VMS
System Manager's Manual

Order Number: AA-LA00A-TE

April 1988

This manual provides the basic concepts and procedures for VMS system management; it is especially intended for managers of small clusters and systems.

Revision/Update Information: This is a new manual.

Software Version: VMS Version 5.0

digital equipment corporation
maynard, massachusetts

April 1988

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem-10	PDP	VT
DECSYSTEM-20	PDT	
DECUS	RSTS	
DECwriter	RSX	

digital™

ZK3388

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

USA*

Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire
03061

CANADA

Digital Equipment
of Canada Ltd.
100 Herzberg Road
Kanata, Ontario K2K 2A6
Attn: Direct Order Desk

INTERNATIONAL

Digital Equipment Corporation
PSG Business Manager
c/o Digital's local subsidiary
or approved distributor

In Continental USA, Alaska, and Hawaii call 800-DIGITAL.

In Canada call 800-267-6215.

*Any order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575).

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminster, Massachusetts 01473.

Production Note

This book was produced with the VAX DOCUMENT electronic publishing system, a software tool developed and sold by DIGITAL. In this system, writers use an ASCII text editor to create source files containing text and English-like code; this code labels the structural elements of the document, such as chapters, paragraphs, and tables. The VAX DOCUMENT software, which runs on the VMS operating system, interprets the code to format the text, generate a table of contents and index, and paginate the entire document. Writers can print the document on the terminal or line printer, or they can use DIGITAL-supported devices, such as the LN03 laser printer and PostScript® printers (PrintServer 40 or LN03R ScriptPrinter), to produce a typeset-quality copy containing integrated graphics.

® PostScript is a registered trademark of Adobe Systems, Inc.

Contents

Chapter 4 Managing Users

Chapter 5 Performing Batch and Print Operations

Chapter 6 Setting Up a Network

Chapter 7 Setting Up a Local Area VAXcluster Environment

Chapter 8 BACKUP Procedures

Chapter 9 Maintaining Acceptable System Performance

SYSTEM MANAGER'S REFERENCE

Examples

Figures

Tables

Preface

The *VMS System Manager's Manual* provides system managers with the concepts and procedures needed to manage daily operations on a VMS system. This manual contains a subset of the information included in the Extended VMS System Management documentation subkit.

Intended Audience

This manual can be used by anyone who performs the functions of a system manager or operator on a VMS system. It is especially intended for managers of small clusters and systems.

Document Structure

The *VMS System Manager's Manual* is divided into two main sections: System Management Tasks and Reference.

Part I (Chapters 1 through 11) are task-oriented descriptions of the functions that are generally assigned to system managers. Part II, the Reference section, documents the utilities that serve as system management tools on a VMS system.

Chapter 1 describes each chapter in some detail. Read Chapter 1 to determine which of the remaining chapters in the book are appropriate for your needs.

The Reference section contains quick reference information on the VMS system management utilities. Each utility chapter includes a usage summary and a subset of frequently used commands and qualifiers.

The Reference section includes the following utilities:

- Accounting Utility
- Analyze/Disk_Structure Utility
- Authorize Utility
- Backup Utility
- Bad Block Locator Utility
- Error Log Utility

- Exchange Utility
- Install Utility
- LAT Control Program Utility
- Mount Utility
- Network Control Program (NCP) Utility
- SYSGEN Utility
- SYSMAN Utility
- Terminal Fallback Utility

Associated Documents

In the VMS Base documentation set:

- For guidance in finding additional system management information, see the *Overview of VMS Documentation.*
- For general information about how to use a VMS system, see the *VMS General User's Manual.*

In the extended VMS documentation set:

(The extended VMS documentation set is the complete set of software manuals for the VMS operating system. For information about ordering any of the manuals in the extended VMS documentation set, see the *Overview of VMS Documentation* or contact your DIGITAL representative.)

- For general background information about the system, see the *Introduction to VMS.*
- For more information on setting up the system for operation, see the *Guide to Setting Up a VMS System.*
- For more information on maintaining the system, see the *Guide to Maintaining a VMS System.*
- For information on security management, see the *Guide to VMS System Security.*
- For more information on networking, see the *Guide to DECnet–VAX Networking.*
- For more information on VMS clusters, see the *VMS VAXcluster Manual.*
- For more information on performance tuning, see the *Guide to VMS Performance Management.*
- For more information on utilities, see the individual VMS utility manuals.
- For complete descriptions of DCL commands, see the *VMS DCL Dictionary.*

- For descriptions of system messages and suggested user action, see the *VMS System Messages and Recovery Procedures Reference Volume*.

Other related documentation:

- For information on system installation and other processor-specific procedures, see your VAX processor installation and operations guide.
- If you have purchased the volume shadowing option, see the *VAX Volume Shadowing Manual* for information on creating and maintaining volumes using volume shadowing.
- If you have purchased the RMS journaling option, see the *VAX RMS Journaling Manual* for information on RMS journaling.
- For hardware operating instructions, see the appropriate hardware owner's manual.

Conventions

Convention	Meaning
RET	In examples, a key name (usually abbreviated) shown within a box indicates that you press a key on the keyboard; in text, a key name is not enclosed in a box. In this example, the key is the RETURN key. (Note that the RETURN key is not usually shown in syntax statements or in all examples; however, assume that you must press the RETURN key after entering a command or responding to a prompt.)
CTRL/C	A key combination, shown in uppercase with a slash separating two key names, indicates that you hold down the first key while you press the second key. For example, the key combination CTRL/C indicates that you hold down the key labeled CTRL while you press the key labeled C. In examples, a key combination is enclosed in a box.
$ SHOW TIME 05-JUN-1988 11:55:22	In examples, system output (what the system displays) is shown in black. User input (what you enter) is shown in red.
$ TYPE MYFILE.DAT . . .	In examples, a vertical series of periods, or ellipsis, means either that not all the data that the system would display in response to a command is shown or that not all the data a user would enter is shown.
input-file, . . .	In examples, a horizontal ellipsis indicates that additional parameters, values, or other information can be entered, that preceding items can be repeated one or more times, or that optional arguments in a statement have been omitted.
[logical-name]	Brackets indicate that the enclosed item is optional. (Brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks (″). The term apostrophe (′) is used to refer to a single quotation mark.

Chapter 1

Introduction

The VMS operating system and the other software products that run on your computer provide you and the other users on your system with a wide range of computing capabilities. In order to create and maintain a proper and efficient computing environment, certain administrative tasks must be undertaken. These tasks are called *system management,* and they include the following:

- Setting up the system
- Giving individual users access to the system
- Installing software (and software updates)
- Maintaining acceptable performance levels
- Preventing the loss of important information that you keep on line
- Making sure that the system is secure
- Handling media (such as disks or magnetic tapes)
- Setting up the software to allow for printers and for batch jobs
- Setting up a cluster
- Setting up a network

As system manager, you may need to do some of these tasks only once (for example, setting up software to allow for printers or batch jobs, or setting up a network); others are done on a continuing basis (for example, maintaining system security and preventing the loss of data). At some sites, one or more people are designated as system managers, and other individuals are designated as *operators.* In these cases, operators are responsible for tasks such as physically mounting magnetic tapes and disks, monitoring printers, responding to emergencies or security alarms, and maintaining system log files.

Not all of the tasks described in this manual may be necessary for your site. This chapter provides an overview of the information that this manual contains. You should read this introductory chapter to determine which parts of the manual may be applicable to your site.

1.1 Who Should Use This Manual?

This manual is divided into two parts. Part I (Chapters 1 through 11) describes the tasks that managers of small standalone systems and Ethernet-based (low end) VAXcluster configurations are likely to encounter. Part II is a condensed reference section describing the system management tools that the VMS operating system provides.

If you are a manager of a small system or cluster, you can use this manual for most or all of your system management tasks. Managers of all types of systems can use Part II of this manual as a centralized source of information for system management utilities. You should also be aware that expanded documentation exists for all of the topics discussed in this manual. See the *Overview of VMS Documentation,* included in the base set documentation, for a complete list of the technical manuals for the VMS operating system.

The next sections describe the remaining chapters in this manual. Read these sections to determine which parts of this manual are applicable to your site.

1.1.1 Chapter 2 — Starting Up the System

This chapter describes the procedure for starting up your system, for the first time and for subsequent system startups. It discusses how to customize your startup procedure, so that your system automatically provides the proper environment each time that the system is started. The chapter also tells you how to shut down the system in an orderly manner.

All managers of small systems and clusters should read this chapter. It describes the procedures that are needed to boot your system and to create a proper environment for the users on your system.

1.1.2 Chapter 3 — Installing Software

Software such as the VMS operating system and any layered products that you use must be installed on your system. You must also use a similar procedure when you upgrade software (that already exists on your system) to a more recent version. Chapter 3 describes the procedures that you should follow when you install or upgrade software. This chapter also tells you how to remove a software product that has previously been installed.

All managers of small systems or clusters should read this chapter, because it contains information that system managers need when installing the VMS operating system and layered products. (In addition to the information in this chapter, you will also need the specific installation instructions for the software you want to install.)

1.1.3 Chapter 4 — Managing Users

Chapter 4 describes how to give access to users on your system. It tells you how to add and maintain user accounts, and it describes the privileges that you can give and the resources that you can allocate to the users on your system.

If you are a manager of a system with more than one user, or if you are the manager of a single-user system and would like more than one user account on your system, you should read this chapter.

1.1.4 Chapter 5 — Performing Print and Batch Queues

Chapter 5 tells you how to set up and maintain queues for printers and for batch jobs. If one or more printers are connected to your system, then you must have a print queue in order to use them. (You do not, however, need a print queue to use a hardcopy terminal.) If you want batch processing to be available to users, then you must also establish one or more batch queues.

If you have a printer connected to your system, or if you want to use batch processing capabilities, you should read this chapter.

1.1.5 Chapter 6 — Setting Up a Network

A computer *network* allows you to exchange information between two or more individual computers. In the VMS operating system, the DECnet–VAX product provides networking capabilities. In order to use the DECnet–VAX functions, you must have the appropriate hardware and software.

Chapter 6 tells you how to set up a basic network using DECnet–VAX. The chapter describes how to set up the basic network control functions that allow you to communicate with other systems, and it also tells you how to control certain network functions such as stopping and restarting the network, monitoring network activity, and so on.

If you are a manager of a small system that is part of a network of computers or if you are a manager of an Ethernet-based (low end) cluster, then you should read this chapter.

1.1.6 Chapter 7 — Cluster Configurations

A *cluster* is a group of two or more processors that share some or all of their resources. When a group of VAX processors share resources in a VAXcluster environment, the storage and computing resources of all of the processors are combined, which can increase the processing. capability, communications, and availability of your computing system. Clusters also provide an environment in which additional computers can easily be added.

Chapter 7 tells you how to create a VAXcluster environment. It discusses the software and hardware that is required, the various types of VAXcluster configurations, how to use DECnet–VAX functions in your cluster, and the resources that you can share in the cluster.

If you manage a cluster, you should read this chapter.

1.1.7 Chapter 8 — Backup Procedures

One of the best ways to prevent the loss of important data on any system is to make *backup* copies of your data at regular intervals. A backup copy is a reserve copy of the data that you keep in a safe place (for example, on a magnetic tape, or on a different disk). If the data that is on line is lost (for example, because of inadvertent deletion or a hardware failure), you can use the backup copy of the data.

All system managers should have a plan for regular backups of the data on their systems. Chapter 8 describes procedures for making full and incremental backups for your system, and it also tells you how to restore the data from a backup copy. This chapter contains essential information for all managers of small systems.

1.1.8 Chapter 9 — Maintaining Acceptable Performance

The *performance* of a system refers to the speed of interactive and batch processing. Performance can be measured in the response time for interactive processing and the time that it takes to complete batch processing jobs.

Chapter 9 describes some of the actions that you can take to optimize your system's performance. This chapter tells you how to monitor the use of system resources; it shows you how to reset system parameters to optimize performance, and it provides some hints for making some other performance improvements.

In most cases, performance tuning is not necessary for small systems. The VMS operating system provides tools that automatically set system parameters that provide for optimum performance. This chapter is useful for acquiring background information about performance issues, or for determining whether your system performance might benefit from some additional tuning.

1.1.9 Chapter 10 — Operator Tasks

Chapter 10 describes maintaining media, maintaining print devices, system problem diagnosis and recovery, error log issues, the operator console, sending system messages to interactive users, and other functions that may be assigned to an operator. When there is no individual designated as the operator, these tasks may be the responsibility of the system manager.

You should read this chapter if you are the manager or an operator in a small system environment.

1.1.10 Chapter 11 — System Security Issues

Chapter 11 discusses security issues in the context of a small system or cluster. These issues include basic security for single-user and multi-user systems, network security, user privileges (including rights and proxies), system passwords, and ongoing security practices (such as security audits).

Security is important for any system, and no system manager should take security for granted. Although some parts of this chapter may not be applicable to all sites, all system managers should read this chapter in order to provide a secure data processing environment.

1.1.11 Part II — Reference Section

Part II provides reference information for VMS utilities that you can use for system management tasks. For each utility, the Reference section provides a brief description of the utility, format statements for using the utility, and a description of the commands and qualifiers that you can use with the utility.

The following utilities are included in Part II of this manual:

- Accounting Utility
- Analyze Disk Structure Utility
- Authorize Utility
- Backup Utility
- Bad Block Locator Utility
- Error Log Utility
- Exchange Utility
- Install Utility
- LATCP Utility
- Mount Utility
- NCP Utility
- SYSMAN Utility
- System Generation Utility
- Terminal Fallback Utility

1.2 System Management Concepts and Terms

Some concepts and terms are used frequently in system management, and you should become familiar with them. The following terms and concepts are used both in the context of everyday general use in a VMS system and in the context of system management; they are described in the *VMS General User's Manual*:

- **Accounts and Directories**
- **Command procedures**
- **DIGITAL Command Language (DCL)**

The following concepts and terms apply primarily to system management:

- **SYSTEM account and [SYSMGR] directory**

  The SYSTEM account is reserved for use by the system manager. When you log in to the SYSTEM account, your default directory (which is also reserved for the system manager) is SYS$SYSROOT:[SYSMGR].

  Always be careful when you are logged in to the SYSTEM account. When you are logged in to the SYSTEM account, all privileges are enabled, by default. You need these privileges to perform many system management tasks; however, they can also produce unwanted or even destructive results if you use them carelessly.

- **Console (Operator's) terminal**

  You can perform most system management tasks from any terminal that is connected to the processor (or the cluster). However, certain tasks such as bootstrapping the system and communicating with the VAX processor's console subsystem must be performed at a special terminal called the *console terminal.*

  The console terminal, which always has the designation OPA0, is also usually designated as the *operator's terminal.* You use the operator's terminal to send messages to system users and respond to user requests, using the operator communication process (OPCOM).

Chapter 2

Starting Up the System

The system startup procedure establishes the computing environment for your system.

This chapter covers three major topics:

- How to start your system for the first time
- How to create the proper computing environment whenever you restart your system
- How to shut down your system

Before you can use the procedures described in this chapter, you must first set up the hardware for each VAX processor. To set up the hardware and install the VMS operating system, refer to the instructions in your VAX processor installation and operations guide. After you have installed the operating system, you will be able to log into the SYSTEM account on your computer.

After the operating system has been successfully loaded, you can set up the proper computing environment for your system. The site-specific system startup file, SYSTARTUP_V5.COM, is an essential aspect of establishing an environment specific to the needs of your site. Section 2.4 describes how to modify SYSTARTUP_V5.COM to meet the needs of your site.

2.1 Starting Up Your System for the First Time

Instructions for installing the VMS operating system are included in the installation and operations guide for your processor. You must choose whether you are installing the VMS operating system as a *new installation* or as an *upgrade*. If you are installing the VMS operating system for the first time, you must use the new installation procedure. If you already have a previous version of the VMS operating system on your processor, then you should use the upgrade procedure. Instructions for a new installation are found in your processor installation and operations guide; instructions for an upgrade procedure are found in the Release Notes for the VMS operating system.

When you install the VMS operating system using the new installation procedure, the disk on which you install the operating system is first erased, and then a directory structure and the operating system itself is put in place. When you use the upgrade procedure, the files for the VMS operating system are replaced (with files for the upgraded operating system), and all other files on your system disk (for example, data files, executable images that are not part of the operating system, and so on) remain as they are.

CAUTION: If you use the new installation procedure for a processor that already has a previous version of the VMS operating system, you will destroy the previous contents of the disk that you designate as the system disk.

2.2 Booting the System

Booting is the process of loading the operating system from the system disk into processor memory. You can perform either a *nonstop* boot or a *conversational* boot. A nonstop boot is the quickest and easiest method, and the operating system will automatically set system parameters that are appropriate for most computing activities for your system's hardware configuration. A conversational boot requires you to supply more information during the boot process, but it allows you to change system parameters during the boot procedure. See your VAX processor installation and operations guide for detailed booting instructions.

After a system shuts down, it must be rebooted before you can use it. Some processors provide the capability of an automatic reboot; when you enable this feature, the system automatically attempts to reboot itself after it has been shut down. For example, if your site experiences a power failure, a processor that has automatic reboot enabled restarts itself automatically after the power has been restored. See your VAX processor installation and operations guide for information about automatic rebooting.

In unusual cases, the normal startup procedures will not work properly and troubleshooting may be necessary. Section 2.9 describes procedures that you should follow if the normal startup procedure fails, or if you find yourself locked out of your system.

2.3 Logging In to the New System

When the boot procedure is complete, a message is displayed on the terminal from which the system is booted (except on workstations, where the message is displayed on the operator's window). The message is similar to the following:

```
  VAX/VMS Version 5.0 <dd-mmm-yyyy hh:mm:ss.s>

%%%%%%%%%%  OPCOM, <dd-mmm-yyyy hh:mm:ss.s>  %%%%%%%%%%
Logfile has been initialized by operator _OPA0:
Logfile is SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;1

%SET-I-INTSET, login interactive limit = 64, Current interactive value = 0
  SYSTEM       job terminated at <dd-mmm-yyyy hh:mm:ss.s>
```

After you see this display, you can then log in to the system manager's account, using the following procedure:

1. Press the RETURN key on the console terminal.
2. In response to the system's request for your *username*, type SYSTEM.
3. In response to the system's request for your *password*, type the password that you chose for the SYSTEM account during installation. You should change your system password immediately after logging in to the system for the first time. To change your password, enter the DCL command SET PASSWORD.

CAUTION: DIGITAL recommends that you change the system manager's account password frequently to maintain system security. The system manager's account has full privileges by default; therefore, you should exercise caution when using it.

