readable format, by telephone, or electronically? This may differ by PAK type.
- Decide how you are going to package each PAK document.
- How will you distribute PAKs (by mail, with kits)?
- Determine how you will convert existing customers to the new process, if this is required by your licensing policy.

  If you have a large installed base, this can be the most difficult part of the implementation. The availability of good license records will make this process much easier.
- Decide how you will handle Product Authorization Amendments (PAAMs) and, if so how you will handle them. (The PAK Generator does not generate PAAMs.)
- Determine if you will print PAKs in foreign languages.

  > NOTE
  > The PAK Generator doesn't currently support the creation of PAKs in foreign languages.
- Decide your policy for the situation when another vendor sells your software and how you will implement that policy.

Background Information

You must decide whether you want each software product to implement this support individually, or whether you want your engineers to write a single routine that behaves consistently, and which each of your products must use.

You need to determine what the risk is to customer satisfaction should your product refuse use. Who will they call for support, for example?

You may want to call LMF in multiple places in your product.

Your product may be comprised of several optional modules, each of which is licensed separately. For example, the part of your product that produces one type of output may be separate from the part that produces another type of output.

4.7 Deciding Product Behaviors

You must decide what you want to have happen if the LMF tells your product that it is not authorized to run. This could happen if the system manager has not registered the data, if there are not enough license units, or if the PAK has expired.

Will your product:

- Run upon completion of installation?

  If a product is run as part of the installation procedure, a customer should be advised to enter the PAK information before beginning installation.

- Refuse use?

  What's the risk if a product does not run, especially in a medical or production environment?

- Issue a message?

  Designing and implementing a Vendor Specific Policy Module (VSPM)– the concept of layering a routine (supplying consistent enforcement policies and error messages) between the core system services and a product– is appropriate if you have many products and want to have them exhibit a relatively consistent behavior with respect to the LMF.

  A Vendor Specific Policy Module can restrict access to your products by checking that the Issuer and/or Producer fields are yours.

- Write a message to an audit file?
- Do nothing at all?

Some of these decisions can be made for all products and should therefore be included in the Vendor Specific Policy Module. You may want to leave other decisions up to the specific software products.

Background Information

How will all facets of your business be affected? You may want to diagram your order administration process.

Your order administration system may use a Quality field to indicate the number of a specific product to ship to a customer. You might want to use the contents of the Quantity field to "trigger" the generation of a related PAK. However, you could be faced with sending out 100 PAKs for a 100 unit license of your XYZ compiler.

Will you need to have separate parts numbers for each incremental amount of your product (100 units, 200 units, 300 units)? This may result in many part numbers for a single product, which can be confusing.

You may want to generate Field Test PAKs (and other generic PAKs) manually since these PAKs do not represent licenses. Do you need to track this information?

You may find that modifications to your order administration systems are helpful.

4.8 Deciding Business Practices and Administrative Impact

- What triggers the generation of a PAK? This may differ by PAK type.
- How is PAK generation integrated with your invoicing (billing) system?
- Will you provide separate part numbers for each license?
- How will you handle version updates?

  Are updates traditionally supplied as a result of a service contract. Can customers buy updates independent of a service contract? Based on these practices, how will you ensure that all your customers, both new and existing, have the PAKs they need to run your software?
- Will you audit customers to ensure they're complying? If so, what is the role of the LMF and PAK documents?
- How will you support LMF-compliant products when technical or license policy questions arise? What is your policy when customers don't have the PAK they need?
- What changes are required to your administrative processes and systems to support the LMF and PAK implementations?

Background Information

Now you should have enough information to give to your software engineers to let them make the necessary code changes. If you are not using the Product Token or Hardware ID fields, about 25 lines of code should be added. There are examples in *The DEC VMS/LMF System Services Reference Manual.*

4.9 Modifying Your Product

1. Modifying existing code will amount to about 25 lines of code, more if you make use of the Product Token or Hardware ID fields.
2. To test it, you must generate a test PAK. How will you do this? Will you use the PAK Generator to accomplish this task? Then, register it on your test system.
3. Install and run your LMF-compliant product.
4. Make any necessary changes to your documentation.
5. Run a Field Test.

   This is a **highly recommended** step. You have changed your software product and your business practices. Test these changes with some of your customers.
6. You should be sure that you have a support plan and policy to ensure that customers can call you (not Digital) if there is a problem with your product or PAK.

Background Information

You should review these topics.