After you enter your password, the system prints a welcome message on the console terminal. If it is not your first time logging in, the system also prints the time of your last login, for example:

```
Welcome to VAX/VMS Version n.n

Last interactive login at 15-APR-1988 15:13:21.07
```

The command procedure SYS$EXAMPLES:MGRMENU.COM generates the system manager menu. This command procedure can serve as a sample for designing site-specific system manager menus.

2.4 Startup Command Procedure for Your Site (SYSTARTUP_V5.COM)

A command procedure that sets up a computing environment for the specific needs of your site is executed each time that your system starts up. This file is located in the system manager's directory, [SYSMGR], and it is called SYSTARTUP_V5.COM. In order to customize SYSTARTUP_V5.COM for the needs of your site, you must make the appropriate edits to the file. This section describes how to customize the SYSTARTUP_V5 command procedure.

After you install the VMS operating system, the file SYSTARTUP_V5.COM is placed in the [SYSMGR] directory. SYSTARTUP_V5.COM is a template file, which means that it can be used as a basis or guide for creating a startup file that suits your own system. In particular, the SYSTARTUP_V5.COM template includes sections that can perform the following tasks at startup time:

- Mounting public disks
- Setting the characteristics of terminals and other devices
- Initializing and starting queues
- Installing known images
- Starting up the DECnet network
- Running the System Dump Analyzer
- Purging the operator's log file
- Submitting batch jobs that are run at system startup time
- Limiting the number of interactive users
- Starting up the LAT network
- Site-specific LAT command procedure
- Creating systemwide announcements
- Defining a system login command procedure
- Backing up the system

To modify SYSTARTUP_V5.COM, you can use any text editor. This file is a command procedure, so any changes that you make must conform to the rules for command procedures, as described in the *VMS General User's Manual*. In order to be used as a site-specific startup file, be sure to keep the file in the [SYSMGR] directory and use the file name SYSTARTUP_V5.COM.

To allow SYSTARTUP_V5.COM to continue in the event of an error, include the DCL command SET NOON at the beginning of the file, as follows:

```
$ SET NOON
```

This command disables error checking after the execution of each command in SYSTARTUP_V5.COM.

The following sections describe many of the elements of your user environment that you can establish with SYSTARTUP_V5.COM.

2.4.1 Mounting Public Disks

A *public disk* is a disk that can be accessed by any system process. In order to make a public disk available for use, the disk must be physically mounted and you must then use the MOUNT command. You do not need to use the mount command for the system disk, because the system disk is already mounted when the system starts up.

This section describes how to mount disks in the SYSTARTUP_V5.COM file. If your system uses any disks that should be mounted whenever the system is running, you should read this section.

To include MOUNT commands in SYSTARTUP_V5.COM to mount your public disks for systemwide access, use the following MOUNT command syntax:

```
$ MOUNT/SYSTEM ddcu: volume_label logical_name
```

You use the /SYSTEM qualifier to mount the disk for systemwide access; this is called a *public volume*. If you are in a VAXcluster environment, then you should also use the /CLUSTER qualifier in order to make the volume accessible to any user in the cluster.

The expression *ddcu* represents the physical device name. You must always include a colon after the device name. The expression *volume_label* is a label that you choose for the disk. For example, if you mount a disk with the physical device name DRA1, and you choose USERFILES as the volume label for this disk, then you would include the following command in the SYSTARTUP_V5.COM file:

```
$ MOUNT DRA1: USERFILES
```

The expression *logical_name*, in the context of the MOUNT command, is a logical volume name that is associated with the volume that you mount. You can use the logical volume name (instead of the physical device name) in programs and procedures that are used on your system, and it is not necessary to know the physical drive on which the volume is mounted.

If you do not specify a logical volume name in the MOUNT command, then the logical volume name is in the form DISK$volume_label. In the previous example, where no logical name was specified and the volume label was USERFILES, the MOUNT command would automatically assign the logical name DISK$USERFILES to the disk.

The following command produces the logical volume name USER and equates it to DRA1, the device name. Note that the logical volume name USER is equivalent to DRA1 only while the disk is actually mounted; if the volume is dismounted, then the logical volume name no longer has any systemwide meaning.

```
$ MOUNT/SYSTEM  DRA1:  USERFILES  USER
```

2.4.2 Setting Device Characteristics

On some systems, certain devices (such as terminals or printers) should have the same characteristics whenever the system is running. Characteristics include defining the device as a printer, setting the transmission speed for terminals, and so on. You can define these characteristics in the SYSTARTUP_V5.COM procedure. Read this section if you want to define certain characteristics for specific devices on your system.

To establish the characteristics of the terminals and other devices on the system, use a series of SET commands in SYSTARTUP_V5.COM. Use the SET TERMINAL command for terminals; you may want to include comments to remind yourself of the user to whom specific terminals may be assigned.

Use the SET PRINTER command for printers. Printer characteristics must be set before you establish queues for the printers.

The following example shows how you could modify SYSTARTUP_V5.COM to set up characteristics for terminals and a printer:

```
$ SET TERMINAL TTC2: /SPEED=300 /DEVICE_TYPE=LA36 /PERMANENT  !JONES
$ SET TERMINAL TTD1: /SPEED=9600 /PERMANENT                   !WRENS
$ SET TERMINAL TTD4: /SPEED=1200 /PERMANENT                   !JRSMITH
$ SET TERMINAL TTG4: /SPEED=1200 /MODEM /PERMANENT            !DIALUP1
$ SET PRINTER /LA11 /PAGE=60 /WIDTH=80  LPA0:
```

For more information about the qualifiers available with the SET TERMINAL and SET PRINTER commands, see the *VMS General User's Manual.*

2.4.3 Printers and Batch Processing: Initializing and Starting Queues

If your system has a printer that you want to make available for general use (that is, a printer that is not connected directly to an individual terminal), you must establish a *print queue.* Similarly, if you want to allow batch processing on your system, you must establish a *batch queue.* A queue allows users to submit requests for printing or batch processing, and the system prints or processes the users' jobs as resources allow.

If you want to include printing or batch processing capabilities on your system, you should include commands in SYSTARTUP_V5.COM that do the following:

1. Start the system job queue manager

2. Initialize and start each queue with a separate INITIALIZE/QUEUE/START command line

The following example shows how to start the system job queue manager and initialize and start queues in SYSTARTUP_V5.COM:

```
$ !
$ !Start the system job queue manager
$ !
$ START/QUEUE/MANAGER
$ !
$ !Set printers spooled and establish printer queues
$ !
$ SET PRINTER/LOWER  LPA0:
$ SET DEVICE/SPOOLED=SYS$PRINT  LPA0:
$ INITIALIZE/QUEUE/START/DEFAULT=FLAG/NOENABLE_GENERIC  LPA0:
$ !
$ SET PRINTER/LOWER  LPB0:
$ SET DEVICE/SPOOLED=SYS$PRINT  LPB0:
$ INITIALIZE/QUEUE/START/DEFAULT=FLAG/NOENABLE_GENERIC  LPB0:
$ !
$ INITIALIZE/QUEUE/START/GENERIC=(LPA0,LPB0) SYS$PRINT
$ !
$ !Establish batch queues
$ !
$ INITIALIZE/QUEUE/START/BATCH/JOB_LIMIT=2/BASE_PRIORITY=3 SYS$BATCH
```

NOTE: DIGITAL recommends using the /RESTART qualifier with the START /QUEUE/MANAGER command. This qualifier causes the queue manager to restart automatically if the job controller should abort.

A *spooled device* directs the output of an application to an intermediate file until the application program finishes. When the application completes, the file is submitted for printing. A spooled device can help balance the workload demand on line printers if you are running applications on a time-shared system. Use the SET DEVICE /SPOOLED command to establish spooled devices.

2.4.4 Installing Known Images

You can *install* commonly used programs as *known images* to reduce the I/O overhead in activating those images and to assign attributes or privileges to the images. If you have programs on your system that meet any of the following conditions, you should read this section and install such programs as known images in the SYSTARTUP_V5.COM startup file:

- Programs that are frequently run
- Programs that are usually run concurrently by several processes
- Programs that require special privileges

All known images must be reinstalled each time the system is rebooted, because known file lists are not saved if the system is shut down or fails. You include INSTALL commands in SYSTARTUP_V5.COM to install programs as known images. Chapter 9 includes a discussion about performance characteristics and known images.

The following example shows a command sequence that might appear in SYSTARTUP_V5.COM for installing additional known images:

```
$ INSTALL
ADD/OPEN/SHARED/HEADER_RESIDENT BLISS32
ADD/OPEN/SHARED MACRO32
ADD/OPEN DIRECTORY
```

2.4.5 Starting Up the DECnet Network

The DECnet software lets your system communicate with other computers. If you install DECnet software on your system, you must include commands in SYSTARTUP_V5.COM that start up the DECnet network. Read this section if you have the DECnet software on your system.

If you have started a batch queue on your system (as described in an earlier section), then you should start the network using the following commands in SYSTARTUP_V5.COM:

```
$ IF F$SEARCH("SYS$SYSTEM:NETACP.EXE") .NES. "" - !This is faster, if you
$ THEN SUBMIT SYS$MANAGER:STARTNET.COM            !have batch queues set up.
```

These commands submit the network startup procedure (SYS$MANAGER:STARTNET.COM) as a batch job, which reduces the amount of time it takes to boot your system. Alternatively, if you have not started a batch queue, use the following command in SYSTARTUP_V5.COM to start up the network:

```
$ IF F$SEARCH("SYS$SYSTEM:NETACP.EXE") .NES. "" THEN @SYS$MANAGER:STARTNET
```

2.4.6 Running the System Dump Analyzer

In the event of a system failure, the System Dump Analyzer (SDA) can help you determine why the system failed. In order to use SDA for this purpose, you should make sure that the system dump file is available for analysis and not overwritten by a new crash. Read the rest of this section if you want to learn about using SDA with SYSTARTUP_V5.COM.

You can start SDA in your site-specific startup file by using the following lines in SYSTARTUP_V5.COM:

```
$ ANALYZE/CRASH_DUMP SYS$SYSTEM:SYSDUMP.DMP
COPY SYS$ERRORLOG:SYSDUMP.DMP  SYSDUMP.BACK
```

For further information, invoke the System Dump Analyzer for an interactive session upon completion of startup, or refer to the SDA documentation in the extended VMS documentation set.

CAUTION: If you use the page file for the crash dump file, when the system reboots, you must enter the SDA command COPY to copy the dump from the page file to another file suitable for analysis. If you fail to perform the copy operation, pages used to save the crash dump information are not released for paging, and your system hangs while executing STARTUP.COM in the rebooting process.

2.4.7 Purging the Operator's Log File

Each time the system is rebooted, a new version of OPERATOR.LOG is created in the SYS$MANAGER directory. You should devise a plan for regular maintenance of these files. Adding the following command to SYSTARTUP_V5.COM purges all but the last two versions of the operator's log file:

```
$ PURGE/KEEP=2 SYS$MANAGER:OPERATOR.LOG
```

See Chapter 10 for additional suggestions for maintaining the operator's log file.

2.4.8 Submitting Batch Jobs That Are Run at Startup Time

Some sites may have batch jobs that are submitted at system startup time. To submit such batch jobs, add SUBMIT commands to your SYSTARTUP_V5 file, in the following format:

```
$ SUBMIT [/qualifier,...] file-spec
```

In the following example, a batch job is submitted to run a command procedure that rebuilds the disks each time the system is initialized.

```
$ SUBMIT  SYS$MANAGER:SYSDISK_REBUILD
```

If you submit batch processing jobs in SYSTARTUP_V5.COM, make sure that the batch processing jobs are submitted after the batch queues have been initialized. See Chapter 5 for more information on submitting batch jobs.

2.4.9 Defining the Number of Interactive Users

You can set a limit for the number of interactive users that are allowed to be logged in to your system at one time. To do this, include the following command in SYSTARTUP_V5.COM:

```
$ STARTUP$INTERACTIVE_LOGINS == n
```

Where *n* is the maximum number of interactive users that are permitted to log in at one time.

NOTE: The number of interactive users must be set to a value no greater than that which is authorized by your VAX processor license.

2.4.10 Starting Up the LAT Network

A LAT network is any local area network where terminal servers and operating systems use the Local Area Transport (LAT) protocol. A LAT network can coexist on the same Ethernet with other protocols. If your system uses a LAT network, you should read this section.

To configure your system as a service node within a LAT network, execute the command procedure SYS$MANAGER:LTLOAD.COM from within SYSTARTUP_V5.COM. LTLOAD.COM starts up the LAT protocol. In the LAT protocol, a VMS operating system advertises its services over the Ethernet and responds to connection requests from terminal servers supporting user terminals and other asynchronous devices.

To start up the LAT network, add the following command line to SYSTARTUP_V5.COM:

```
$ @SYS$MANAGER:LTLOAD
```

To configure a node as a service node that connects only to interactive terminals on a terminal server, include the command line shown in the last example in SYSTARTUP_V5.COM.

However, if you want to use remote printers on a terminal server or to create dedicated application services on the VMS service node, you must modify LTLOAD.COM.

Supporting User Terminals on a Terminal Server

To create a VMS service node on a LAT network that supports only interactive terminals is a one-step procedure. You insert the command @SYS$MANAGER:LTLOAD into SYSTARTUP_V5.COM and append any of the following arguments:

```
$ @SYS$MANAGER:LTLOAD "P1" "P2" "P3" "P4"
```

The arguments P1 through P4 have the following meanings:

Argument	Format	Meaning
P1	Service-name	Name of the VMS service. For clustered VMS service nodes, use the cluster name as the service name. For independent VMS service nodes, use the physical node name.
P2 - P4	Any of the following:	
	/IDENTIFICATION="string"	Description of the node and its services that are advertised over the Ethernet. The default is the string defined by the logical name SYS$ANNOUNCE.
	/ENABLE=group-list	Terminal server groups qualified to establish connections with the VMS service node. By default, Group 0 is enabled.
	/DISABLE=group-list	Removes previously enabled terminal server groups.

The argument P1 assigns a service name to the node, using the LATCP command CREATE SERVICE. Arguments P2 through P4 can be any valid qualifier to the SET NODE command.

For example, the following command creates the service OFFICE on the VMS service node, MOE, which is part of the OFFICE cluster.

```
$ @SYS$MANAGER:LTLOAD OFFICE "/ENABLE=1" "/DISABLE=0"
```

2.4.11 Creating Systemwide Announcements

This section describes how to define the following types of systemwide announcements in your SYSTARTUP_V5.COM file:

- A message to users informing them that the system is available for use (after a system boot)
- A message to users when first accessing the system for login
- A welcoming message when a user logs in

When your system has completed the startup procedure and is up and running, you can send a message to all connected terminals announcing the system's availability. To do this, include a line, with an appropriate message within the quotation marks, before the $EXIT command in your SYSTARTUP_V5.COM file:

```
$ REPLY/ALL/BELL "VMS Operating System at WUZNOT, INC., ready for use."
```

If you want to display a message at the beginning of each user's login procedure, include a line, with an appropriate message within the quotation marks, in SYSTARTUP_V5.COM:

```
$ DEFINE/SYSTEM SYS$ANNOUNCE "WUZNOT, INC. -- Authorized Use Only"
```

You can also display a message to all interactive users immediately after they log in by including a line similar to the following in SYSTARTUP_V5.COM:

```
$ DEFINE/SYSTEM SYS$WELCOME "Welcome to the VMS Operating System at WUZNOT, INC."
```

If you do not define SYS$WELCOME, the following standard message is displayed:

```
Welcome to VMS Version n.n
```

The SYSTARTUP_V5 command file supplied as a template with DIGITAL's distribution kit includes additional command examples for SYS$ANNOUNCE and SYS$WELCOME.

You may also want to display various system announcements to users at the time that they log in. You do this with a command in the systemwide login command procedure, SYLOGIN.COM, as explained later in this chapter.

2.5 Defining a System Login Command Procedure

A system login command procedure is executed for each interactive user when the user logs in. With a system login command procedure, you can establish elements of a computing environment that are the same for all interactive users. To use a system login procedure, you should do as follows:

1. Define the logical name SYS$SYLOGIN, usually in your site-specific startup file (SYSTARTUP_V5.COM).
2. Create a system login command procedure.

To define the logical name SYS$SYLOGIN and point to a system login command file named SYS$MANAGER:SYLOGIN.COM, include the following line in SYSTARTUP_V5.COM:

```
$ DEFINE/SYSTEM/EXEC/NOLOG SYS$SYLOGIN SYS$MANAGER:SYLOGIN.COM
```

A template for a system login command procedure is found in SYS$MANAGER:SYLOGIN.COM. This file includes commands that you can modify and add to according to the needs of your site.

You can use the system login command procedure to display announcements for your site. To do this, you would do the following:

1. Create a text file that has current announcements, for example with the filename SYS$MANAGER:ANNOUNCEMENTS.TXT. You could then update this file (adding and deleting announcements) as needed.
2. Include a line at the end of your system login command procedure that displays the announcements file, such as the following:

   ```
   $ TYPE SYS$MANAGER:ANNOUNCEMENTS.TXT
   ```

In addition to a system login command procedure, users can also have their own login command procedures. User login command procedures are executed immediately after the system login command procedure.

2.6 Backing Up the System

To limit the risk of losing your operating system environment, you should perform the following sequential operations after installing and customizing your system:

1. Back up the console volume
2. Build a standalone backup kit
3. Back up the system disk

If your processor has a console storage device, DIGITAL recommends that you make a backup copy of your console volume; it is useful to have a backup copy in case your original becomes corrupted. The VMS operating system provides a command procedure called CONSCOPY.COM in the SYS$UPDATE directory that copies your console volume to a blank one.

To back up your system disk, DIGITAL recommends that you use standalone BACKUP, which uses a subset of Backup Utility qualifiers. If your system was not distributed on magnetic tape, you must build a standalone BACKUP kit either on console media or on disk. You can then boot standalone BACKUP from the console block storage device or from the alternate directory root SYSE on a Files–11 disk.

Installing and using standalone BACKUP in an alternate root on your system disk saves time when you are backing up your system disk, because you do not have to boot standalone BACKUP from your console volume.

NOTE: The procedures for backing up the console volume and backing up the system disk vary for different VAX processors. See your VAX processor installation and operations guide for the step-by-step procedures that apply to your processor.

2.7 Building and Copying a VMS System Disk

The command procedure SYS$UPDATE:VMSKITBLD is used for building and copying a VMS system disk. The procedure provides you with the following options:

- **BUILD** — Destroys all previous information on the target disk and then builds the new system disk.
- **ADD** — Adds another copy of the operating system to an alternate system root directory on the same system disk.
- **COPY** — Copies the operating system files to a target disk without destroying the files that are currently on the target disk.
- **COMMON** — Initializes the target disk and builds it as a cluster-common system disk.

CAUTION: The VMSKITBLD BUILD and COMMON options initialize the target disk, deleting all of its previous contents.

In some cases, you may want the operating system to exist on another disk. The following paragraphs describe two such cases and the procedures that you would use.

You may want to move your operating system files to another disk. For example, assume that your operating system is initially stored on a disk together with some of your user files. Suppose that you want to move only the operating system files from original disk to a smaller disk. You can build the operating system on the smaller disk (called the target, or destination, disk) without affecting the user files on the original disk (the source disk) by using the BUILD option of the VMSKITBLD command procedure.

You may want to create a separate test environment where you can modify the operating system without affecting current operations. You could use the ADD option to copy the operating system to an alternate system root directory and create a boot command procedure to select that version for testing sessions. In addition, you may want to preserve the current version of the operating system before upgrading your system to the next major version. To do so, use the ADD option to make a copy of the current operating system in an alternate system root directory (SYSA) and then upgrade and run the new version of the operating system in SYS0.

CAUTION: When you copy the system disk using VMSKITBLD.COM, SYSUAF.DAT and all user-modified command files are NOT copied to the target disk. VMSKITBLD.COM uses the site-specific template files with the TEMPLATE file type in building the new system disk.

2.8 System Startup Procedures

This section describes the process that the VMS operating system follows when you boot your system. This section is mostly informational—that is, you usually do not have to do anything during the booting process, but you may want to know how the operating system is set up.

Each time that your system is booted, the VMS operating system initiates a startup procedure. The startup procedure includes the execution of the following series of command procedures:

- **SYS$SYSTEM:STARTUP.COM**—A file containing a series of procedures that must execute at system startup time in order for the system to run properly. STARTUP.COM is the site-independent startup command procedure supplied by DIGITAL. Do not modify this command procedure. The STARTUP.COM procedure invokes the site-specific procedures that are described in this section.
- **SYS$MANAGER:SYCONFIG.COM**—A template file supplied by DIGITAL to which you can add site-specific device configuration commands.

- **SYS$MANAGER:SYLOGICALS.COM**—A template file supplied by DIGITAL for defining logical names. This file contains a command procedure for adding system logical names for a MicroVAX that is not in a cluster. If your processor is not a standalone MicroVAX, you can ignore that section of the procedure that pertains only to MicroVAX systems and add any systemwide logical name assignments for your own system to the end of this file.
- **SYS$MANAGER:SYLOGIN.COM**—A template file supplied by DIGITAL to which you can add commands that are executed whenever a user logs in.
- **SYS$MANAGER:SYSTARTUP_V5.COM**—A template file supplied by DIGITAL to which you can add various commands for setting up site-specific operations that are executed at startup time. The template contains commands that you can modify to meet the needs of your processing environment.
- **SYS$MANAGER:SYPAGSWPFILES.COM**—A file supplied by DIGITAL to which you can add commands to install page and swap files on any disk.

Two versions of the template files are included in your VMS distribution kit: an executable version with the file extension COM, and a nonexecutable version with the file extension TEMPLATE (for example, SYS$MANAGER:SYCONFIG.COM and SYS$MANAGER:SYCONFIG.TEMPLATE). The files with the COM filetype are executed at startup time, and those are the files that you should modify to meet the needs of your site. The files with the TEMPLATE filetype should not be modified.

CAUTION: Do not delete the DIGITAL-supplied template command files with the TEMPLATE file type. The VMSKITBLD.COM procedure uses the TEMPLATE versions to create a new system disk.

More information on STARTUP.COM and the site-specific command procedures is provided in the sections that follow.

2.8.1 Startup Command Procedure for the System (STARTUP.COM)

This section describes the system startup file (STARTUP.COM). STARTUP.COM is executed whenever the system is booted, and it creates the basic environment for the operating system and some software products. It is not a startup file that is customized for your site. You should not modify the STARTUP.COM file. Read this section if you are interested in learning about the startup process.

The file SYSTARTUP_V5.COM, which is also executed each time the system is booted, is the startup file where you include features specific to your site. To learn how to customize the startup process for your site by modifying SYSTARTUP_V5.COM, see Section 2.4.

The file SYS$SYSTEM:STARTUP.COM executes immediately after the operating system is booted. It is a driver that uses a series of component files to perform the following startup tasks:

- Defines systemwide logical names required for the symbolic debugger, language processors, linker, image activator, and help processor.

- Starts processes that control error logging, SMISERVER (the system management server), the job controller, and the operator's log. (On a standalone workstation, the operator's log is not automatically started.)
- Connects and configures devices that are physically attached to the system and loads their I/O drivers by invoking the SYCONFIG.COM procedure.
- Installs known images to reduce I/O overhead in activating the most commonly run images or to identify images that must have special privileges.

CAUTION: Do not modify SYS$SYSTEM:STARTUP.COM. This file is deleted and replaced each time you upgrade your system with the next version of the VMS operating system. Leaving STARTUP.COM intact prevents you from inadvertently altering any commands in the file, which in turn could cause the startup procedure to fail.

All of the component files used by STARTUP.COM are in the directories with the logical name *SYS$STARTUP*. SYS$STARTUP is actually a searchlist that includes both SYS$SYSROOT:[SYSMGR] (the SYS$MANAGER directory) and SYS$SYSROOT:[SYS$STARTUP].

In VMS Version 5.0, the following three data files are involved in the startup process and located in SYS$STARTUP:

1. VMS$PHASES.DAT—This file determines the order of the phases of the startup procedure. It is a sequential list of the phases that will be started by STARTUP.COM. It includes a series of four basic phases (INITIAL, CONFIGURE, SYSFILES, and BASEENVIRON) needed to bring the VMS operating system up to a basic working environment, followed by a series of phases for optional software products. This file must not be modified.
2. VMS$VMS.DAT—This is a component data file for starting the base VMS operating system environment. You should not modify this file.
3. VMS$LAYERED.DAT—This is a component file for optional software products that are installed using the callback procedure of VMSINSTAL. It is an indexed-sequential file, containing the following fields for each file:
    1. Name of the component file (either .EXE or .COM) to be run.
    2. Phase in which the component file is to be run. The valid phases are LPBEGIN, LPMAIN (default), LPBETA, and END.
    3. Method (or mode) by which the component file is to run. The valid choices are DIRECT (the default, where the command procedure or image is executed immediately), BATCH (valid only for command procedures), or SPAWN.
    4. Node restrictions for the component. This is either the node or nodes on which the component file should *only* be run, or the node or nodes on which the component file should *not* be run.

5. Node restriction byte field. This field determines whether the nodes listed in the previous field are allowed or disallowed (for running the component).

6. Parameters passed to the component file for execution. You can pass up to eight parameters, using the following format:

   ```
   (P1:args,P2:args,...)
   ```

   (The parentheses can be omitted if you pass only a single parameter.)

An important function of each phase is to meet the prerequisites of the following phase; therefore, the ordering of the phases is extremely important. Components that occur in a phase cannot have dependencies on components that are in the same phase or in subsequent phases. When installing optional software products as known images using the STARTUP.COM procedure, be sure that all requisite components occur in a previous phase.

If an optional software product can use the callback procedure included in VMSINSTAL, then you can install it as a known image at system startup using the method described earlier in this section, and you do not have to include the product in the site-specific startup file (SYSTARTUP_V5.COM). In these cases, the component files must be in the SYS$STARTUP directory. Software products that do not use the callback procedure should be installed as known images at system startup using SYSTARTUP_V5.COM.

You can also use the System Management Utility (SYSMAN) to manage the new startup process. With the STARTUP command of SYSMAN, you can add, modify, display, or remove elements of existing component files, create a new startup file, and perform other startup functions. A description of SYSMAN commands is found in the Reference section.

Several site-specific command procedures are executed from within STARTUP.COM. You can add commands to these files or modify the template files supplied in your VMS distribution kit. Remember, however, to modify only the executable version of the file (with the file extension COM) and not the template version (with the file extension TEMPLATE). If you have an existing COM file and you want to modify a version of the original TEMPLATE file, then you should first make a copy of the TEMPLATE file (giving the copy a filetype of COM).

STARTUP.COM executes the site-specific command procedures in the following sequence:

1. SYS$MANAGER:SYPAGSWPFILES.COM
2. SYS$MANAGER:SYCONFIG.COM
3. SYS$MANAGER:SYLOGICALS.COM
4. SYS$MANAGER:SYSTARTUP_V5.COM

2.8.2 Setting Up Logical Names for Your Site (SYLOGICALS.COM)

A logical name is a name that is equivalent to a file specification, a directory, a device name, another logical name, or some other equivalence string. For example, when you have a logical name associated with a device name, you can use the logical name instead of the formal device name.

You can assign logical names that apply for the entire system; these are called systemwide logical names, and they can be used by any process on the system. For example, if a systemwide logical name equated the logical name FINANCE_DISK to the device DRA2, any user on the system (and any program running on the system) could use the name FINANCE_DISK in place of DRA2.

The file SYS$MANAGER:SYLOGICALS.COM can be used for assigning systemwide logical names. SYLOGICALS.COM is executed as part of the STARTUP.COM procedure whenever your system is booted. The logical names defined in SYLOGICALS.COM (as well as the logical names assigned automatically in STARTUP.COM) are always included in the system logical name table.

If your system is a MicroVAX that is *not* in a cluster, you should use the file SYLOGICALS.COM as a template for assigning systemwide logical names. If you have a MicroVAX that is not in a VAXcluster environment and you want to have systemwide logical names, you should read this section.

Unless your processor is a MicroVAX that is not in a VAXcluster environment, the template procedure that is found in SYLOGICALS.COM has no effect. However, if your processor is one where the template procedure does not apply, you can still use SYLOGICALS.COM to assign systemwide logical names by adding them to SYLOGICALS.COM before the EXIT command, as indicated in the SYLOGICALS.COM template.

During VMS system operations when the integrity of the system could be compromised by incorrect logical names, such as the activation of privileged images (LOGINOUT, MAIL, and so forth), only executive-mode and kernel-mode logical names are used; supervisor-mode and user-mode names are ignored. DIGITAL therefore recommends that logical names for system components (for example, public disks and directories) be defined in executive mode, for example:

```
$ DEFINE/SYSTEM/EXECUTIVE/NOLOG SYSDSK SYS$SYSDEVICE:
```

See the *VMS General User's Manual* for information about logical name assignments and the privilege modes.

2.9 Emergency Startup Procedures

The startup and login procedures provided by DIGITAL should always work; however, certain user interventions may cause them to fail. For example, if you modify the startup or login procedures, or modify the login accounts, you may accidentally lock yourself out of the system. A very simple way to lock yourself out of the system is to set passwords and forget them. Another way to lock yourself out is to introduce an error condition or an infinite loop into a startup or login procedure. Under such circumstances, use the emergency startup procedure described in this section.

2.9.1 Bypassing the User Authorization File

The preferred method of breaking into a locked system is to set the alternate user authorization file. This method requires setting the system parameter UAFALTERNATE, which defines the logical name SYSUAF to refer to the file SYS$SYSTEM:SYSUAFALT.DAT. If this file is found during a normal login, the system uses it to validate the account and prompts you for the user name and password.

If this file is not located, the system assumes that the UAF is corrupt and accepts any user name and password to log you into the system from the system console. Logins are prohibited from all other locations.

NOTE: You can use this method only to log into the system from the console terminal; you cannot use other terminal lines.

To set the alternate user authorization file, use the following procedure:

1. Perform a conversational boot by following the instructions in your VAX processor installation and operations guide.
2. When the SYSBOOT> prompt appears, enter the following command:

   ```
   SYSBOOT> SET UAFALTERNATE 1
   ```

3. Type CONTINUE and press RETURN.
4. When the startup procedure completes, log in on the console terminal by entering any user name and password in response to the *Username:* and *Password:* prompts.

   The system assigns the following values to your user account:

   - Name—User name
   - UIC—[001,004]
   - Command interpreter—DCL
   - Login flags—None
   - Priority—Value of the system parameter DEFPRI

- Resources—Values of the PQL system parameters
- Privileges—All

The process name is usually the name of the device on which you logged in (for example, _OPA0).

5. Fix the problem that caused you to be locked out of the system. That is, make the necessary repairs to the UAF or to the startup or login procedures. (If you modify a login or startup procedure and the problem is still not solved, you should restore the procedure to its previous state.)

   If the problem is a forgotten password, reset the UAFALTERNATE system parameter to 0, as explained in the next step. Then invoke the Authorize Utility and type HELP MODIFY for information on modifying passwords.

6. Clear the UAFALTERNATE parameter by running SYSGEN and giving appropriate SYSGEN commands. To run SYSGEN, enter the following command at the DCL prompt:

   ```
   $ RUN SYS$SYSTEM:SYSGEN
   ```

   The SYSGEN> prompt is displayed, and you should enter the following commands:

   ```
   SYSGEN> SET UAFALTERNATE 0
   SYSGEN> WRITE CURRENT
   SYSGEN> EXIT
   ```

7. Shut down and reboot the system.

2.9.2 Emergency Startup After Modifying System Parameters

In some cases, modifying system parameters may cause the system to become unbootable. If this occurs, use the following emergency startup procedure to restore normal operation:

1. Perform a conversational boot by following the instructions in your VAX processor installation and operations guide.
2. When the SYSBOOT> prompt appears, enter the following commands:

   ```
   SYSBOOT> USE DEFAULT.PAR
   SYSBOOT> CONTINUE
   ```

3. When the system finishes booting, review any changes you made to SYSGEN parameters, modify MODPARAMS.DAT as necessary, and reexecute AUTOGEN.

2.9.3 Bypassing Startup and Login

If the system does not complete the startup procedures or does not allow you to log in, bypass the startup and login procedures by following these steps:

1. Perform a conversational boot operation by following the instructions in your VAX processor installation and operations guide.
2. Define the console to be the startup procedure by entering the following command at the SYSBOOT> prompt:

   ```
   SYSBOOT> SET/STARTUP OPA0:
   ```

   Type CONTINUE and press RETURN in response to the next SYSBOOT> prompt. Wait for the DCL prompt to return.
3. Correct the error condition that caused the login failure. That is, make the necessary repairs to the startup or login procedures, or to the UAF. You may want to enter the following DCL commands because bypassing the startup procedures leaves the system in a partially initialized state:

   ```
   $ SET NOON
   $ SET DEFAULT SYS$SYSROOT:[SYSEXE]
   ```

   Invoke a text editor to correct the startup or login procedure file. Note that some system consoles may not supply a screen-mode editor. You can also copy a corrected file and delete the incorrect version by using the RENAME and DELETE commands.
4. Reset the startup procedure by invoking SYSGEN and entering the following commands:

   ```
   $ RUN SYS$SYSTEM:SYSGEN
   SYSGEN> SET/STARTUP SYS$SYSTEM:STARTUP.COM
   SYSGEN> WRITE CURRENT
   SYSGEN> EXIT
   ```
5. Perform a normal startup by entering the following command:

   ```
   $ @SYS$SYSTEM:STARTUP
   ```

2.9.4 Startup Problems

Sometimes the operating system does not boot after you enter the BOOT command. This can be caused by either a hardware or software malfunction.

A read error on a disk drive or console medium, or a machine check error, may indicate a hardware malfunction. When a hardware problem occurs, a question mark (?) usually precedes the error message that is displayed on the system console terminal. You should then do the following:

1. Consult the hardware manual for your VAX processor.

2. If you still cannot correct the problem, contact your DIGITAL Field Service representative.

When the operating system is loaded into memory but the STARTUP.COM command procedure does not execute, a software malfunction has probably occurred. You should suspect this condition if the usual message specifying the number of interactive users does not appear.

Perform one or both of the following actions to correct the situation:

- Try again, by repeating the boot procedure to restart the system (see the installation guide for your VAX processor).
- Leave the system disk in the original drive. Restore a backup copy of the system disk using Standalone Backup.

2.10 Shutdown Procedures

The VMS operating system provides the following types of shutdown procedures:

- **An orderly shutdown with SYS$SYSTEM:SHUTDOWN.COM.** This procedure shuts down the system while performing housekeeping functions such as disabling future logins, stopping the batch and output queues, dismounting mounted volumes, and stopping user processes.

  SHUTDOWN.COM optionally invokes a site-specific command procedure named SYS$MANAGER:SYSHUTDWN.COM, which you can modify to meet the needs of your specific installation. An empty SYSHUTDWN.COM file is included in your VMS distribution kit.

- **An emergency shutdown with OPCCRASH.** If you are unable to perform an orderly shutdown with SHUTDOWN.COM, run the OPCCRASH emergency shutdown program.

- **An emergency shutdown with CRASH.** Use this emergency shutdown procedure if OPCCRASH fails. Note that not all VAX processors have the CRASH emergency shutdown program. If your VAX processor has the CRASH procedure, it is located on the console media, and it can only be executed from the console terminal. See your VAX processor installation and operations guide for a description of the CRASH procedure or for the equivalent commands to use to force an abrupt emergency shutdown.

2.10.1 Orderly Shutdown with SHUTDOWN.COM

Use SHUTDOWN.COM to shut down the system in an orderly fashion. Do not modify SHUTDOWN.COM. Add commands to the SYS$MANAGER:SYSHUTDWN.COM command procedure to perform site-specific functions.

To execute SHUTDOWN.COM, you must have either the SETPRV privilege or all the following privileges: CMKRNL, EXQUOTA, LOG_IO, OPER, SYSNAM, SYSPRV, TMPMBX, and WORLD. Usually, you shut down the system from the SYSTEM account, which includes these privileges by default.

2.10.1.1 SHUTDOWN.COM Sequence of Prompts and Messages

To perform an orderly shutdown of the system, invoke SHUTDOWN.COM from any terminal and any privileged account with the following DCL command:

```
$ @SYS$SYSTEM:SHUTDOWN
```

The procedure then prompts you with a series of questions and messages. The default responses appear in brackets at the end of each question. Press the RETURN key to select the default response. A summary of the questions follows:

- Minutes until shutdown:

  ```
  How many minutes until final shutdown [0]?
  ```

  Enter an integer. If the system logical name SHUTDOWN$MINIMUM_MINUTES is defined, its integer value is the minimum value that you can enter. Therefore, if the logical name is defined as 10, you must specify at least 10 minutes to final shutdown or an error message is returned. If you do not enter a value, the logical name value is used. If the logical name is not defined, and you do not enter a value, 0 minutes is the default.

- Reason for shutdown:

  ```
  Reason for shutdown [standalone]:
  ```

  Enter a one-line reason for shutting down the system.