4.10 Summary

- Deciding upon your License policy
- Matching your product and terms to the License Management Facility
  - Pricing
- Designing your company's PAK documents
- Distributing PAKs
- Deciding product behaviors
- Deciding business practices and administrative impact
- Modifying your product

Appendix A
Case Study

A.1 Examining the Company

The case study described is fictional and designed to facilitate your understanding of business decisions regarding implemention of DDSLA/LMF.

Medical Equipment Company (MedEC)is a small corporation that is building a diverse set of highly-sophisticated medical software packages for data entry, retrieval, display, and reporting.

MedEC software runs on Digital's processors and has recently been given the opportunity to utilize Digital's License Management Facility as the technology for its license management architecture.

Like the rest of the software industry, MedEC does not sell its software, which results in the transfer of intellectual property and copyright, but rather licenses a particular customer to use its software on a particular computer or computers in restricted ways.

MedEC's software products:

1. **MedECmonitor (VAX, VMS, Workstation-based)**

   MedECmonitor is a computer-based data storage and analysis system designed to assist the clinician in the daily care of critically ill patients.

   The system:

   - Collects data
   - Analyzes and displays critical care patient information
   - Derives serial indices
   - Provides trends and pictorial displays of this data.
   - Provides medication calculation routines that the clinical staff uses to assure accuracy and timeliness

2. **MedEClaim (VAX Hardware, VMS System)**

   MedEClaim is an integrated, full-function administrative system for health maintenance organizations, preferred provider organizations, third-party administrators, and self-funded groups.

   The system provides administrative support including:

   - Enrollment and eligibility
   - Premium billing
   - Accounts receivable
   - Provider relations

- Benefit plans
- Claims processing
- Reference data
- Claims payment
- Trust fund processing
- Utilization review
- Commission and system reports

3. **MedECommunicate (VAX Hardware, VMS System)**

   MedECommunicate is an electronic mail gateway that seamlessly integrates another medical software vendor's MenuPlanner and ALL-IN-1 software.

   MenuPlanner, which is designed for long term health care facilities, provides menu planning and patient dietary profiles (linked to a common database) that include dietary restrictions and preferences. This system produces patient meal cards that are coded for special diets and production schedules.

   MedECommunicate allows ALL-IN-1 software users on the medical staff to send mail, including revisable menu cards, to MenuPlanner and vice versa. It operates over a common serial line connection.

4. **MedEC Rx (VAX Hardware, VMS System)**

   MedEC Rx, a pharmacy computer management system for retail, nursing home, hospital, and Health Maintenance Organization (HMO) pharmacies, provides:

   - Prescription filling (both new and refill)
   - Third-party billing
   - Sales analysis
   - Inventory control for all prescription medications

Customers have indicated that they would like indicating capability to:

- **Move a MedEC product, including its license, from one processor to another**

  If the system fails for any reason in an operating room, it's critical that the software run on another processor. Failover recovery must be simple, fast, and efficient.

  Current terms and conditions state that a software product should be used only on the hardware for which it is licensed. Customers must notify MedEC if they intend to transfer the software.

- **Limit the usage of software to individual CPUs in the cluster**

  MedEClaim is used only by the administrative departments in a hospital. It runs optimally on larger CPUs, and its use should be restricted to large processors for maximum performance.

- **Purchase highly specialized licenses for software used for long periods of time by only one or two users**

  Currently, the product cost assumes that the software will be to the full capacity of the system.

  For example, although MedECmonitor is be available for use in all areas of the hospital, it may only be utilized by a single operating room.

- **Obtain one piece of media**

  Each of MedEC's software products are provided to customers as separate pieces of media. Many customers buy all four products. The cost for preparing, producing, testing, and shipping media and documentation is a rising percentage of MedEC's cost of doing business.

  The current method of distributing disks, cartridges, tapes, and manuals can result in users not obtaining and using the most current software version (with the appropriate documentation and on-the-job training) when they need it.

MedEC feels that participation in the Software Business Technology program meets their needs because they want to:

1. Encourage customers to comply with their license terms and conditions by helping them verify that the software is running on the licensed hardware

   (Although MedEC is convinced that most of its customers want to pay the appropriate amount for specific software, they want to increase the compliance rate.)

2. Make it easier for their customers to manage license data
3. Distribute their products on consolidated distribution (for example, CDROM)
4. Formalize a pricing and licensing model that meets customers' usage and buying needs

   - **Fixed (Flat)**

     A fixed pricing scheme is appropriate for products if it doesn't matter on which processor the product is licensed. Communications gateway products, for example, generally has the same power or value on any size processor.

   - **Concurrent User**

     A concurrent user pricing scheme is appropriate for products that have a high duty cycle (long term use over, for example, the work day). Word processing software is a good example. Concurrent user pricing may not be appropriate for short duty cycle products like compilers, since a great number of potential users may have access to the product through a batch queue executing one job at a time.

   - **Tiered**

     A tiered pricing scheme is appropriate for products if the relative performance of the product differs from processor to processor. In some cases a short duty cycle (many users can access the product for a short amount of time) can influence pricing. Compilers or database management systems are good examples.

5. Manage the specific versions of a software product that a customer is licensed to execute

The Task

You are the Vice President of your company. Justify that DDSLA/LMF can handle MedEC business problems. Prepare a memo or presentation to the Board of Directors.

A.2 Expanding Business Opportunities

The case study described is fictional and designed to facilitate your understanding of business decisions regarding implemention of DDSLA/LMF.

MedEC has grown tremendously over the last few years but it still wants to continue to increase its market share.

Its latest product is MedEChart. The business manager is seeking new ways of marketing and distributing this software:

MedEChart, their first product that runs on both VMS and ULTRIX, provides nursing departments and hospital administration with data that quantifies nursing care requirements. The data can be plotted against scheduled staffing to help with daily and annual planning.

The product adapts to an agency's particular nursing model and can be operated manually or by computer, depending on the resources of the agency. It:

- Provides comparison to other agencies using the same system
- Adapts to changing nursing practices
- Responds to changes in staffing
- Links with an agency's quality assurance program

The Task

You are the MedEChart business manager. Propose the business opportunities that the Software Business Technologies program presents. Prepare a memo or presentation to your manager.

A.3 Supporting LMF-Compliant Products

The case study described is fictional and designed to facilitate your understanding of business decisions regarding implementation of DDSLA/LMF.

Because they are Digital customers, most MedEC customers are familiar with the License Management Facility. However, it's important that Customer Service and Sales be able to describe the purpose of the LMF and the types of licensing available.

As MedEC defines its own license management policies, it must determine a procedure for the situation when a customer:

- Throws away a PAK
- Wants two copies of the PAK
- Wants 200 more units
- Needs a temporary PAK

The Task

You are a customer service manager. Outline your company's policy for handling PAKs. Prepare a memo or presentation to your staff.

A.4 Making Decisions About Product Development

The case study described is fictional and designed to facilitate your understanding of business decisions regarding implementation of DDSLA/LMF.

One of MedEC's products, MedEC Rx is generally sold to pharmacies where the staff is not computer literate. Therefore, installation and use of the systems needs to be very simple, since the druggist may not be familiar with the computer system.

MedEC has just sold this product to a large chain of pharmacies.

Originally, this product was a two-part product (HOST and USER). To make the cost more attractive to large customers like the chain-store pharmacy, it's now becoming two separate products.

The HOST product operates on one CPU as a central server of all media (database node) and on all other CPUs in the same cluster as a requester of central services (remote node). The USER product on a remote node communicates with the remote node server, which then communicates with the central server that owns the databases.

MedEC used to sell the product as one kit with two identical (HOST) licenses. Now, they would like to sell two types of licenses (HOST and USER). The USER license would be less expensive.

However, MedEC has many customers who are already licensed to use the USER software with a HOST license.

The Task

You are an engineering manager. Outline the tasks your company must complete to make this two-part product LMF-compliant without requiring existing customers to purchase a USER license. Be sure to include information on product behavior. Prepare a memo or presentation to your staff.

Appendix B
Exercise Your Knowledge

Select one answer for each question that follows.

1. The License Management Facility, part of the VMS Version 5 operating system, is intended for use as an enforcement tool.

    a. True

    b. False

2. A software license is a contract between a software issuer and software user that:

    a. Grants ownership of a specific software product.

    b. Specifies the terms and conditions of ownership of a specific software product.

    c. Grants the right to execute a specific software product in accordance with the specified terms and conditions.

    d. Identifies the information that must be registered in the License database in order to use a specific software product.

3. An availability license is one that makes software available:

    a. To a certain number of users concurrently accessing it.

    b. To anyone on a particular processor or processors in a cluster.

    c. To specific users on a particular processor or processors in a cluster.

    d. To anyone on the local area network.