- Decide if you want to spin down the disk volumes:

  ```
  Do you want to spin down the disk volumes [No]?
  ```

  Enter YES or NO (Y or N). Note, however, that the system disk cannot be spun down.

- Decide if you want to invoke a site-specific shutdown procedure:

  ```
  Do you want to invoke the site-specific shutdown procedure [Yes]?
  ```

  Enter YES or NO. This refers to SYS$MANAGER:SYSHUTDWN.COM.

- Decide if you want an automatic system reboot:

  ```
  Should an automatic system reboot be performed [No]?
  ```

By default, the system does not automatically reboot. However, if you respond with YES, the system attempts to reboot automatically when the shutdown is complete.

- Message broadcast to users about rebooting the system:

```
When will the system be rebooted [later]?
```

Enter the expected reboot time in the format you want printed in the message that will be broadcast to users. For example, you could specify IMMEDIATELY, or IN 10 MINUTES, or a time such as 2 P.M. or 14:00. If you do not know when the system will be available again, press RETURN to specify "later" as the time when the system will reboot.

- Shutdown options:

```
Shutdown options (enter as a comma-separated list):
 SAVE_FEEDBACK       Saves feedback data for AUTOGEN calculations
 REMOVE_NODE         Remaining nodes in the cluster should adjust quorum
 CLUSTER_SHUTDOWN    Entire cluster is shutting down
 REBOOT_CHECK        Check existence of basic system files
Shutdown options [NONE]
```

The procedure prompts you to specify one or more shutdown options.

Entering the SAVE_FEEDBACK option records feedback data collected from the system since it was last booted. This option creates a new version of the AUTOGEN feedback data file, which can be used when you next run AUTOGEN.

If your system is a cluster member, all options are listed. When the REMOVE_NODE option is specified on one cluster member system, users on all member systems are notified. Clusterwide notification is required, because users logged in to any member system may be affected by the shutdown of another system, for example:

- Users may have batch jobs running on other systems.
- If terminal servers are in operation, users may have alternate terminal sessions in progress on the system being shut down.

Otherwise, only the REBOOT_CHECK and SAVE_FEEDBACK options are listed. Enter REBOOT_CHECK to verify the presence of a subset of files necessary to reboot the system after shutdown completes. (If you are certain that the files exist, press RETURN.)

If you select the REBOOT_CHECK option, the procedure checks for the necessary files and notifies you if any are missing. Replace missing files before proceeding. If all files are present, the following success message appears:

```
%SHUTDOWN-I-CHECKOK, Basic reboot consistency check completed.
```

The following events occur as the shutdown procedure continues, and the corresponding messages are printed on the terminal:

1. A message requesting users to log out is broadcast at decreasing time intervals to all users on the system.
2. The system logical name SHUTDOWN$TIME is defined as the absolute time of shutdown. For example, if the value 10 is specified at 12:00 in response to the first question, the logical name is assigned the absolute time value 12:10 along with the date. By requesting the logical name definition for SHUTDOWN$TIME (with the SHOW LOGICAL command), you can see if a shutdown is in progress or determine the actual time of shutdown. This feature is useful if a user missed the shutdown broadcast message.
3. At six minutes or less until system shutdown, the terminal from which shutdown was invoked is made an operator's console and all future nonoperator logins are disabled. If the DECnet network is running, it is shut down.
4. When there is one minute left until system shutdown, batch and device queues and the system job queue manager are stopped.
5. At zero minutes before system shutdown, the site-specific command procedure SYS$MANAGER:SYSHUTDWN.COM is invoked.
6. All user processes are stopped; however, system processes continue. Ancillary Control Processes (ACPs) may delete themselves when their mounted volumes are finally dismounted.
7. For dual-processor systems, the secondary processor is stopped.
8. All installed images are removed.
9. All mounted volumes are dismounted and, if you request it, the disks are spun down. Note, however, that the system disk cannot be spun down. Also, the quorum disk (if present on your system) is not dismounted or spun down.
10. The operator's log file is closed.
11. The program SYS$SYSTEM:OPCCRASH is invoked to shut down the system.
12. If you did not request an automatic reboot, the following message appears on the system console:

    ```
    SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
    ```

    If you requested an automatic reboot, the system reboots, provided the necessary controls are set.
13. If you are not automatically rebooting, halt the system after the previous message is printed at the console terminal.

Example 2–1 demonstrates an orderly system shutdown on standalone node AVALON.

Example 2–1: Orderly System Shutdown with SHUTDOWN.COM

```
$ @SYS$SYSTEM:SHUTDOWN

        SHUTDOWN -- Perform an Orderly System Shutdown

How many minutes until final shutdown [0]: 10
Reason for shutdown: [Standalone] MONTHLY PREVENTIVE MAINTENANCE.
Do you want to spin down the disk volumes [No]? RET
Do you want to invoke the site-specific shutdown procedure [Yes]? RET
Should an automatic system reboot be performed [No]? RET
When will the system be rebooted [later]? 12:30
Shutdown options:
 REBOOT_CHECK        Check existence of basic system files
Shutdown options [NONE] RET

SHUTDOWN message on AVALON, from user SYSTEM at _AVALON$OPA0:   12:00:00.20
AVALON will shut down in 10 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE

%SHUTDOWN-I-OPERATOR, This terminal is now an operator's console.
%%%%%%%%%%  OPCOM, 16-JUL-1988 12:01:00.15  %%%%%%%%%%
Operator status for operator _AVALON$OPA0:
CENTRAL, PRINTER, TAPES, DISKS, DEVICES, CARDS, NETWORK, OPER1, OPER2,
OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9, OPER10, OPER11,
OPER12

%SHUTDOWN-I-DISLOGINS, Interactive logins will now be disabled.
%SET-I-INTSET, login interactive limit = 0 current interactive value = 17
%SHUTDOWN-I-SHUTNET, The DECnet network will now be shut down.
%SHUTDOWN-I-STOPQUEMAN, The queue manager will now be stopped.


SHUTDOWN message on AVALON, from user SYSTEM at _AVALON$OPA0:   12:05:00.20
AVALON will shut down in 5 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE

17 terminals have been notified on AVALON.

SHUTDOWN message on AVALON from user SYSTEM at _AVALON$OPA0:   12:06:55.28
AVALON will shut down in 4 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE
```

Example 2–1 Cont'd. on next page

Example 2–1 (Cont.): Orderly System Shutdown with SHUTDOWN.COM

```
%%%%%%%%%%  OPCOM, 16-JUL-1988 12:07:12.30  %%%%%%%%%%
Message from user DECnet on AVALON
DECnet event 2.0, local node state change
From node 2.161 (AVALON), 16-JUL-1988 12:07:22.26
Operator command, Old state = On, New state = Shut

SHUTDOWN message on AVALON from user SYSTEM at _AVALON$OPA0:   12:07:12.56
AVALON will shut down in 3 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE

%SHUTDOWN-I-STOPQUEMAN, The queue manager will now be stopped.
SHUTDOWN message on AVALON user SYSTEM at _AVALON$OPA0:   12:08:12.56
AVALON will shut down in 2 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE

%%%%%%%%%%  OPCOM, 16-JUL-1988 12:08:12:30  %%%%%%%%%%
Message from user JOB_CONTROL on AVALON
-SYSTEM-S-NORMAL, normal successful completion

%%%%%%%%%%  OPCOM, 16-JUL-1988 12:08:42.30  %%%%%%%%%%
Message from user DECNET on AVALON
DECnet shutting down

SHUTDOWN message on AVALON from user SYSTEM at _AVALON$OPA0:   12:09:12.56
AVALON will shut down in 1 minute; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE


17 terminals have been notified on AVALON
%SHUTDOWN-I-SITESHUT, The site-specific shutdown procedure will now be invoked.
%SHUTDOWN-I-STOPUSER, All user processes will now be stopped.
%SHUTDOWN-I-REMOVE, All installed images will now be removed.
%SHUTDOWN-I-DISMOUNT, All volumes will now be dismounted.
%%%%%%%%%%  OPCOM, 16-JUL-1988 12:09:42.30  %%%%%%%%%%
Message from user System on AVALON
_AVALON$OPA0:, AVALON shutdown was requested by the operator.

%%%%%%%%%%  OPCOM, 16-JUL-1988 12:10:02.44  %%%%%%%%%%
Logfile was closed by operator _AVALON$OPA0:
Logfile was SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;8

%%%%%%%%%%  OPCOM, 16-JUL-1988 12:10:32.20  %%%%%%%%%%
Operator _AVALON$OPA0: has been disabled, username SYSTEM

       SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
```

2.10.1.2 Defining SHUTDOWN$INFORM_NODES

Before you execute SYS$SYSTEM:SHUTDOWN.COM, you can define the logical name SHUTDOWN$INFORM_NODES to be a list of cluster member nodes. The nodes listed in SHUTDOWN$INFORM_NODES will be notified when the system is shutdown, as shown in the following example:

```
$ DEFINE SHUTDOWN$INFORM_NODES "NODE1,NODE2,NODE3"
$  @SYS$SYSTEM:SHUTDOWN
```

If you define SHUTDOWN$INFORM_NODES and then execute SYS$SYSTEM:SHUTDOWN.COM, all cluster member nodes included in the list are notified of the shutdown. Users on the node that is being shut down are always notified regardless of whether you define SHUTDOWN$INFORM_NODES. If

you omit the name of the node that is being shut down from the list specified in the DEFINE command, the name is automatically added to the list by the SHUTDOWN.COM procedure.

2.10.2 Emergency Shutdown with OPCCRASH

This section describes how to halt the system immediately without performing any of the housekeeping functions that ensure an orderly shutdown. Usually, you shut down the system using the orderly shutdown procedure. You use the OPCCRASH procedure only if SHUTDOWN.COM fails. OPCCRASH performs only the following minimal housekeeping functions:

- Marks the system disk for dismount and empties all file system data caches.
- Writes the modified page list back to the disk. This ensures that all writable section files are updated to their correct state before the system crashes and all in-memory data is lost.

To perform this procedure, you must have the CMKRNL privilege. You can enter the commands from any terminal.

1. Enter the following command to force an immediate shutdown of the system:

   ```
   $ RUN SYS$SYSTEM:OPCCRASH
   ```

2. If the system fails to respond after a few minutes, use the CRASH procedure or, if your system does not have a CRASH procedure, enter the emergency shutdown commands described in your VAX processor installation and operations guide.

3. At the system console, the following message is displayed:

   ```
   SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
   ```

4. Halt the system.

Example 2–2 illustrates an emergency shutdown using the OPCCRASH procedure.

Example 2–2: Emergency Shutdown Using OPCCRASH

```
$ RUN SYS$SYSTEM:OPCCRASH

     SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM

CTRL/P
>>>HALT

     HALTED AT 8000708A
```

Chapter 3

Installing Software

Any software product that you use on your system must be *installed.* When you install software, the software is made available every time that the system starts up. You must use an installation procedure for software when it is first acquired, and you must also use the installation procedure for any subsequent upgrades to the software.

On the VMS operating system, you use the command procedure SYS$UPDATE:VMSINSTAL.COM to install most software products and any upgrades. Use VMSINSTAL.COM to install the most recent version of the VMS operating system and any subsequent maintenance updates, and to install many optional software products.

This chapter describes the following:

- How to prepare your system for running VMSINSTAL
- How to start VMSINSTAL
- How to select appropriate VMSINSTAL options
- What to do if the system fails while running VMSINSTAL

This chapter does not describe VMSINSTAL procedures that are specific to any upgrade, update, or product. The examples used are for illustration only. For details of a particular product, refer to the VMSINSTAL procedure described in the installation documentation for the specific product or update.

3.1 Preparing Your System for VMSINSTAL

Before you use VMSINSTAL, do the following:

1. Back up your system disk. Use the backup copy as a working copy for the installation.

   If you back up your system disk to magnetic tape, you must restore the tape to a Files–11 disk to get a working copy. The working copy has more contiguous space than the original system disk because of the way BACKUP creates the copy. You might need this additional contiguous space during the installation.

   If the system fails during installation, VMSINSTAL might delete the older version of the product before it installs the newer version. You might have to make a new working copy of the system disk and restart the installation.

2. Log in at the console terminal under the SYSTEM account. (If SYSGEN parameters MOUNTMSG or DISMOUMSG are set to 1, OPCOM displays a message each time a disk or tape is mounted or dismounted. If these messages are not disabled, and you are installing from an operator's terminal, they appear within 30 seconds of each mount or dismount.)

3. Be sure that all users are logged out and that all batch jobs are completed.

4. Keep users off the system using the following command:

   ```
   $ SET LOGINS/INTERACTIVE=0
   ```

5. If your system includes it, you should shut down DECnet–VAX if DECnet–VAX would be affected by the software product that you are installing. To shut down DECnet–VAX, do the following:

   a. Start the Network Control Program (NCP):

      ```
      $ RUN SYS$SYSTEM:NCP
      ```

   b. At the NCP prompt (NCP>), enter the following command and press RETURN:

      ```
      NCP> SET EXECUTOR STATE SHUT
      ```

      DECnet–VAX performs an orderly shutdown. The OPCOM facility notifies you when DECnet–VAX is shut down.

   c. At the NCP prompt, enter EXIT and press RETURN.

6. Make sure the limits in the SYSTEM account authorization record are equal to or greater than the following:

```
Buffered byte count quota (BYTLM) = 20480
Enqueue quota (ENQLM) = 200
Direct I/O limit (DIOLM) = 18
Buffered I/O limit (BIOLM) = 18
Open file quota (FILLM) = 40
AST limit (ASTLM) = 24
```

a. To check these limits, run the Authorize Utility (AUTHORIZE). To run AUTHORIZE, enter the following commands and press RETURN after each one:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
```

b. At the UAF prompt (UAF>), enter the following command and press RETURN:

```
UAF> SHOW SYSTEM
```

AUTHORIZE displays the current limits of the SYSTEM account's user authorization file.

7. If you need to modify the current limits, enter the AUTHORIZE command MODIFY in the following format and press RETURN:

```
UAF> MODIFY SYSTEM/limit=new_value
```

For example, to increase the DIOLM limit to 18, enter the following command and press RETURN:

```
UAF> MODIFY SYSTEM/DIOLM=18
```

8. When you have changed the limits, at the UAF prompt (UAF>) enter EXIT and press RETURN. This returns you to the dollar-sign prompt ($). Your changes will not take effect until you log out and log in again. (See the Reference section for a description of the commands in the Authorize Utility.)

9. Physically mount the first distribution medium that contains the software product.

3.1.1 Starting VMSINSTAL

This section tells you how to start VMSINSTAL and describes options that you can use.

When you start VMSINSTAL, it displays several prompts and messages that direct and explain the installation. These prompts and messages differ, depending on the software product that you are installing. Before you begin, make sure that you read and understand the installation procedures for the specific product or update. If you need assistance during an installation, entering a question mark (?) at a prompt gives you an explanation of acceptable responses.

When you start VMSINSTAL, it asks if you are satisfied with the backup of your system disk. Make sure that the backup copy is fairly recent before you continue.

If you have not satisfied all conditions required to start VMSINSTAL, it displays a warning message explaining the problem and a prompt asking if you want to continue. DIGITAL strongly recommends that you correct these conditions before you try to start VMSINSTAL again.

NOTE: If you continue without making the required corrections, DIGITAL might not support the resulting installation.

To start VMSINSTAL, enter a command in the following format and press RETURN:

```
$ @SYS$UPDATE:VMSINSTAL product-list source: [OPTIONS option-list] -
_$ [destination] [qualifiers]
```

The following sections describe the parameters and the keyword that you must supply in this command.

3.1.1.1 Selecting a Product-List

This parameter lists the products that you want to install. If you use a wildcard character (*), all versions and updates of all products from the distribution media will be installed in alphabetical order.

If you want to specify more than one item in the product list parameter, you must separate the items using commas and no intervening spaces. Specify the product list in the following format:

facvvu

fac	The product name code (1 to 36 alphanumeric characters)
vv	The major version number (2 digits)
u	The update number (1 digit)

For example, if you upgrade your operating system to VMS Version 5.2, the product name (fac) is VMS, the major version (vv) is 05, and the update number (u) is 2. Therefore the product-list parameter is VMS052.

Using this format, you can install a specific version and update of a product from distribution media containing several versions and updates. If you do not include a version or update number, all versions and updates from the source are installed in alphabetical order.

If you omit the product-list parameter, VMSINSTAL asks you for it.

If you are installing from a distribution kit, the list of products on your distribution media is included with the bill of materials for the distribution kit. If the list is not available, enter a DIRECTORY command so that the distribution media will find the products that are included. To obtain the product list, enter commands in the following format and press RETURN after each one:

```
$ MOUNT/OV=ID ddcu:
$ DIRECTORY ddcu:[0,0]
```

where *ddcu* is the drive that holds the distribution media.

If you are installing from a disk directory, obtain the product list by entering a DIRECTORY command, specifying the disk directory in the following format and pressing RETURN.