4. Compound licensing is a licensing style currently supported by the License Management Facility.

    a. True

    b. False

5. A Product Authorization Key (PAK):

   a. Contains information that a customer must register in the License database.

   b. Is usually printed on bank note paper and multiple copies are then sent to the customer.

   c. Contains product data that each user must register in the License database.

   d. Provides information to amend the license for an existing licensed software product.

6. A VMS system manager may use the LICENSE REGISTER command to enter PAK information into the License database.

   a. True

   b. False

7. To allow users to access a licensed product, a VMS system manager must activate each registered license by:

   a. Invoking the software product.

   b. Using the LICENSE REGISTER command.

   c. Entering the PAK information.

   d. Using the LICENSE LOAD command.

8. You can create an LMF-compliant product by:

   a. Designing a PAK document for your product.

   b. Including the system service calls to LMF in your product.

   c. Shipping LMF with your product.

   d. Restricting the use of your product to specific nodes.

9. The License Unit Rating Table specifies a set of license unit ratings for each:

   a. System marketing model

   b. VMS layered product

   c. System integrated product

   d. VMS server

10. A checksum is an encoded number calculated from the other information supplied with a PAK.

    a. True

    b. False

11. The combination of Issuer and Product Name uniquely identifies a software product.

    a. True

    b. False

12. The combination of Issuer and Authorization Number uniquely identifies a particular PAK.

    a. True

    b. False

13. If the Version field is used, it represents the:

    a. Highest version number of your software that is an LMF-compliant product.

    b. One version number of your product that the PAK will authorize.

    c. Lowest version number of your product that the PAK will authorize.

    d. Highest version number of your product that the PAK will authorize.

14. If used, the Product Release Date represents the latest release date of your software product that the PAK will authorize. If 1-JAN-1991 appears in this field and the current system date is 12-OCT-1991:

    a. All versions of the product released on or before 1-JAN-1991 are authorized.

    b. No versions of the product are authorized.

    c. Products with 12-OCT-1991 encoded are authorized.

    d. All versions of your software product are authorized.

15. You may include both a Version and a Product Release Date on a PAK.

    a. True

    b. False

16. If used, the Key Termination Date contains a date on which authorization to use the software product is terminated. It is recommended for use with:

    a. Loans and field test versions of software.

    b. Specifying the last date your product will be sold.

    c. Major releases of your software.

    d. Update releases of your software.

17. If you specify CONSTANT=0 in the Availability Table Code field (and no entry in the Activity Table Code field), it effectively grants unlimited use in any hardware configuration.

    a. True

    b. False

18. If there is an entry in the Activity Table Code field (such as F), the "charge" for the product is determined:

    a. When the license is loaded, based on a user on a specific processor.

    b. At execution time, based on a user on a specific processor.

    c. When the license is loaded, based on a user on any processor.

    d. At execution time, based on a user on any processor.

19. A unit is:

    a. An online table that specifies ratings for Digital processors.

    b. A unique identifier used to specify how much product use a license provides.

    c. An arbitrary measure used to specify a license size.

    d. An arbitrary value that indicates the version of the product.

20. If MOD_UNITS is specified as an option, the system manager will be able to use the LICENSE MODIFY command to:

    a. Specify a license size greater than the number of units on a PAK.

    b. Restrict users to using the software on a specific single node.

    c. Change the Activity Table Code entry on a PAK.

    d. Change the Availability Table Code entry on a PAK.

21. The contents of the Product Token and Hardware ID fields can be returned to your software product when it issues its system service call.
    a. True
    b. False

22. Digital's approach to Software Access Management is an extension of the copy-protection schemes of the Personal Computer software market.
    a. True
    b. False

23. Consolidated Software Distribution:
    a. Puts two or more separately licensed products in a single kit.
    b. Puts as many similar software products as possible on a set of distribution media.
    c. Requires Compact Disk, Read-Only Memory (CDROM) as the distribution medium.
    d. Uses a DECwindows application programming environment.

24. As part of SBT, Digital provides Online Documentation, which is:
    a. Final form documents distributed with software products.
    b. Online HELP documents embedded in software products.
    c. Documentation displayed by the Bookreader application.
    d. Computer-Based Instruction.

25. A Vendor Specific Policy Module is:

    a. The concept of layering a routine (supplying consistent enforcement policies and error messages) between the core system services and a product.

    b. The system service calls SYS$GRANT_LICENSE and SYS$RELEASE_LICENSE.

    c. The use of LMF as a software access management tool by Digital's third-party vendors.

    d. The concept of layering a user interface (that ensures compliance) between the core system services and a product.

26. Once you develop an LMF-compliant product, you must ship a PAK with every copy of the application..

    a. True

    b. False

Appendix C
Case Study Solutions

C.1 Examining the Case Study Company

In reviewing our customer feedback and our current business practices, I have concluded that developing DDSLA/LMF-compliant products might assist us and our customers in a number of areas:

- While DDSLA/LMF is not intended as an enforcement tool, use of the Product Token and Hardware ID fields (fields that Digital doesn't use) on our Product Authorization Keys can make LMF more restrictive in managing access to our products.

  It is possible, however, for users of our software products to circumvent the system by patching the operating system, patching our products, or even simply copying the PAK.

  Digital has discovered in licensing its own software products that limiting a license to a particular piece of hardware can be too restrictive, particularly if that processor is not available and the customer must move software to a backup processor.

  This limitation would be especially important in the our operating room emergency situations. By entering MOD_UNITS in the Key Options field on PAKs for capacity-based licenses, we can allow system managers to use the LICENSE MODIFY command to change the number of units on a PAK in an emergency.

  A Practical Guide to Managing Software Licenses (EJ-31899-76) recommends the use of the following to restrict access:

  — Batch queues allow system managers to route jobs automatically to licensed processors in compliance with license terms.

  — VMS Access Control Lists (ACLs) allows system managers to control product access on a node-by-node basis.

  — Digital Terminal Server products are supported by the LAT architecture. The LAT protocol provides many services. Some of these services can be used in configuring VAXcluster systems on which layered products are licensed to only a portion of the cluster.

- DDSLA/LMF is designed to facilitate the management of license data. In fact, we can encourage our users to register all their licenses on one processor in a cluster (for efficient record keeping) and then enable the licenses on the appropriate nodes in the cluster.

- When Product Authorization Keys are used to permit access to software, DDSLA/LMF is an enabling technology for consolidated distribution.

- Our products fit into the following models and license styles. DDSLA/LMF can support these business practices.

Table C–1 describes the MedEC products, pricing models, and licensing styles.

Table C–1 Product Pricing and Licensing

Product	Pricing Scheme	License Type	PAK Data
MedECmonitor	Concurrent User	User	Activity Table Code=100
MedECommunicate	Fixed (Flat)	Capacity	Availability Table Code=100
MedEClaim	Tiered	Capacity	Availability Table Code=F
MedEC Rx	Tiered	Capacity	Availability Table Code=F

Note that DDSLA/LMF does not **require** that we associate it with pricing. For example, by setting the Number of Units Availability Table Code fields to zero, we can disable most of the sizing attributes of the LMF. The remaining aspects are primarily for tracking and management which are mostly price independent.

DDSLA/LMF provides sizing computations and several default sizings for VAX processors. We may choose to associate pricing with that sizing. That pricing can have a one-to-one relationship with sizing, or it could include various discounts and other schedules that are outside of DDSLA/LMF.

It is **not** a goal of the DDSLA/LMF technology to bring about any commonality of pricing, sizing, or price-related business practices for non Digital products.

- Use of the Version and Product Release Date fields on our Product Authorization Keys may help us more accurately track which versions of our software products we've sold and are supporting.

In summary, our business practices are closely matched by the DDSLA/LMF.

C.2 Expanding Business Opportunities

After participating in Digital's *Guidelines for License Management* course, I'm convinced that we can use software business technologies to our advantage.

As we've observed recently, both ULTRIX and VMS systems often reside in the same hospital. We want our concurrent user licenses to be applicable for any Digital system, including RISC processor and all licenses applicable on any VAX system.

These strategies should help us to provide additional service for our customers (increasing customer satisfaction) and opportunities to generate revenue as well:

- **Software Loans**
  - The software loan PAK will represent a temporary license.
  - Use the Key Termination Date field on our Product Authorization keys to issue 90-day loans for each of our products. Each PAK will have a 120-day limit (with one month overlap for grace period and shipping delays).
- **Demo**
  - The demo PAK will not represent a license.
  - Develop special demo modes, training, or promotional materials for our products that exercise the unique features of our software.
  - Use the Product Token field to enable products to identify a demo mode.
  - Distribute our product and demo PAKs (including a Key Termination Date) at medical conferences or other sales opportunities.
  - Publish the PAK information in appropriate trade journals.

  NOTE
  Loan and demo PAKs will never be issued for production systems. Expiring software is unacceptable in most customer environments.