```
$ DIRECTORY node::drive:[directory]
```

If you are accessing a remote node, you must have READ and EXECUTE (R,E) access to the directory.

3.1.1.2 Selecting the Source

This parameter identifies the source of the optional software product as one of the following:

- A drive that holds the distribution media; for example, the RX50 drive designated CSA2: on a VAX 8200.
- A disk directory to which the product save set has been transferred from the distribution media for later installation.

  You specify a disk directory as the source when you select the GET SAVE SET option. For more information about this option, see Section 3.1.3.
- A disk directory on another node.

You also can use a logical name to specify the source. If you do not specify the source, VMSINSTAL asks you for it.

3.1.1.3 Selecting Options

This keyword is optional. It precedes the list of options to be installed. If you enter an option list without the OPTIONS keyword, VMSINSTAL displays an informational error message and the installation ends.

The options-list parameter lists the options requested. The VMSINSTAL command procedure permits the use of five options. You specify each option by entering the appropriate option letter after the keyword OPTIONS in the command that starts VMSINSTAL.

If you specify more than one option, do not separate the letters with spaces or commas. For more information on choosing VMSINSTAL options, see Section 3.1.3.

3.1.1.4 Selecting the Destination

This parameter is *optional*. By default, VMSINSTAL assumes that the product is to be installed in the system-specific directory root on the system disk.

There are two instances in which you must use this parameter:

- If you want to install the product in an alternate root. The product is installed on a system root other than that on which the target system is running. Specify the alternate system root in the following format:

  ddcu:[SYSn.]

ddcu	The device on which the alternate root resides.
SYSn.	The top-level directory of the alternate system root.

  You also may specify a previously defined logical name for the alternate root.

- If you specify the GET SAVE SET option to copy the product kit save sets into a storage directory for later installation. For more information on the GET SAVE SET option, see Section 3.1.3.

  Specify the destination directory in the following format:

  [node::]ddcu:[directory]

node	A node remote from the target system. (DECnet software must be installed on your system to use this option.) If you do not specify a node, VMSINSTAL assumes that the product save sets are to be stored on the local node.
ddcu	The destination disk device.
directory	Usually, a directory dedicated to product save sets on the specified disk.

3.1.1.5 Qualifying the BACKUP Command

You can specify this parameter along with the GET SAVE SET option to qualify the BACKUP command further. The BACKUP command copies the save sets to the destination directory.

The following example includes the GET SAVE SET option and BACKUP qualifiers:

```
$ @SYS$UPDATE:VMSINSTAL TEST042 DUA0:[KITS] OPTIONS G DUB0:[KITS] -
_$ "/VERIFY/LOG/CONFIRM"
```

3.1.2 When the Installation Is Complete

When the installation is complete, VMSINSTAL does one of the following, depending on the requirements of the product you have installed:

- Performs an automatic shutdown of the system and instructs you to reboot manually
- Returns you to the dollar-sign prompt ($)

When the product is installed, back up the updated system disk.

3.1.3 Choosing VMSINSTAL Options

The VMSINSTAL command procedure permits the use of five options:

- Auto-answer option (A)
- Get save set option (G)
- File log option (L)
- Release notes option (N)
- Alternate root option (R)

To specify each option, do the following:

- Enter the appropriate option letter after the keyword OPTIONS in the command line that starts VMSINSTAL.
- Press RETURN.

If you specify more than one option, do not separate the letters with commas or spaces. For example:

```
$ @VMSINSTAL product source: OPTIONS AN
```

The following sections describe the VMSINSTAL options.

3.1.3.1 Auto-answer (A)

The AUTO-ANSWER option makes it easier to *reinstall* a product by providing responses to VMSINSTAL questions and prompts during the reinstallation. You use the AUTO-ANSWER option most often to reinstall products after an upgrade.

If you specify the AUTO-ANSWER option when you install a product, an answer file is created in the form product.ANS in the SYS$UPDATE directory, where *product* is the product-list parameter that you provide when you start VMSINSTAL.

The file type of an answer file is ANS. The answer file contains a record of your responses to questions and prompts from VMSINSTAL. For example, if you install the product, NEWAID, with the AUTO-ANSWER option, VMSINSTAL creates an answer file designated NEWAID.ANS.

When you reinstall the product and specify the AUTO-ANSWER option (typically after upgrading your operating system), VMSINSTAL reads the answer file instead of asking you questions.

If you want to create a new answer file when you reinstall a product, you must delete the existing answer file first.

3.1.3.2 Get save set (G)

Installing products either from a distribution tape or from console media directly onto your system disk is time-consuming. The GET SAVE SET option saves you time by allowing you to store product save sets temporarily on a magnetic tape or in a disk directory.

You might consider dedicating a user disk on a node that other licensed system users can access. You can store product save sets on this dedicated user disk to give other licensed system users fast access to the product save-set directory.

To store a product save set on a disk directory using the GET SAVE SET option, enter a command in the following format and press RETURN:

```
$ @SYS$UPDATE:VMSINSTAL product-list source:-
$ _OPTIONS G device:[directory]
```

NOTE: The directory that you specify must exist, and the device must be mounted; VMSINSTAL does not perform any CREATE/DIRECTORY or MOUNT operations.

You specify the disk directory immediately after the OPTIONS G parameter. For example, if you are storing save sets for a product named NEWAID from the console drive into disk directory USER1:[PRODUCTS], enter the following command and press RETURN:

```
$ @SYS$UPDATE:VMSINSTAL NEWAID CSA1: OPTIONS G -
$_USER1:[PRODUCTS]
```

VMSINSTAL creates one or more files to store the product save set in the disk directory. The save set file name has the following format:

facvvu.x

fac	The product name code (1 to 36 alphanumeric characters).
vv	The major version number (2 digits).
u	The update number (1 digit).
x	A literal extension that is used to identify save set files, where A is the first save set, B the second, and so forth.

NOTE: Valid extensions for save set files include the literal range A through Z and the numeric range VMI_0027 through VMI_9999.

For example, suppose you are storing update 1 to Version 2.0 of the product, NEWAID, and this process requires four save sets. VMSINSTAL creates the following four files:

NEWAID021.A
NEWAID021.B
NEWAID021.C
NEWAID021.D

When you want to install the product on your system, enter a command in the following format and press RETURN:

```
$ @SYS$UPDATE:VMSINSTAL product-list device:[directory]
```

For the product NEWAID, enter this command and press RETURN:

```
$ @SYS$UPDATE:VMSINSTAL NEWAID USER1:[PRODUCTS]
```

VMSINSTAL installs the NEWAID product on your system disk.

3.1.3.3 File Log (L)

The FILE LOG option logs all file activity to the terminal during installation. File activity is defined as any action that alters the disposition of a file, such as creating a new file, updating a library, or deleting a file.

3.1.4 Release Notes (N)

Use the RELEASE NOTES option to display or print the online release notes file supplied by the layered product.

NOTE: Not all layered products provide online release notes.

The person who builds the product kit names the release notes file. The release notes file is given the file name *facvvu.release_notes*, where *facvvu* represents the product name code, version, and update numbers.

If release notes are available and you specify option N, VMSINSTAL asks you the following questions. (The default answers are indicated in brackets.)

```
Release Notes Options:
    1. Display release notes
    2. Print release notes
    3. Both 1 and 2
    4. Copy release notes to SYS$HELP
    5. Do not display, print, or copy release notes

*Select option [2]:

*Queue name [SYS$PRINT]:

*Do you want to continue the installation [N]:
```

- The first prompt (*Select option:) allows you to choose options 1 through 5.
- The second prompt (*Queue name:) is displayed only if you select option 2 or option 3. If you enter the name of a print queue, the system displays a message saying that the release notes have been queued successfully to the printer. If you do not specify a print queue, the release notes are sent to SYS$PRINT by default.
- The third prompt (*Do you want to continue the installation:) allows you either to continue or to end the installation. The default is to end the installation.

If release notes are not supplied with the product, VMSINSTAL displays two error messages. It also asks whether you want to continue or to end the installation, as follows:

```
%VMSINSTAL-W-NOFILE, New File facvvu.RELEASE_NOTES does not exist.
%VMSINSTAL-W-NORELNOTE, unable to locate release notes.

*Do you want to continue the installation [N]:
```

To continue the installation (whether or not release notes are available), type Y (for YES) and press RETURN.

3.1.4.1 Alternate Root (R)

The ALTERNATE ROOT option lets you install the product to a system root other than that of the running system. The VMS operating system in the alternate root must be complete and the same version/update level as the running system. All files and software products that the product installation refers to must be present in the alternate root.

NOTE: Not all optional software products allow a product to be installed to an alternate system root.

Consult the documentation of the specific product to determine whether the product can be installed to an alternate system root.

If you specify option R, the product is installed on the alternate root. However, you cannot create accounts or request a system reboot.

3.1.5 Recovering from a System Failure

If the system fails during installation of an update or optional software product, VMSINSTAL attempts to continue the installation upon rebooting. Depending on when the system failed, one of three conditions exists:

- The system disk was not altered before the system failure. In this case, VMSINSTAL instructs you to restart the installation.
- The system disk or a library used by the installation was corrupted. In this case, VMSINSTAL instructs you to restore either the system disk or the corrupted library from the backup copy and to restart the installation.
- VMSINSTAL continues the installation. In this case, VMSINSTAL performs most of the installation. Also, it may tell you that you must perform some tasks manually to complete the installation. If VMSINSTAL instructs you to do so, do the following:

  a. Reboot the system

  b. Log in as system manager

c. Purge all system files that have been replaced, even if you requested an automatic purge.

```
$ PURGE/LOG SYS$SYSROOT:[*...]*.*
```

SYSTEM MANAGER'S REFERENCE

Chapter 4

Managing Users

As a system manager, it is your job to create and maintain user accounts on the system. To create accounts for users and effectively manage the use of the system, you must determine which users need access to the system and what system resources they require.

Once you understand user needs, you can establish controls that customize the system appropriately.

The VMS operating system provides the Authorize Utility (AUTHORIZE) to authorize and control the use of system resources by individual users. This chapter describes the use of AUTHORIZE to do the following:

- Add a user account
- Modify a user account
- Remove a user account
- List the user accounts

See the Authorize Utility chapter in the Reference section for more information on AUTHORIZE.

4.1 The User Authorization File (UAF)

You manage VMS users by creating and maintaining user accounts, which control who can log in to the system and how it can be used. Use the Authorize Utility (AUTHORIZE) to do the following:

- Create new records and modify existing records in the system user authorization file (SYS$SYSTEM:SYSUAF.DAT) and the network user authorization file (SYS$SYSTEM:NETPROXY.DAT)
- Create new records and modify existing records in the rights database file (SYS$SYSTEM:RIGHTSLIST.DAT)

Whenever a user logs in, the system uses the information contained in the user authorization file (UAF) to validate the login attempt, establish the account's environment, and create a process with appropriate attributes. In this way, the system restricts users to the resources you assign to each account.

As system manager, you may want to create a private copy of SYSUAF.DAT in a directory other than SYS$SYSTEM as an emergency backup for the system SYSUAF.DAT file. Note that, to have an effect on user processes, any private version of SYSUAF.DAT must be copied to the SYS$SYSTEM directory and have the system user identification code (UIC).

Because certain images (such as MAIL and SET) require access to the system UAF and are normally installed with the SYSPRV privilege, make certain that you always grant system access to SYSUAF.DAT. The authorization files are created with the following default protection:

```
SYSUAF.DAT       S:RWED, O:RWED, G, W
NETPROXY.DAT     S:RWED, O:RWED, G:RWED, W
RIGHTSLIST.DAT   S:RWED, O:RWED, G:RWE, W:R
```

If you need to maximize the protection for SYSUAF.DAT or NETPROXY.DAT, use the following DCL command (note, however, that RIGHTSLIST.DAT must be world-readable):

```
$ SET PROTECTION=(S:RWED,O,G,W) SYS$SYSTEM:filename
```

Using the Authorize Utility, you create and maintain UAF records by assigning values to various *fields* within each record. The values you assign identify the user, define the user's work environment, and control use of system resources. Example 4–1 presents a typical UAF record for a nonprivileged user account.

To gain access to a specific user record, set the default directory to SYS$SYSTEM, enter the command RUN AUTHORIZE to invoke the Authorize Utility, and enter the command SHOW *username* at the UAF> prompt. You can then enter AUTHORIZE commands such as ADD and MODIFY to create new user accounts or change the information in the fields of an existing UAF account.

Example 4–1: Sample UAF Record Display

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> SHOW WELCH

Username: WELCH                          Owner:  ROB WELCH
Account:  INVOICE                        UIC:    [21,51] ([INV,WELCH])
CLI:      DCL                            Tables: DCLTABLES
Default:  USER3:[WELCH]
LGICMD:
Login Flags:
Primary days:    Mon Tue Wed Thu Fri
Secondary days:                  Sat Sun
No access restrictions
Expiration:            (none)    Pwdminimum:  6   Login Fails:     0
Pwdlifetime:           (none)    Pwdchange:   15-APR-1988 13:58
Last Login:            (none) (interactive),           (none) (non-interactive)
Maxjobs:          0  Fillm:        20  Bytlm:          8192
Maxacctjobs:      0  Shrfillm:      0  Pbytlm:            0
Maxdetach:        0  BIOlm:        10  JTquota:        1024
Prclm:            2  DIOlm:        10  WSdef:           150
Prio:             4  ASTlm:        10  WSquo:           256
Queprio:          4  TQElm:        10  WSextent:        512
CPU:        (none)   Enqlm:       100  Pgflquo:       10240
Authorized Privileges:
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX
```

4.1.1 System-Supplied UAF Records

The Authorize Utility provides a set of commands and qualifiers to assign values to any field in a UAF record. The software distribution kit provided with a new VMS system contains a UAF of four records:

- **DEFAULT**—Serves as a template for creating user records in the UAF. A new user record is assigned the values of the DEFAULT record except where you explicitly override those values. Thus, whenever you add a new account, you need only specify values for fields that you want to be different. For example, the following AUTHORIZE command creates a new record having the same values as the DEFAULT record, except that the password, UIC, and default directory fields are changed.

  ```
  UAF> ADD MARCONI/PASSWORD=QLP6YT9A/UIC=[033,004] -
  _UAF> /DIRECTORY=[MARCONI]
  ```

  Section 4.2 gives an example of how to use AUTHORIZE to add a user account.

  NOTE: The default record cannot be renamed or deleted from the UAF.

- **FIELD**—Permits DIGITAL Field Service personnel to check out a new system. The FIELD record should be disabled once the system is installed.

- **SYSTEM**—Provides a means for you to log in with full privileges. The SYSTEM record can be modified but cannot be renamed or deleted from the UAF.

  CAUTION: Do not change the SYSTEM account UAF record fields for the default device and directory, and privileges. Installation of VMS maintenance releases and optional software products depends on certain values in these fields.

- **SYSTEST**—Provides an appropriate environment for running the User Environment Test Package (UETP). The SYSTEST record should be disabled once the system is installed.

4.1.2 General Maintenance of the UAF

Usually, you use the UAF supplied with the distribution kit. (You can, however, rename the UAF with the DCL command RENAME, and then create a new UAF with AUTHORIZE.) You should limit any kind of access to this file to the SYSTEM account. Furthermore, each time you modify the file, create a backup copy so that in case of a system failure you do not lose the modifications. See Chapter 8 for procedures for backing up files.

The UAF is accessed as a shared file, and updates to the UAF are made on a per-record basis, which eliminates the need for both a temporary UAF and a new version of the UAF after each AUTHORIZE session. Updates become effective as soon as AUTHORIZE commands are entered, not after the termination of AUTHORIZE. (For this reason, you should not enter temporary values with the intent of fixing them later in the session.)

After installing the system, you should make the following modifications to the UAF:

- **SYSTEM, FIELD, and SYSTEST accounts**—If the passwords on these accounts are not secure or if they have not been changed recently, be sure to change the passwords. Use obscure passwords of six characters or more and continue to change them on a regular basis. You should not permit general users access to these accounts.

  In addition to changing the password, you can disable an account, especially if it is used infrequently. To disable an account, specify the following AUTHORIZE command:

  ```
  UAF> MODIFY username /FLAGS=DISUSER
  ```

  The login flag DISUSER disables the account and prevents anyone from logging into the account. To enable the account when it is needed, run AUTHORIZE and specify MODIFY username /FLAGS=NODISUSER. However, you should be cautious about disabling the SYSTEM account, because some optional software and some command procedures may not start up properly if the SYSTEM account is disabled.

  CAUTION: Be careful not to disable all of your privileged system accounts. If you inadvertently do so, you can recover by setting the UAFALTERNATE

SYSGEN parameter during a conversational bootstrap operation. See Chapter 2 for information on emergency startup procedures.

- **DEFAULT account**—You may want to change several fields in this account. For example:

```
UAF> MODIFY DEFAULT/DEVICE=DISK$USER/WSQUO=750
```

The default device is set to the name most commonly used for user accounts that will be added. Likewise the working set value is set to a value appropriate for most users on the system.

Use the SYSTEM account only for system functions such as performing backups and installing maintenance updates. The account comes to you with full privileges, so exercise caution in using it. For example, because you have BYPASS privilege, the system will allow you to delete any file no matter what its protection. If you type an incorrect name or spurious asterisk, you may destroy files that you or other users need to keep. For this reason, use another account with fewer privileges for day-to-day system management activities.

If you want to receive mail sent to the SYSTEM account, use the SET FORWARD command in the MAIL Utility to have any SYSTEM mail forwarded to any other account. To use the SET FORWARD command for this purpose, do the following:

1. Make sure that you are logged in to the SYSTEM account
2. Enter the MAIL Utility by entering the MAIL command at DCL level
3. At the MAIL> prompt, enter the command SET FORWARD *username*.

4.2 Adding a User Account

How you set up a user account depends on the needs of the individual user. In general, there are two types of accounts:

- **Interactive**—A person using an interactive account has access to the system software and can perform work of a general nature (program development, text editing, and so on). Usually, such an account is considered individual; that is, only one person can use it.
- **Captive**—A person using a captive account (also called a turnkey or application account) has access only to limited user software and can only perform work that is limited to a particular function. Access to a captive account is limited by function; that is, only those who perform a particular function can use it. For example, you might develop an inventory system. Anyone whose job entails inventory control can access your system, but that person cannot access other subsystems or the base software.