In addition, as a pilot program, I propose that we offer our new product, MedEChart, now in a field test, on a demo disk. The demo that is now under development will show the features of this new product, including online documentation and training, and ensure that our users utilize its full capacity. We could distribute this disk at the Medical Solutions Conference in Houston, Texas.

We could publish the PAK information in several trade journals, establish a 900 phone number, or set up an answering machine at the corporate office that supplies the PAK information. Our local sales offices will follow up where possible.

I suggest that we adopt these business practices to reach these goals. I look forward to meeting with you next week to discuss the implementation.

C.3 Supporting LMF-Compliant Products

Thank you for participating in our brainstorming session yesterday on the topic of customer service and Product Authorization Keys.

As you know, it's critical that we consistently adhere to the following four rules when responding to customer calls.

1. If a customer calls and requests a duplicate PAK because he threw it away or lost it before entering the information, direct the calls to Ms. Susan Smith, License Administration.

   She will generate a duplicate PAK. If the customer requests a new PAK, a new one will be generated and the customer will be encouraged to destroy the old PAK, if it is located.

   PAKs are packaged in a small box and are easy to overlook. Encourage customers to treat PAKs as the important documents they are.

2. If a customer calls and requests a duplicate PAK because he wants another copy, explain that the PAK is their proof of license. The customer can make a copy of the PAK if the PAK must be stored at multiple locations, but any copy (since we print our PAKs on bank note paper) will be visibly different from the original.

3. If a customer calls and requests additional units for cluster-wide licenses, direct all calls to Mr. John Arnold, Order Administration.

   Licenses are now separate from media, due to the introduction of CDROM. Customers who have the software may call and request licenses or additional units; this is handled by the normal order processes.

   For example, if the customer requires additional units of MedECmonitor, the customer must supply the contents of the PAK Authorization Number field. Mr. Arnold will verify that the license information is valid and generate a Product Authorization Amendment (PAAM) by using the same identification number.

4. If a customer calls and requires a temporary PAK to run one of our software products on a different processor than the one for which they're licensed, generate a new, temporary PAK by using the PAK Generator. As the Product Release Date entry, enter the date of the next scheduled release.

   Key termination dates should not be generated for any emergency cases, unless the customer is field testing our product; in that case, enter the field test completion date.

   The date encoded in the product should be less than the date on the PAK. When the new version is released, the later version will contain a date that's greater than the PAK date and the product will no longer run.

C.4 Making Decisions About Product Development

As we discussed at this morning's staff meeting, the following tasks are assigned to the people listed. Please let me know if you have any questions regarding the schedule.

As we decided, the USER software will run if the HOST PAK information has been entered into the license database. This arrangement accommodates existing customers, who already have two HOST licenses, but are really running the new USER software.

As you know, we chose not to distribute generic PAKs (0 units), but rather to incur the cost of generating customized kits for our preferred customers who don't want the added burden of entering the PAK information.

Table C–2 outlines the team member, task, and time allowed to complete the task.

Table C–2 Work Chart

Person	Task	Time Allowed
Dave Reynolds	Design PAK layouts and text for terms and conditions of PAKs.	One month
Pam Whalen	Develop/test the MedEC Vendor Specific Policy Module supplying consistent enforcement policies and error messages.	One week
Jane Jones	Modify MedEC Rx code to support the new license policy that MedEC Rx-USER should run if Rx-HOST PAK information is in the license database.	One week

Table C–2 (Cont.) Work Chart

Person	Task	Time Allowed
Al Ross	Modify installation procedure to embed the PAK information into the special large company kits.	One week

MedEC Vendor Specific Policy Module messages should be:

- If no license is active:

  No license is active for this software product

- If usage exceeds license limits:

  Attempted usage exceeds active license limits

- If the product release date is invalid:

  License is invalid for this product release date

- If the version is invalid:

  License is invalid for this product version

- If an active license has terminated:

  License has terminated

Appendix D
Exercise Your Knowledge Solutions

One correct answer is indicated for each question.

1. The License Management Facility, part of the VMS Version 5 operating system, is intended for use as an enforcement tool.

   a. True

   b. *False*

   The **Enforcement** section provides more detail.

2. A software license is a contract between a software issuer and software user that:

   a. Grants ownership of a specific software product.

   b. Specifies the terms and conditions of ownership of a specific software product.

   c. *Grants the right to execute a specific software product in accordance with the specified terms and conditions.*

   d. Identifies the information that must be registered in the License database in order to use a specific software product.

   The **License Management** section provides more detail.

3. An availability license is one that makes software available:

   a. To a certain number of users concurrently accessing it.

   b. *To anyone on a particular processor or processors in a cluster.*

   c. To specific users on a particular processor or processors in a cluster.

   d. To anyone on the local area network.

   The **Types of Licensing** section provides more detail.

4. Compound licensing is a licensing style currently supported by the License Management Facility.

   a. *True*

   b. False

   The **Types of Licensing** and **Compound Licensing** sections provide more detail.

5. A Product Authorization Key (PAK):

   a. *Contains information that a customer must register in the License database.*

   b. Is usually printed on bank note paper and multiple copies are then sent to the customer.

   c. Contains product data that each user must register in the License database.

   d. Provides information to amend the license for an existing licensed software product.

   The **LMF Components** section provides more detail.

6. A VMS system manager may use the LICENSE REGISTER command to enter PAK information into the License database.

   a. *True*

   b. False

   The **License Database (LDB) and Utility** section provides more detail.

7. To allow users to access a licensed product, a VMS system manager must activate each registered license by:

   a. Invoking the software product.

   b. Using the LICENSE REGISTER command.

   c. Entering the PAK information.

   d. *Using the LICENSE LOAD command.*

   The **LMF Components** section provides more detail.

8. You can create an LMF-compliant product by:

   a. Designing a PAK document for your product.

   b. *Including the system service calls to LMF in your product.*

   c. Shipping LMF with your product.

   d. Restricting the use of your product to specific nodes.

   The **LMF Components** section provides more detail.

9. The License Unit Rating Table specifies a set of license unit ratings for each:

   a. *System Marketing Model*

   b. VMS layered product

   c. System Integrated Product

   d. VMS server

   The **License Unit Rating Table (LURT)** section provides more detail.

10. A checksum is an encoded number calculated from the other information supplied on with a PAK.

    a. *True*

    b. False

    The **Product Authorization Keys (PAKs)** section provides more detail.

11. The combination of Issuer and Product Name uniquely identifies a software product.

    a. True

    b. *False*

    The **Producer** and **Product Name** sections provide more detail.

12. The combination of Issuer and Authorization Number uniquely identifies a particular PAK.

    a. *True*

    b. False

    The **Issuer** and **Authorization Number** sections provide more detail.

13. If the Version field is used, it represents the:

    a. Highest version number of your software that is an LMF-compliant product.

    b. One version number of your product that the PAK will authorize.

    c. Lowest version number of your product that the PAK will authorize.

    d. *Highest version number of your product that the PAK will authorize.*

    The **Version** section provides more detail.

14. If used, the Product Release Date represents the latest release date of your software product that the PAK will authorize. If 1-JAN-1991 appears in this field and the current system date is 12-OCT-1991:

    a. *All versions of the product released on or before 1-JAN-1991 are authorized.*

    b. No versions of the product are authorized.

    c. Products with 12-OCT-1991 encoded are authorized.

    d. All versions of your software product are authorized.

    The **Product Release Date** section provides more detail.

15. You may include both a Version and a Product Release Date on a PAK.

    a. True

    b. *False*

    The **Product Release Date** section provides more detail.

16. If used, the Key Termination Date contains a date on which authorization to use the software product is terminated. It's recommended for use with:

    a. *Loans and field test versions of software.*

    b. Specifying the last date your product will be sold.

    c. Major releases of your software.

    d. Update releases of your software.

    The **Key Termination Date** section provides more detail.

17. If you specify CONSTANT=0 in the Availability Table Code field (and no entry in the Activity Table Code field), it effectively grants unlimited use in any hardware configuration.

    a. *True*

    b. False

    The **Availability Table Code** section provides more detail.

18. If there is an entry in the Activity Table Code field (such as F), the charge for the product is determined:

    a. When the license is loaded, based on a user on a specific processor.

    b. *At execution time, based on a user on a specific processor.*

    c. When the license is loaded, based on a user on any processor.

    d. At execution time, based on a user on any processor.

    The **Activity Table Code** section provides more detail.

19. A unit is:

    a. A online table that specifies ratings for Digital processors.

    b. A unique identifier used to specify how much product use a license provides.

    c. *An arbitrary measure used to specify a license size.*

    d. An arbitrary value that indicates the version of the product.

    The **Number of Units** section provides more detail.

20. If MOD_UNITS is specified as an option, the system manager will be able to use the LICENSE MODIFY command to:

    a. *Specify a license size greater than the number of units on a PAK.*

    b. Restrict users to using the software on a specific single node.

    c. Change the Activity Table Code entry on a PAK.

    d. Change the Availability Table Code entry on a PAK.