You should perform the following tasks in conjunction with adding a user account:

1. Determine a user name and password.

2. Determine a unique user identification code (UIC).
3. Decide where the account's files will reside (the device and directory).
4. Create a default directory on the appropriate volume, using the following DCL command:

```
$ CREATE/DIRECTORY directory-spec/OWNER_UIC= uic
```

5. Determine the security needs of the account (that is, the level of file protection, privileges, and access control).

Once you analyze the purpose of a user account and decide which attributes and resources it requires, you can use the Authorize Utility to create the account. Give yourself the SYSPRV privilege. Then enter the following commands to set your default device and directory to that of SYS$SYSTEM and invoke the utility as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF>
```

When the utility responds with the UAF> prompt, use the AUTHORIZE command ADD to specify attributes in the UAF fields as shown in this example:

```
UAF>  ADD JONES/PASSWORD=LPB57WM/UIC=[014,1] -
_UAF> /DEVICE=DISK$USER/DIRECTORY=[JONES] -
_UAF> /LGICMD=DISK$USER:[NEWPROD]GRPLOGIN -
_UAF> /OWNER="ROBERT JONES"/ACCOUNT=DOC
```

The /OWNER and /ACCOUNT entries are primarily for accounting purposes and can be omitted unless required by your site. The following unspecified qualifiers usually take their default values from the DEFAULT record:

- **Limits and Quotas** (/ASTLM, /BIOLM, /CPUTIME, /DIOLM, /ENQLM, /FILLM, /JTQUOTA, /MAXACCTJOBS, /MAXDETACH, /MAXJOBS, /PGFLQUOTA, /PRCLM, /SHRFILLM, /TQELM, /WSDEFAULT, /WSEXTENT, /WSQUOTA)—These qualifiers impose limits on the use of reusable system resources; the default values are adequate in most cases.
- **Priority** (/PRIORITY, /QUEPRIORITY)—The default values are usually adequate for accounts not running real-time processes.
- **Privileges** (/DEFPRIVILEGES, /PRIVILEGES)—The default privileges (TMPMBX, NETMBX) are usually adequate, depending on the purpose of the account.
- **Primary and Secondary Login Times; Login Functions** (/ACCESS, /DIALUP, /FLAGS, /INTERACTIVE, /LOCAL, /PRIMEDAYS, /REMOTE)—By default, users are allowed to log in at any hour of any day. To override the setting of a particular day, use the DCL command SET DAY. Use this command if a holiday occurs on a day that would normally be treated as a primary day and you want it treated as a secondary day.

The following example shows an AUTHORIZE command that adds a UAF record for a captive account:

```
UAF> ADD INVENTORY/PASSWORD=QRC7Y94A/UIC=[033,066] -
_UAF> /DEVICE=DISK$INVENT/DIRECTORY=[INV]/LGICMD=INVENTORY -
_UAF> /FLAGS=CAPTIVE/NOACCESS=(PRIMARY, 18-8, SECONDARY, 0-23)
```

In this example, the /FLAGS and /NOACCESS qualifiers restrict users from logging in to the captive account. The /NOACCESS qualifier limits logins to specific hours. The /FLAGS=CAPTIVE qualifier adds the login flag CAPTIVE to the captive account record. The CAPTIVE flag locks the person using the account into the application software by doing the following:

- Disabling the CTRL/Y function to prevent users from interrupting the execution of the command procedure and gaining access to the command interpreter
- Preventing the user from specifying an alternate command interpreter with the /CLI qualifier at login time
- Preventing the user from specifying an alternate default disk device with the /DISK qualifier at login time

The following examples summarize the steps for setting up an individual user account and a captive account:

Setting Up an Individual User Account with AUTHORIZE

```
$ SET DEFAULT SYS$SYSTEM
$
$ RUN AUTHORIZE
UAF>ADD JONES -                  ! User name
_/PASSWORD=ROCKET -              ! Password
_/UIC=[014,1] -                  ! UIC
-/ACCOUNT=DOC -                  ! Accounting group name
_/OWNER="ROCKET JONES" -         ! Owner
_/DEVICE=$DISK1 -                ! Default directory
_/DIRECTORY=[JONES]
UAF>EXIT
$
$ ! Create top-level directory for individual
$ CREATE/DIRECTORY $DISK1:[JONES] -
_$ /OWNER_UIC=[DOC,JONES] -
_$ /PROTECTION=(S:RWE,O:RWE,G:RE,W:RE)
$
```

Setting Up a Captive Account with AUTHORIZE

```
$ SET DEFAULT SYS$SYSTEM
$
$ RUN AUTHORIZE
UAF>ADD INVENTORY -                 ! User name
_/PASSWORD=RIZUPE -                 ! Password
_/UIC=[033,066] -                   ! UIC
-/ACCOUNT=INV                       ! Accounting group name
_/LGICMD=$DISK1:[INVTORY]LOGIN -    ! Login file
_/FLAGS=(DEFCLI,DISCTLY, -          ! Set flags
_DISNEWMAIL,DISWELCOME,DISMAIL)
UAF>EXIT
```

4.3 Setting Up an Automatic Login Account with ALFMAINT

You use the automatic login facility (ALFMAINT) to set up a terminal that accepts automatic logins from authorized users. For example, a terminal might be set up for the account INVENTORY, which automatically logs a user into a captive account when INVENTORY is specified as the user name.

First, you must follow the steps described in the previous sections to create the top-level default directory and add the account. Once the account has been added, you set your default directory to SYS$MANAGER and invoke the ALFMAINT command procedure. ALFMAINT prompts you for the name of the terminal that you want associated with the user name of the automatic login account.

The following example summarizes the steps for setting up automatic logins for an individual user account and a captive account:

Individual Account with Automatic Login

```
$ SET DEFAULT SYS$SYSTEM
$
$ RUN AUTHORIZE
UAF>ADD JONES -                  ! Username
_/PASSWORD= -                    ! Null password
_/UIC=[014,1] -                  ! UIC
-/ACCOUNT=DOC                    ! Accounting group name
_/OWNER="ROCKET JONES" -         ! Owner
_/DEVICE=$DISK1 -                ! Default directory
_/DIRECTORY=[JONES]
UAF>EXIT
$
$ ! Create top-level directory for individual
$ CREATE/DIRECTORY $DISK1:[JONES] -
_$/OWNER_UIC=[DOC,JONES] -
_$/PROTECTION=(S:RWE,O:RWE,G:RE,W:RE)
$

$
$ SET DEFAULT SYS$MANAGER
$
$ @ALFMAINT
```

```
Enter the name of the terminal that you would like to set for
automatic login, or a blank line or EXIT to exit.

Terminal (ddcu)? TTA1           ! Assigned terminal
Username? JONES
Terminal (ddcu)? EXIT
```

Captive Account with Automatic Login

```
$ SET DEFAULT SYS$SYSTEM
$
$ RUN AUTHORIZE
UAF>ADD INVENTORY -                 ! Username
_/PASSWORD= -                       ! Null password
_/UIC=[033,066] -                   ! UIC
_/ACCOUNT=INV -                     ! Accounting group name
_/LGICMD=$DISK1:[INVTORY]LOGIN -    ! Login file
_/ACCESS=(PRIMARY,8-17) -           ! No off hours
_/FLAGS=CAPTIVE                     ! All flags on
UAF>EXIT
$
$ SET DEFAULT SYS$MANAGER
$ @ALFMAINT

Enter the name of the terminal that you would like to set for
automatic login, or a blank line or EXIT to exit.

Terminal (ddcu)? TTA0               ! All terminals
Username? INVENTORY                 ! on automatic
Terminal (ddcu)? TTA1               ! login except
Username? INVENTORY                 ! the console terminal
Terminal (ddcu)? TTA2               ! (the console terminal
Username? INVENTORY                 ! for this system is TTA4)
Terminal (ddcu)? TTA3
Username? INVENTORY
Terminal (ddcu)? EXIT
```

4.4 Modifying a User Account

Use the AUTHORIZE command MODIFY to change any of the fields in an existing user account. For example, the following command is used to change user WELCH's password:

```
UAF> MODIFY WELCH/PASSWORD=newpassword
```

4.5 Listing User Accounts

Use the AUTHORIZE command LIST to create the file SYSUAF.LIS containing a summary of all user records in the UAF, as follows:

```
UAF> LIST
%UAF-I-LSTMSG1, writing listing file
%UAF-I-LSTMSG2, listing file SYSUAF.LIS complete
```

By default, the LIST command produces a brief report containing the following information from the UAF:

- Account owner
- User name
- UIC
- Account names
- Privileges
- Process priority
- Default disk and directory

Use the /FULL qualifier to create a full report of all the information contained within the UAF, as follows:

```
UAF> LIST/FULL
%UAF-I-LSTMSG1, writing listing file
%UAF-I-LSTMSG2, listing file SYSUAF.LIS complete
```

4.6 Deleting a User Account

The main problem in deleting an account, especially an interactive account, is cleaning up the files used by the account. The following steps are suggested:

1. Copy (or have the outgoing user of the account copy) any files of value to the ownership of another account. Be sure to change the owner UIC of the files to match the owner UIC of the new owner. You can also use the Backup Utility (BACKUP) to copy the files to a backup tape or disk.
2. Change the password, and log in to the account that you want to delete. (By working from a nonprivileged account, you can avoid inadvertently deleting files that may be owned by an account other than the one that you want to delete.)
3. Delete the account's files and directories from the deepest level up to the top level using the following procedure:
   a. Locate and examine all subdirectories using the DCL command DIRECTORY [default . . .], where *default* is the name of the account's default directory.

b. Delete the files in each subdirectory and then delete the subdirectory. Note that directory files are protected against owner deletion, therefore, you must change the protection before deleting directory files.

c. Delete the account's top-level directory. Example 4–2 illustrates a command procedure that deletes an account's files from the bottom level up.

NOTE: The command procedure in Example 4–2 should not be executed from a privileged account.

4. Remove the account, using the Authorize Utility.
5. Remove the user's disk quota entry from the disk quota file, if one existed, with the SYSMAN Utility.
6. Remove associated VAXmail information by entering the MAIL command REMOVE *username*.

Example 4–2: Command Procedure Template for Deleting an Account's Files

```
$ !     DELTREE.COM - deletes a complete directory tree
$ !
$ !     P1 = pathname of root of tree to delete
$ !
$ !     All files and directories in the tree, including
$ !     the named root, are deleted.
$ !
$ IF "''DELTREE'" .EQS. "" THEN DELTREE = "@SYS$LIBRARY:DELTREE"
$ ON CONTROL_Y THEN GOTO DONE
$ ON WARNING THEN GOTO DONE
$ DEFAULT = F$LOGICAL("SYS$DISK") + F$DIRECTORY()
$10:
$ IF P1 .NES. "" THEN GOTO 20
$ INQUIRE P1 "Root"
$ GOTO 10
$20:
$ IF F$PARSE(P1) .EQS. "" THEN OPEN FILE 'P1'
$ SET DEFAULT 'P1'
$LOOP:
$ FILESPEC = F$SEARCH("*.DIR;1")
$ IF FILESPEC .EQS. "" THEN GOTO LOOPEND
$ DELTREE [.'F$PARSE(FILESPEC,,,"NAME")']
$ GOTO LOOP
$LOOPEND:
$ IF F$SEARCH("*.*;*") .NES. "" THEN DELETE *.*;*
$ DIR = (F$DIRECTORY()-"]"-">")-F$PARSE("[-]",,,-
        "DIRECTORY")-"]"-">")-"."-"[-"<"
$ SET PROTECTION=WORLD:RWED [-]'DIR'.DIR;1
$ DELETE [-]'DIR'.DIR;1
$DONE:
$ SET DEFAULT 'DEFAULT'
```

If you never assign multiple users the same UIC, you can use the Backup Utility to remove the user's files, even if the files are scattered throughout the directory structure. The following is an example of a BACKUP command used to remove files:

```
$ BACKUP/DELETE PUBLIC:[...]/OWNER=[21,103] MTA0:PUBLICUIC.SAV
```

This BACKUP command copies and deletes only those files owned by the specified UIC on disk PUBLIC. The files are copied into a save set named PUBLICUIC on device MTA0. Note that the BACKUP/DELETE command does not delete the directory files (file extension DIR) for the account.

Disabling a User Account

If you want to disable an account without deleting it, set the disable user flag (/FLAGS=DISUSER) using AUTHORIZE. If the user is logged in, the account is disabled only after the user logs out.

Disabling a powerful yet infrequently used account provides an extra security measure by eliminating the risk of guessed or stolen passwords.

Chapter 5

Performing Batch and Print Operations

If you have a printer on your system, or if you want to use batch processing on your system, then you must use *queues*. A queue allows users to submit requests for printing or batch processing, and the system prints or processes the users' jobs as resources allow.

The system manager is responsible for setting up batch and print queues and making sure that they function properly. This chapter describes how to set up (initialize) and maintain batch and print queues for your system or cluster.

Setting up and maintaining batch and print queues are closely related system management tasks. However, you are not required to set up both types of queues if you need only one type. In a VAXcluster environment, queues can be accessed from any node on the cluster.

5.1 Generic Queues and Execution Queues

In the VMS operating system, batch and print operations support two types of queues: generic queues and execution queues.

An *execution queue* is a queue through which the job (either print or batch) is actually processed or executed. For printing, an execution queue is associated with a specific printer; for batch processing in a VAXcluster environment, an execution queue is associated with a specific node. When a print or batch job is submitted to an execution queue, the job is ultimately printed on the output device associated with that queue or processed on the associated batch queue.

You can also designate one or more individual terminals as execution queues for print jobs. You should set up a terminal as a queue when you want to allow users on your system to send output to a hardcopy terminal.

A *generic queue* is an intermediate queue that holds a job until an appropriate execution queue becomes available to initiate the job. Users can submit jobs to a generic queue, and the generic queue then directs the job to an appropriate execution queue; alternatively, users can submit jobs directly to an execution queue.

For example, suppose you have a system with several printers. You would set up individual execution queues for each of the printers, and you could also set up a generic print queue. Users would then normally submit a print job to the generic queue, and the generic queue would subsequently direct the print job to an available printer.

For batch processing, generic queues are often used in clustered systems to distribute the workload across several nodes. For example, suppose you have a Local Area VAXcluster environment with each of the satellite nodes having a batch processing queue. You could then establish a generic batch queue for the cluster. When users submit batch jobs to the generic queue, the generic queue would direct individual batch jobs to the execution queue that is best able to handle the workload.

If you only have a single printer for your system or cluster, or if you only establish a single batch queue, then there is no value in establishing generic queues.

5.2 Setting Up Queues

Set up your queues by including the appropriate commands in your site-specific startup file, SYS$MANAGER:SYSTARTUP_V5.COM. Section 2.4.3 describes the commands that should be included in SYSTARTUP_V5.COM, and this section summarizes them.

To establish and use queues, you must first start the queue manager and identify a *queue file*. To do this, include the following command in SYS$MANAGER:SYSTARTUP_V5.COM, making sure that this command appears before any other queue commands:

```
$ START/QUEUE/MANAGER/RESTART SYS$COMMON:[SYSEXE]JBCSYSQUE.DAT
```

If you have a cluster, you should use only one queue file for the cluster. Make sure that the queue file is on a disk that is accessible to all of the nodes in the cluster from which you may want to submit batch or print jobs.

When you create a generic queue, you specify a list of execution queues to which the generic queue ultimately directs jobs. In a VAXcluster environment, the execution queues that you specify for a generic queue can be on the same node as the generic queue, and they can also be on different nodes within the cluster.

Once you have established the queue file, you can set up individual execution queues and generic queues by using the INITIALIZE/QUEUE command in your SYSTARTUP_V5.COM file. Be sure to initialize execution queues before initializing the generic queues. For example, you could include the following series of commands to set up execution and generic queues for batch and print operations in a VAXcluster environment.

```
$ INITIALIZE /QUEUE /ON=BLUE::LPA0 /START  BLUE_LPA0 ❶
$ INITIALIZE /QUEUE /ON=GREEN::LPA0 /START  GREEN_LPA0 ❷
$ INITIALIZE /QUEUE /GENERIC=(BLUE_LPA0,GREEN_LPA0) /START  SYS$PRINT ❸
$
$ INITIALIZE /QUEUE /BATCH /ON=BLUE:: /START  BLUE_BATCH ❹
$ INITIALIZE /QUEUE /BATCH /ON=RED:: /START  RED_BATCH ❺
$ INITIALIZE /QUEUE /BATCH /GENERIC=(BLUE_BATCH,RED_BATCH) /START  SYS$BATCH ❻
```

This series of commands in SYSTARTUP_V5.COM does the following:

❶ Sets up an execution printer queue associated with LPA0 on node BLUE with a queue name of BLUE_LPA0.

❷ Sets up an execution printer queue associated with LPA0 on node GREEN with a queue name of GREEN_LPA0.

❸ Sets up a generic print queue for the cluster. The generic print queue has the name SYS$PRINT and directs print jobs either to BLUE_LPA0 or to GREEN_LPA0.

❹ Sets up an execution batch queue on node BLUE with the queue name BLUE_BATCH.

❺ Sets up an execution batch queue on node RED with the queue name RED_BATCH.

❻ Sets up a generic batch queue for the cluster. The generic batch queue has the name SYS$BATCH. When a job is submitted to SYS$BATCH, this generic queue directs the job either to BLUE_BATCH or to RED_BATCH.

If you want to set up a terminal as an execution queue, use exactly the same procedure as for setting up print queues and use the /DEVICE=TERMINAL qualifier in your INITIALIZE command line.

5.3 Maintaining Batch and Print Queues

Once you have modified SYS$MANAGER:SYSTARTUP_V5.COM to establish your queues properly, they will be set up and available every time that your system is booted. From time to time, however, some additional maintenance of your queues may be needed.

The VMS operating system provides several DCL-level commands that you can use to manage your queues. Table 5–1 shows some of the commands that are available for queue management. More information about these commands is available in the *VMS General User's Manual*.

Table 5–1: Queue Management Commands

Command	Description
SET QUEUE	Allows you to change the attributes of a queue (for example, the number of jobs that can execute simultaneously in a batch queue) without having to stop the queue, initialize it, and then restart it.
SHOW QUEUE	Provides the status of queues, listing the jobs that are currently executing, as well as the jobs that have not yet begun execution.
STOP /QUEUE	Allows you to pause a queue temporarily. Using the appropriate qualifiers, you can use the STOP /QUEUE command to stop jobs that are currently executing, to stop a queue after the completion of all jobs that are currently executing, to shut down the queue manager on the node from which you execute the command, and to perform other related functions.
START /QUEUE	Resumes execution of a queue that has been temporarily halted by the STOP /QUEUE command.
STOP /QUEUE /MANAGER	Shuts down the queue manager on the node from which you execute the command.
START /QUEUE /MANAGER	Starts the queue manager on the node from which you execute the command.

5.4 Monitoring Jobs

As system manager, you use SHOW QUEUE as the primary command to monitor the overall status of a queue. The SHOW QUEUE command displays the status of each queue selected, and it also shows the status of all jobs in each of the selected queues. With SHOW QUEUE, you can also obtain a summary of the status of jobs in each queue by using the /SUMMARY qualifier.

Additionally, you can also use the SHOW ENTRY command to monitor the status of jobs belonging to a particular user, or to determine the status of individual batch and print jobs. SHOW ENTRY and SHOW QUEUE each provide complete information about jobs, but SHOW QUEUE also provides status information about the queues themselves. SHOW ENTRY generally provides a faster response time than SHOW QUEUE. For a full description of the SHOW ENTRY and SHOW QUEUE commands, see the Reference section of the *VMS General User's Manual.*

The following list describes the types of job status returned by the SHOW QUEUE and SHOW ENTRY commands:

Status	Description
Aborting	Executing job is terminating
Executing	Job is executing from a batch queue
Holding	Job is being held until explicitly released
Holding until	Job is being held until a specified time
Pending	Job is in a waiting state
Printing	Job is executing from a printer or terminal queue
Processing	Job is executing from a server queue
Retained on Completion	Job remains in the queue upon completion
Retained on Error	Job remains in the queue upon encountering an error
Waiting	Symbiont refuses the job

5.4.1 Deleting a Job

Under certain circumstances, it is necessary to terminate an executing batch or print job. For example, you may need to terminate a program that has entered an endless loop or a job that is executing on a faulty print device.

Follow this procedure to delete a job:

1. Determine the entry number of the job
2. Delete the job by entering the DELETE/ENTRY command

The DELETE/ENTRY command is restricted to users with either OPER privilege, E access to the queue, or D access to the specified job.

For example, assume that you observe a job that appears to be processing in an endless loop and is using an inappropriate amount of system resources. You can delete the job by entering the command DELETE/ENTRY=entry_number. To determine the entry number, enter the command SHOW QUEUE/BATCH/ALL /BY_JOB_STATUS=EXECUTING. The following example shows how to determine the entry number and delete the job:

```
$ SHOW QUEUE/BATCH/ALL/BY_JOB_STATUS=EXECUTING

Batch queue ZEUS_BATCH, on ZEUS

 Jobname         Username     Entry  Status
 -------         --------     -----  ------
 2307SMRCL       MARCO         1719  Executing
 TEST            JONES         1720  Executing

$ DELETE/ENTRY=1719
```

5.4.2 Retaining Jobs in a Queue

To retain a job in a queue after it has been processed, specify the /RETAIN qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. The /RETAIN qualifier has the following format:

/[NO]RETAIN[=option]

By using the ERROR option with the /RETAIN qualifier, you indicate that the jobs in a queue will be held if they do not successfully complete. A job that has been held by the /RETAIN=ERROR qualifier can later be released or requeued after the problem that caused the error has been resolved.

By default, jobs are **NOT** retained.

For example, to retain print jobs that do not sucessfully complete, you could include the following command in your SYSTARTUP_V5.COM file, after you have initialized and started the queue:

```
$ SET QUEUE /RETAIN=ERROR GREEN_LPA0
```

5.4.3 Modifying Job Processing Attributes

You can modify certain job processing attributes by specifying qualifiers with the command SET ENTRY entry-number, as shown in the following table:

Qualifier	Description
/[NO]AFTER	Controls whether a job is held until a specified time
/[NO]HOLD	Controls whether a job is available for immediate processing or held until it is released for processing
/NAME	Specifies a new name for a job
/[NO]PASSALL	Specifies whether the symbiont bypasses all formatting and sends the output directly to the device driver
/PRIORITY	Specifies the relative scheduling priority of the job, with respect to the priorities of other jobs in the queue
/RELEASE	Releases a previously held job
/REQUEUE	Requests that the job be moved from the original queue to the specified queue; this qualifier can also be used with the STOP/QUEUE/ENTRY command

5.4.3.1 Holding and Releasing a Job

The /HOLD qualifier of the SET ENTRY command controls whether a job is to be made available for immediate processing. To release a held job, use either the /NOHOLD or the /RELEASE qualifier.

To request that the job be held until after a specified time, use the /AFTER qualifier with the command SET ENTRY. The job is queued for immediate processing when the specified time arrives. The /AFTER=time qualifier accepts either absolute or delta time values in the format [dd-mmm-yyyy] [hh:mm:ss.cc]. You can also specify the following keywords:

TODAY
YESTERDAY
TOMORROW

The following command holds a print job until it is queued for processing at the specified date and time:

```
$ SET ENTRY 1121/AFTER=12-JUL-1988:17:30
```

You can use the /NOAFTER qualifier to release immediately a job that is being held until a specified time.

The /RELEASE qualifier releases a job that is being held for any of the following reasons:

- A job was submitted with the /HOLD qualifier
- A completed job was held in a queue by the /RETAIN qualifier
- A job was submitted with the /AFTER qualifier

Use the SET ENTRY command with the /HOLD and /RELEASE qualifiers to hold and release a batch job. This procedure applies only to a batch job that is currently in a pending state (that is, a job that has not yet begun to execute). The following example shows how to hold and release a batch job that has not begun execution:

```
$ SET ENTRY 1234/HOLD
$ SET ENTRY 1234/RELEASE
```

5.4.3.2 Requeuing a Job

To requeue a job that has not begun execution, use the SET ENTRY /requeue COMMAND. If you want to requeue a job that has already begun execution, use the STOP/QUEUE/REQUEUE command. The STOP/QUEUE/REQUEUE command suspends the currently executing job and requeues it to the specified queue, for example:

```
$ STOP/QUEUE/REQUEUE=ALPHA_LPA0 ALPHA_LPB0
```

This command causes the executing print job on ALPHA_LPB0 to be stopped and requeued to ALPHA_LPA0. The queue does not stop; only the currently executing job is affected. Other jobs remain pending in the queue until they are processed.

You can further qualify the STOP/QUEUE/REQUEUE command with the /HOLD qualifier. To hold an aborted print job, enter the STOP/QUEUE/REQUEUE/HOLD command in the following format:

STOP/QUEUE/REQUEUE/HOLD [queue-name]

When you specify /HOLD, the aborted job is placed in a hold state for later release with the SET ENTRY/RELEASE command. If you do not need to process a job that is being held in a queue, you can delete the job with the DELETE/ENTRY command.

NOTE: If you are requeuing a job on a batch queue, you must include the /ENTRY=n qualifier, for example:

```
$ STOP/QUEUE/ENTRY=1251/REQUEUE=FRED_BATCH
```

5.4.3.3 Changing the Scheduling Priority of a Job

You can change the scheduling priority of a job by using the /PRIORITY=n qualifier with the SET ENTRY command. Do not confuse the job scheduling priority with the base priority of a queue.

The job scheduling priority value must be in a range of 0 through 255, where 0 is the lowest priority and 255 is the highest. The default value for /PRIORITY is the value of the SYSGEN parameter DEFQUEPRI (usually set at 100). You must have either OPER or ALTPRI privilege to raise the priority value above the value of the SYSGEN parameter MAXQUEPRI. No privilege is needed to set the priority of your own job lower than the MAXQUEPRI value. The following example changes the priority of a job to 50:

```
$ SET ENTRY 1131/PRIORITY=50
```

Chapter 6

Setting Up a Network

As the manager of a VMS system, you may want to connect your system to a network. This chapter describes the following network topics:

- What a DECnet network is
- How a VMS system can be part of a DECnet network
- The responsibilities of the system manager in a network environment
- The procedures needed to bring up a VMS system as a node on an existing network
- Techniques to keep the network running

NOTE: Refer to Chapter 7 if you intend to set up and manage a Local Area VAXcluster configuration. That chapter outlines the tasks required to configure a Local Area VAXcluster and describes CLUSTER_CONFIG.COM, the command procedure that you use to perform these tasks.

6.1 General Description of a DECnet Network

A DECnet network permits the linking of computers into flexible configurations to exchange information, share resources, and perform distributed processing. A VMS operating system can participate in a DECnet network through its networking interface, DECnet–VAX. As a part of a network, a VMS system can communicate with other VMS systems running on a full range of VAX processors, as well as with a wide range of non-VMS systems that use DECnet software.

DECnet distributed processing capabilities allow information to be gathered from anywhere in the network. VMS systems can be placed at locations where they are required while still having access to the facilities of other widely dispersed systems. Access to the network is available wherever it is needed: executive offices, factory floors, laboratories, field locations. Information can be exchanged between all parts of an organization or institution in a stable, integrated networking environment.

6.1.1 What Is a DECnet Network?

A DECnet network consists of two or more computer systems, called nodes, that are connected (for example, by means of cables, telephone lines, microwave or satellite links). Adjacent nodes in a network are connected by lines over which circuits operate. The line is the physical data path, and the circuit is the communications data path. All input and output (I/O) between nodes takes place over circuits. A node can be designed to have active circuits operating over a number of lines that connect that node to other nodes in the network.

DECnet permits computer processes running on the same or different computers to communicate with each other over logical links. A logical link connects two processes and carries a stream of two-way communications traffic between the processes over one or more circuits. Many logical links can be supported concurrently over a single circuit established between two nodes.

The process to which a logical link is connected is called an object. Some objects are DECnet–VAX system programs (for example, the MAIL object); other objects are user-written programs. For two programs to communicate over the network, the program on the local node establishes a logical link with the object on the remote node.

In a network of more than two nodes, the process of directing a data message from a source node to a destination node is called routing. DECnet supports adaptive routing, which permits messages to be routed through the network over the most cost-effective path; messages are rerouted automatically if a circuit becomes disabled or a lower-cost path becomes available.

Nodes can be either routing nodes (called routers) or nonrouting nodes (known as end nodes). Both routers and end nodes can send messages to and receive messages from other nodes in the network. However, a router has the ability to forward or route messages from itself to another node. A router can serve as an intermediate node on a path between two nodes exchanging messages, if the two nodes have no direct physical link to each other. Any node that has two or more active circuits connecting it to the network must be a router. An end node can only have one active circuit connecting it to the network.

A DECnet network can vary in size from a small to a very large network. A typical small network might consist of two to four nodes. A maximum of 1023 nodes is possible in an undivided network, but the optimum number is approximately 200 to 300 nodes, depending on the topology (the way the nodes and lines are arranged in the network).

Very large DECnet networks can be divided into multiple areas: up to 63 areas, each containing a maximum of 1023 nodes. In a multiple-area network, the network manager groups nodes into separate areas, with each area functioning as a subnetwork. Nodes in any area can communicate with nodes in other areas. DECnet supports routing within each area and a second, higher level of routing that links the areas, resulting in less routing traffic throughout the network. Nodes that perform routing within a single area are referred to as level 1 routers; nodes that

perform routing between areas as well as within their own area are called level 2 routers (or area routers).

The DECnet architecture follows industry standards and is designed to permit easy expansion and incorporation of new developments in data communications. DECnet offers the option of communicating over different kinds of network connections, which are for the most part transparent to the general user of the network.

6.1.2 How DECnet–VAX Serves as the VMS Interface to the Network

DECnet is the collective name for the software and hardware products that are a means for various DIGITAL operating systems to participate in a network. DECnet–VAX is the implementation of DECnet that allows a VMS operating system to function as a network node. As the VMS network interface, DECnet–VAX supports both the protocols necessary for communicating over the network and the functions necessary for configuring, controlling, and monitoring the network.

DECnet–VAX networking software can be configured on any VMS operating system running on any VAX processor. In a DECnet network, a DECnet–VAX node can communicate with other DECnet–VAX nodes or with any other operating system that supports DECnet. In addition, a DECnet–VAX node can use a packet switching network to communicate with nodes on other networks, and can use gateways and other special software and hardware products to communicate with foreign vendor systems.

DECnet–VAX is tightly coupled to the VMS operating system. It is completely integrated into the operating system and provides a natural extension of local I/O operations to remote systems. VMS users can use the network almost transparently.

Because DECnet–VAX is a part of the VMS operating system, you can use the DECnet–VAX interface as a standard part of a standalone operating system (for example, to prepare network application programs).

Before you can bring up your system as a node in a multinode environment, you must have a DECnet–VAX license and register a DECnet–VAX key on your system.

6.1.3 What Does a DECnet Network Look Like?

DECnet–VAX supports a variety of network connections, permitting computers to be linked in flexible configurations. The basic kinds of environments into which a DECnet network can be configured are the local area network and the wide area network. The local area network permits communication within a limited geographic area, while a wide area network permits long-distance communication. Local area networks and wide area networks can be integrated into a single large network.

A local area network provides a high-speed communications *channel* designed to connect information processing equipment in a specific geographic area, such as an office, a building, or a cluster of buildings (for example, a campus). The DIGITAL local area network uses the Ethernet: a single shared network channel. All nodes have equal access to the channel. Because the Ethernet is a multi-access device, new nodes can be added without affecting existing nodes on the Ethernet.

An Ethernet is a coaxial cable, to which each system or device is conected by a single line. In an office or other area where personal computers and workstations are located, ThinWire Wthernet cabling is usually used. The Ethernet supports a very high rate of data transmission (up to 10 million bits per second) in a limited area.

DECnet–VAX also offers comprehensive wide-area network support and long-haul connectivity over point-to-point connections. Point-to-point connections, which use the DIGITAL Data Communications Message Protocol (DDCMP), are synchronous or asynchronous. Synchronous devices provide high-speed connections over dedicated lines or telephone lines (using modems). Asynchronous devices provide low-speed, low-cost connections over terminal lines that are switched on for network use either permanently (a static connection) or temporarily (a dynamic connection). For example, a user on a MicroVAX can configure a dialup line to another computer as a dynamic asynchronous DECnet line for the duration of a telephone call.

6.1.4 System and Network Manager Responsibilities

As system manager of a DECnet–VAX node, you are responsible for establishing your system as a node in the network, and controlling and monitoring your node. To configure your system as a network node, you must supply information at the local node about network components, including the local node, remote nodes, circuits, lines, and objects. This information constitutes what is called the configuration database for the local node. Each node in the network has such a database. As manager of your system, you supply information about the configuration database using the Network Control Program (NCP) Utility.

If you are configuring a DECnet–VAX node for the first time or rebuilding the configuration database for your local node, you can use the interactive NETCONFIG.COM procedure to configure your node automatically. Once you start up your DECnet–VAX node and verify its connection to the network, you can use the NCP Utility to control and monitor local network operation, and test network software operation.

Planning for configuration of your node in an existing network usually involves coordinating with the system managers of other nodes in the network or with the manager of the network (if a manager has been designated) to ensure uniform network parameter settings.

To create a new network, the managers of individual systems should connect their systems by means of communications lines; the system managers should then configure their own systems as network nodes and start DECnet on their nodes.

A system manager of a network node may also be called upon to provide DECnet–VAX host services for other DECnet nodes. Host services include loading system images and programs downline to unattended remote nodes, and receiving for interpretation upline dumps of system images from nodes that have crashed. For example, DECnet–VAX permits you to load an operating system image or a terminal server image downline to a target node. Another DECnet–VAX host service involves connecting to an unattended remote node (for example, a diskless communications server) to act as its console.

For a larger network, one person, who may be the manager of a network node, is usually designated as the manager of the network. The network manager is responsible for planning, building and fine-tuning a whole network to run with maximum efficiency. The network manager makes networkwide configuration decisions, such as the kinds of paths to be established, which nodes should be routers or end nodes, and whether the network should be divided into areas. The network manager also sets values for network parameters that should be the same across the network.

Managing a network usually involves regular monitoring to detect patterns of usage and error conditions on the network, and performing remote configuration of the network to control traffic patterns and accommodate network growth. System and network managers also perform maintenance procedures (to avoid serious problems from developing) and troubleshooting procedures (to resolve problems quickly). Using network software, the manager can obtain statistics on network usage and routing parameters. Network logging files provide error statistics useful in diagnosing potential problems. NCP commands display the status of nodes, lines and circuits in the network.

6.2 Getting Started on the Network

There are two ways to establish your VMS system as part of a DECnet–VAX network:

- **Bring up your VMS system as a network node:** If you are the manager of a VMS system, you can physically connect your system to an existing DECnet network by means of a communications line, and bring your system up as a network node by performing the DECnet–VAX installation procedure. The DECnet–VAX installation procedure you perform on your system involves registering the DECnet–VAX key using the VMS License Management Utility, configuring your node as part of the network, starting the network, and verifying that you are connected to the network.
- **Create a new network:** If there is no existing network to which you can connect, you can cooperate with the managers of other systems to create a new network. A network is formed when two or more systems are connected by communications lines and each system is brought up as a network node. For larger networks, the system manager of one node may also manage the network.

6.2.1 Preparing to Bring Up Your System as a Node on an Existing Network

If you are the system manager of a VMS system, you can install the DECnet–VAX license and configure your system as a node on an existing network. You can be connected permanently to the network, in which case you will be able to communicate with any other node on the network. You can also optionally choose to establish a temporary connection to another system over a telephone line. This temporary DECnet connection between two systems may result in a network that exists only for the duration of the telephone call.

Before you begin the procedure for starting DECnet–VAX on your system, you should check your hardware and connect any required communications lines. You should also prepare your VMS operating system for the network environment and decide how you want to configure your node.

6.2.1.1 Connecting the Communications Hardware on Your System

A network is a flexible configuration of computers and terminals interconnected by communications lines. You should identify the equipment you need to connect your VAX computer to an existing network. For each connection, this equipment normally includes

- A communications controller device (line device) that contains one or more interface points called ports. (The line device is installed on your processor.)
- A communications line to connect the port to the network.

Consult your DIGITAL sales support representative if you are not familiar with the equipment that you require, or if you need to install such equipment. Following the instructions in the hardware user manuals included with the equipment, you should be able to connect each network communications line to the appropriate port.

A VAX computer on which a VMS operating system is running can be connected to the network by means of high-speed lines (such as an Ethernet cable or a synchronous point-to-point line). A VMS system can also be connected to a network by means of a low-speed, low-cost asynchronous line. An asynchronous point-to-point connection can be established over any VMS terminal line between a VMS system and another system (which can be a non-VMS system) that supports the DECnet asynchronous DIGITAL Data Communications Message Protocol (DDCMP). An asynchronous connection can optionally be made over a dialup line (for example, a telephone line) if a modem is used at each end of the connection. A modem is a device that connects the terminal line to the telephone line. Modems may be purchased separately from DIGITAL, along with the appropriate installation documentation.

A VAX processor can have a number of communications ports, depending on the model. The possible connections are limited only by the number of devices that your processor can support, as specified in the DECnet–VAX Software Product Description load unit tables, and the devices that you configure for your node.

6.2.1.2 Preparing Your VMS System for the Network Environment

Before you bring up DECnet–VAX on your system, you should take the following steps to prepare your system to function as part of the network:

- Check to see if you have the privileges you need to perform network operations. The minimum privileges that a system manager normally requires to configure and control the network and run network programs are SYSPRV, OPER, TMPMBX, NETMBX, and BYPASS. A list of privileges required for network operations appears in Table 6–1.

  Enter the DCL command SHOW PROCESS/PRIVILEGES to determine which of your authorized privileges are currently enabled, and use the SET PROCESS /PRIVILEGES command to enable any additional privileges you are authorized to have that are required for network operations.

- Decide whether you want to establish a default nonprivileged DECnet account and directory. The nonprivileged account is a default DECnet account that is used in either of the following conditions:

  - When a user on a remote network node does not explicitly supply access control information (the user name and password) when requesting a connection to the local node, or
  - There is no proxy account to use on your system

  An account is required to use certain VMS utilities, such as MAIL or PHONE, over the network. If you want the default account, you can request that the DECnet–VAX configuration procedure, NETCONFIG.COM, establish a default nonprivileged account and directory for your node automatically. As an alternative, you can establish a nonprivileged account and directory manually.

- Set up any proxy accounts that you want to establish for your node. A proxy login account allows a user on a remote node on the network to access data by way of a local account on your system. You should never grant proxy access to privileged accounts.

- If necessary, tune your VMS system to accommodate DECnet–VAX software. The network manager who establishes network configuration guidelines should provide you with any required information if you need to update VMS system parameters and quotas.

Table 6-1: VMS Privileges Required for DECnet-VAX Operations

Privilege	Network Operations
ACNT	Required to start the network; permits you to suppress accounting messages
BYPASS	Permits you to view passwords in the DECnet-VAX databases
CMKRNL	Required to start the network; permits a process to access the VMS kernel
DETACH	Required to start the network; allows you to create detached processes
NETMBX	Required for all network users; needed for any network operation; needed to create a logical link
OPER	Allows you to perform operator functions such as modifying the DECnet-VAX volatile database
TMPMBX	Required for all network users and default DECnet accounts; needed to run NCP and to create a temporary mailbox
SYSNAM	Permits you to declare a name or object number in a user task
SYSPRV	Required to access the DECnet-VAX permanent database

6.2.1.3 Planning the Configuration of Your DECnet-VAX Node

Before you specify how your node is to be configured as part of an existing network, you should make the following decisions:

- Select a unique node name and node address for your system. If a network manager has been designated for your network, request a node name and address from the network manager. If your node is a member of a VAXcluster, obtain your node name and address from the VAXcluster manager. (The VAXcluster node name must be set in the VMS system parameter SCSNODE and the node address in SCSSYSTEMID.)

  Each node in the network is identified by a specific name and a numeric address by which the node is known to other nodes in the network. The node name can be no more than six alphanumeric characters (including at least one alphabetic character). The node address consists of an area number (in the range from 1 to 63, with a default value of 1) and a node number (in the range from 1 to 1023) separated by a period (for example, 2.2).

  If your node is a member of a VAXcluster that uses an alias node identifier (an alias name or address), you can obtain the alias identifier from the VAXcluster manager. An alias node identifier, common to some or all nodes in a cluster, permits remote nodes to treat the cluster as though it were a single node. Individual nodes in a VAXcluster can optionally assume the alias, while retaining their individual node names. You can use the alias adopted by the cluster, as well as your own node name, to communicate with other nodes in the network.

- Determine the node names and addresses of all other nodes in your network to which you want to connect. To obtain the correct node name and address of each node, you can contact the network manager or, if necessary, the individual system managers of the other nodes. You must enter this remote node information in your network node database.

  Alternatively, you can determine whether the names and addresses of the nodes that you want to define are already defined in the network database of another node. If you have the appropriate access, you can copy the node database information from a remote node into your node database.

- Decide whether your system is to be a router or an end node. If you have a DECnet full function license and the accompanying DECnet–VAX key, you have the option of configuring your system as either a router or an end node. If your DECnet license and key are for the end node capability, you can only configure your system as an end node.

- Determine the types of connections that will be made to the network: Ethernet, synchronous DDCMP, or asynchronous DDCMP connections. You can use the network configuration procedure NETCONFIG.COM to configure all circuits and lines automatically except for asynchronous circuits and lines.

6.2.2 Installing DECnet–VAX on Your System

This section describes the procedure for installing DECnet–VAX on your VMS operating system. Use this installation procedure to bring up your system as a node on an existing DECnet network.

To perform the installation procedure, you should log in to the SYSTEM account on your node. The DECnet–VAX installation procedure consists of the following steps:

1. Purchase the DECnet–VAX license and the DECnet–VAX key and register the key on your system, using the VMS License Management Utility.
2. Configure your DECnet–VAX node and define the remote node names. You can configure your node and turn on the network at your node either automatically or manually.
3. If you plan to use an asynchronous DECnet connection, perform any steps needed to establish the connection.
4. Verify that your node is connected to the network.

6.2.2.1 Getting a DECnet–VAX License and Key

To permit your node to communicate with other nodes in the network, you must have a DECnet–VAX license and register a DECnet–VAX key on your system using the VMS License Management Utility. You can purchase either an end node or a full function license and the corresponding key. The end node key permits you to configure your node only as an end node. The full function key permits you to configure your node as either a routing node or an end node. You can also use the full function key to upgrade from end node to full function capability.

You can register the DECnet–VAX key as the initial step in bringing up the network, or you can register it after performing the automatic configuration of DECnet–VAX (using NETCONFIG.COM), as described in the following section. Be sure to determine whether the key you are registering is for the full function or end node DECnet capability. The full function DECnet–VAX key is DVNETRTG; the end node DECnet–VAX key is DVNETEND.

6.2.2.2 Configuring Your DECnet–VAX Node

You are now ready to configure your DECnet–VAX system. You can configure the node automatically or manually.

- You can use the automatic configuration procedure when you first bring up the node or when you reconfigure it completely.
- You can use manual configuration techniques to bring up a new node, reconfigure a node, or modify an existing configuration.

The system manager at each node in the network is responsible for the DECnet–VAX configuration database for the node. The database includes files that describe the local (executor) node and the other nodes in the network with which the local node can communicate, as well as the circuits and lines that connect the local node to the network. The network database also includes information on the logging collection points (such as the logging monitor) to which network events are reported. In addition, DECnet–VAX provides a default object database describing objects (such as MAIL) known to the network. Each node in the network has such a database.

The configuration database comprises the *volatile database* (the working copy of the database that reflects current network conditions) and the *permanent database* (which provides the initial values for the volatile database when you start the network). Modifications to the volatile database exist only while the network is running. Changes made to the permanent database remain after the network is shut down, but do not affect the current system.

As system manager, you provide network component information, from the point of view of the local node, in the configuration database at the local node. Use the Network Control Program (NCP) to supply this information in the form of parameter values, which determine how the various components of the network function together. Use NCP DEFINE commands to establish the contents of the permanent database and SET commands to specify the contents of the volatile database. Use

PURGE commands to delete permanent database entries and CLEAR commands to delete or reset volatile database entries.

Configuring Your Node Manually

You can always configure your node manually; however, you have the option of doing it automatically (as described in the next section) if you are configuring a new node or completely reconfiguring a node.

If you decide to configure your node manually, you must enter NCP commands to establish the permanent configuration database and then turn on the network manually, causing the contents of the permanent database to be entered in the volatile database. A brief explanation of how to use NCP to establish your configuration database manually appears later in this section.

If you decide to configure your node manually, you can optionally create a default nonprivileged DECnet account and directory for your node manually.

Configuring Your Node Automatically

You can use the interactive command procedure NETCONFIG.COM to configure your system automatically. NETCONFIG.COM configures all required permanent database entries except for remote nodes, asynchronous circuits, and lines. You can also use the command procedure to set up an optional default nonprivileged DECnet account on your system.

Use NETCONFIG.COM only if you are bringing up your node for the first time, or want to reconfigure your node completely. The procedure purges any existing permanent database entries on your system (except for remote node entries). You must have the privilege SYSPRV to use NETCONFIG.COM to configure your node.

If you decide to use the NETCONFIG.COM command procedure to configure your node automatically, perform the following steps. Default values appear in brackets [] after certain questions in the interactive dialog. To accept a default, press RETURN.

1. **Log in to the SYSTEM account on your node.**
2. **Invoke NETCONFIG.COM.** Enter the following command at the dollar sign ($) prompt:

   ```
   $ @SYS$MANAGER:NETCONFIG
   ```

   The following message is displayed:

   ```
                  DECnet-VAX network configuration procedure