    The **Key Options** section provides more detail.

21. The contents of the Product Token and Hardware ID fields can be returned to your software product when it issues its system service call.

    a. *True*

    b. False

    The **Product Token** and **Hardware ID** sections provide more detail.

22. Digital's approach to Software Access Management is an extension of the copy-protection schemes of the personal computer software market.

    a. True

    b. *False*

    The **Software Access Management** section provides more detail.

23. Consolidated Software Distribution:

    a. *Puts two or more separately licensed products in a single kit.*

    b. Puts as many similar software products as possible on a set of distribution media.

    c. Requires Compact Disc, Read-Only Memory (CDROM) as the distribution medium.

    d. Uses a DECwindows application programming environment.

    The **Consolidated Software Distribution** section provides more detail.

24. As part of SBT, Digital provides Online Documentation, which is:

    a. Final form documents distributed with software products.

    b. Online Help documents embedded in software products.

    c. *Documentation displayed by the Bookreader application.*

    d. Computer-Based Instruction.

    The **Online Documentation** section provides more detail.

25. A Vendor Specific Policy Module is:

    a. *The concept of layering a routine (supplying consistent enforcement policies and error messages) between the core system services and a product.*

    b. The system service calls SYS$GRANT_LICENSE and SYS$RELEASE_LICENSE.

    c. The use of LMF as a software access management tool by Digital's third-party vendors.

    d. The concept of layering a user interface (that ensures compliance) between the core system services and a product.

    The **Deciding Product Behaviors** section provides more detail.

26. Once you develop an LMF-compliant product, you must ship a PAK with every copy of the application.

    a. True

    b. *False*

    The **Distributing PAKs** section provides more detail.

Glossary

activity license: Defines the number of concurrent uses allowed for any product. Each product defines an activity as either an interactive user, a running process, or a job. For example, a license for four activities 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 or PAAM. The PAK issuer name and authorization number identify a license.

availability license: Makes a product available to all 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 or PAAM. The checksum string always begins with the number 1, which is the only number in the string. The other sixteen positions are always alphabetic characters from A through P.

compound license: A license that is a combination of an availability license and an activity license. For example, a product may require a number of license units to operate on a processor *and* it may require more license units for multiple users to access it.

DDSLA: Digital Distributed Software Licensing Architecture

hardware identifier: An optional string that identifies a particular hardware unit.

Integrated Software Business Technologies: Digital'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 (CDROM), which authorizes access by PAKs and LMF.

license amendment: Updates an existing license by entering a Product Authorization Amendment (PAAM) in the License database. Digital is not currently using PAAMs. In the future, Digital may allow license amendment using the LICENSE AMEND command or the VMSLICENSE.COM command procedure.

license combination: Uses the license units from two identical 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 VMS licenses are registered in a second License database.

License Management Facility (LMF): A variety of system-level software components used to maintain software license information in the License database of the VMS operating system. LMF is a management tool; the terms and conditions of your product contract determine your legal use of software.

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: Allows more than one processor to use the license units from a single license. In VMS, this refers to sharing licenses among nodes in a VAXcluster environment. Licenses that specify the NO_SHARE option cannot be shared.

license unit: A basic unit of measurement that Digital uses to specify how much product use a license provides. Digital gives each license intended for use 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 Rating Tables (LURTs): Online tables provided by Digital that specify a series of license unit requirements, essentially performance ratings, for each System Marketing Model. High-performance processors (other ratings may be unrelated to performance) have greater license unit requirements.

PAK identification: The Product Authorization Key (PAK) issuer name and the authorization number. Together, they uniquely identify a license.

PAK issuer: The company that creates the license contract for the software. The PAK issuer name and license authorization number uniquely identify a license. PAK issuers are usually the same as software producers but can operate under agreement with the producer.

Product Authorization Amendment (PAAM): Provides information to amend the license for an existing licensed software product. Digital is not currently using PAAMs. Without a current PAK or the appropriate PAAM, you may not be able to use an installed software product. In the future a PAK issuer may produce the PAAM and transmit it to you by mail, electronic transfer, or by telephone. A PAAM contains a unique authorization checksum and the information needed to amend current license information.

Product Authorization Key (PAK): Contains essential licensing 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.

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

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.

software group: Two or more software products authorized by a single license name. For example, VAX BASIC and VAX COBOL may be offered as the LANGUAGE_CLUB.

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

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

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

termination date: Defines 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. Tokens are reserved for future use.

Index