   This procedure will help you define the parameters needed to get DECnet
   running on this machine.  You will be shown the changes before they are
   executed, in case you want to perform them manually.
   ```

3. **Provide the node name.** You will be prompted as follows:

   ```
   What do you want your DECnet node name to be?
   ```

Enter the node name you have selected (or have been assigned by the network manager). Your node name must be six alphanumeric characters or less, and must be unique among all node names in the network.

(If you are on a VAXcluster node, you must press RETURN to accept the default node name that appears in brackets at the end of the prompt. This default node name is based on the SYSGEN parameter SCSNODE. If no default node name is displayed, exit the procedure and use SYSGEN to set up a value for SCSNODE, then restart the procedure. The DECnet node name of a VAXcluster node must match the value of SCSNODE.)

4. **Provide the node address.** You will be prompted as follows:

```
What do you want your DECnet address to be?
```

Enter the node address you have selected (or been assigned by the network manager). The node address is a numeric value of the following format:

```
area-number.node-number
```

Area-number (1 to 63) designates the area in which the node is grouped and **node-number** (1 to 1023) designates the node's unique address within the area. If you do not specify an area number, the system will supply a default area number (the default value is 1).

(If you are on a VAXcluster node, you must press RETURN to accept the default node address that appears in brackets at the end of the prompt. This default node address is based on the SYSGEN parameter SCSSYSTEMID. If no default node address is displayed, exit the procedure and use SYSGEN to set up a value for SCSSYSTEMID, then restart the procedure. The DECnet node address of a VAXcluster node must match the value of SCSSYSTEMID.)

5. **Specify router or nonrouter status.** You will be prompted as follows:

```
Do you want to operate as a router?  [NO (nonrouting)]
```

Press RETURN to operate as a nonrouter (that is, as an end node). Type YES and press RETURN if you want your system to be a router and if you have registered the DECnet–VAX full function key or will register it before you start up the network.

6. **Set up the nonprivileged DECnet account.** You will be prompted as follows:

```
Do you want a default DECnet account? [YES]
```

Press RETURN to set up the default nonprivileged DECnet account and directory. Type NO and press RETURN if you do not want to set up the account.

7. **Apply the configuration.** The network configuration procedure displays the list of commands necessary to start up your network. (An example showing the commands appears later in this section.)

You will be prompted as follows:

```
Do you want these commands to be executed? [YES]
```

Press RETURN to configure the network; type NO and press RETURN to cancel the configuration operation. If you choose to configure the network, the procedure displays a series of information messages and the following statement:

```
The changes have been made.
```

8. **Turn on the network.** You will then receive the following messages, ending in a prompt:

```
If you have not already registered the DECnet-VAX key, then do so now.
After the key has been registered, you should invoke the procedure
SYS$MANAGER:STARTNET.COM to start up DECnet-VAX with these changes.
(If the key is already registered) Do you want DECnet started? [YES]:
```

You can respond to this prompt in either of the following ways:

- Type NO and press RETURN in response to the last prompt if you need to register the key on your system at this point. Register the key using the VMS License Management Utility. Once the DECnet–VAX key is registered, you can then start up DECnet–VAX manually with these configuration changes by entering the following command:

  ```
  $ @SYS$MANAGER:STARTNET
  ```

  (You can also type NO and press RETURN in response to the last prompt if the key is already registered but you do not want to start the network until a later time.)

- Press RETURN in response to the last prompt if you want to start the network at this time and the key is already registered. The procedure turns on the network and displays the identification numbers of the created processes. When the dollar sign ($) prompt appears, you have successfully configured and turned on the DECnet–VAX network.

  If you want the network to be started automatically each time the VMS operating system is booted, enable the following command in the SYS$MANAGER:SYSTARTUP_V5.COM command procedure (by deleting the exclamation point at the beginning of this command line in the command procedure):

  ```
  $ @SYS$MANAGER:STARTNET
  ```

- Note that you can optionally press RETURN to start the network without the key being registered, if you want to use DECnet–VAX only at your local node. The key is required if you want to establish connections to other nodes in the network.

Example 6–1 shows the interactive dialog that is displayed when you invoke NETCONFIG.COM to configure node PURPLE with address 2.3 as an end node with a default DECnet account. In this example, node PURPLE is connected to Ethernet circuit UNA-0.

Example 6-1: Sample NETCONFIG.COM Dialogue

```
        DECnet-VAX network configuration procedure
This procedure will help you define the parameters needed to get
DECnet running on this machine.  You will be shown the changes before
they are actually executed, in case you want to perform them manually.

What do you want your DECnet node name to be?          :  PURPLE
What do you want your DECnet address to be?            :  2.3
Do you want to operate as a router?   [NO (nonrouting)]:  RET
Do you want a default DECnet account?             [YES]:  RET

         Here are the commands necessary to set up your system.

----------------------------------

$ RUN SYS$SYSTEM:NCP
    PURGE EXECUTOR ALL
    PURGE KNOWN LINES ALL
    PURGE KNOWN CIRCUITS ALL
    PURGE KNOWN LOGGING ALL
    PURGE KNOWN OBJECTS ALL
    PURGE MODULE CONFIGURATOR KNOWN CIRCUITS ALL
$ DEFINE/USER SYS$OUTPUT NL:
$ DEFINE/USER SYS$ERROR NL:
$ RUN SYS$SYSTEM:NCP      ! Remove existing entry, if any
    PURGE NODE 2.3 ALL
    PURGE NODE PURPLE ALL
$ RUN SYS$SYSTEM:NCP
    DEFINE EXECUTOR ADDRESS 2.3 STATE ON
    DEFINE EXECUTOR NAME PURPLE
    DEFINE EXECUTOR MAXIMUM ADDRESS 1023
    DEFINE EXECUTOR ROUTING TYPE NONROUTING IV
    DEFINE EXECUTOR NONPRIVILEGED USER DECNET
$ DEFINE/USER SYSUAF SYS$SYSTEM:SYSUAF.DAT
$ RUN SYS$SYSTEM:AUTHORIZE
    ADD DECNET /OWNER="DECNET DEFAULT" -
        /PASSWORD="" -
        /UIC=[376,376] /ACCOUNT=DECNET -
        /DEVICE=SYS$SYSDEVICE: /DIRECTORY=[DECNET] -
        /PRIVILEGE=(TMPMBX,NETMBX) -
        /FLAGS=(CAPTIVE) /LGICMD=NL: -
        /NOBATCH /NOINTERACTIVE
$ CREATE/DIRECTORY SYS$SYSDEVICE:[DECNET] /OWNER=[376,376]
$ RUN SYS$SYSTEM:NCP
    DEFINE LINE    UNA-0 STATE ON
    DEFINE CIRCUIT UNA-0 STATE ON COST 1
    DEFINE LOGGING MONITOR STATE ON
    DEFINE LOGGING MONITOR EVENTS 0.0-9
    DEFINE LOGGING MONITOR EVENTS 2.0-1
    DEFINE LOGGING MONITOR EVENTS 4.2-13,15-16,18-19
    DEFINE LOGGING MONITOR EVENTS 5.0-18
    DEFINE LOGGING MONITOR EVENTS 128.0-4

----------------------------------
```

Example 6-1 Cont'd. on next page

Example 6–1 (Cont.): Sample NETCONFIG.COM Dialogue

```
Do you want these commands to be executed?                    [YES]: RET
The changes have been made.
If you have not already registered the DECnet-VAX key, then do so now.
After the key has been registered, you should invoke the procedure
SYS$MANAGER:STARTNET.COM to start up DECnet-VAX with these changes.

(If the key is already registered) Do you want DECnet started?[YES]: RET
```

9. **Define the other node names.** At the dollar sign ($) prompt, invoke the Network Control Program (NCP) by entering the following command:

   ```
   $ RUN SYS$SYSTEM:NCP
   ```

   For each remote node in the network that you want to identify by node name, enter the following command to define the node address and name in your permanent node database:

   ```
   NCP>DEFINE NODE address NAME name
   ```

   Address is the existing node address in the form *area-number.node-number*, and **name** is the node name. If you omit the area number from the node address, the area number of your local node is used. The network manager or the system manager of the remote node you want to define can provide you with the correct name and address.

   If a node that you can access on your network has a node database that contains all the node names and addresses you want to define and you have the appropriate privileges to access that database, you can enter the following command at the NCP prompt (provided the network is turned on):

   ```
   NCP>COPY KNOWN NODES FROM node-id TO PERMANENT
   ```

   In this command, **node-id** is the node name or address of the remote node from which you are copying the information. If you specify the node name, that name must be in your volatile database. All the node names and addresses are copied to your permanent node database from the volatile node database of the remote node.

   If your node is a member of a VAXcluster that uses an alias node identifier (an alias node name and address), your node can adopt the alias. Specify the following commands to define the alias node address and name in the permanent node database, and associate the alias identifier with your node:

   ```
   NCP>DEFINE NODE address NAME name
   NCP>DEFINE EXECUTOR ALIAS NODE node-id
   ```

   For the **node-id**, you can specify either the alias node address or name that you have defined. Your node can then be identified by the alias node name and address as well as by its unique node name and address when DECnet is running.

Then enter the following commands to create the volatile node database for your node:

```
NCP> SET KNOWN NODES ALL
NCP> EXIT
```

The other nodes on the network should define your node name and node address in their node databases in order to be able to communicate with your node by node name. If a network manager assigned the unique node name and address to your node, the manager can define your node name in an overall network node database.

10. **Determine how to proceed.** You have completed the network startup procedure. If you plan to use asynchronous DECnet, continue to the next section, which describes how to establish asynchronous connections.

6.2.2.3 Establishing Asynchronous DECnet Connections to Other Systems

The automatic network configuration procedure described in the previous section does not configure asynchronous lines and circuits. As a VMS system manager, you have the option of connecting your VMS system to another system by means of a low-cost, low-speed asynchronous DECnet line. The two types of asynchronous DECnet connections you can establish are as follows:

- A static asynchronous DECnet connection, which creates a permanent DECnet link to a single remote node.
- A dynamic asynchronous DECnet connection, which provides a temporary DECnet link. You can establish dynamic connections to different remote nodes at different times.

Note that non-VMS systems that support DECnet asynchronous DDCMP lines can make asynchronous DECnet connections to VMS systems. The asynchronous connection can be between two routers, a router and an end node, or two end nodes. If you are on an end node and want to make an asynchronous connection, it will be your only connection to the network, because an end node can only have one circuit active at a time.

Establishing a Static Asynchronous Connection

A static asynchronous DECnet connection is a permanent connection between two nodes. This type of connection can be made in one of two ways:

- The nodes can be connected by a physical line (a null modem cable) attached to a terminal port at each system. No modems are required. You can communicate with the other system at any time.
- The connection can be made over a dialup line using modems at both ends of the line. For example, your VMS system can establish a static asynchronous connection to a remote node over a telephone line.

You can configure your static asynchronous line as soon as you have executed NETCONFIG.COM, and then turn on the network manually. If your system is brought up as a routing node, you can establish a static asynchronous connection at any time, no matter how many network connections you already have.

Follow the steps outlined in this section to establish a static asynchronous connection. For the connection to be successful, the node with which you are creating a DECnet link must also establish an asynchronous DECnet connection with your node. (Note that the line speeds at each end of the connection must be the same.)

1. Log in to the SYSTEM account on your VMS node.
2. Load the asynchronous DDCMP driver, NODRIVER (NOA0). Enter the commands shown below at your terminal (or include them in the SYS$MANAGER:SYSTARTUP_V5.COM command procedure before you boot the system).

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  CONNECT NOA0/NOADAPTER
SYSGEN>  EXIT
```

   The asynchronous driver must be loaded before any asynchronous connection can be made.

3. To set up the terminal line to become a static asynchronous DECnet line, enter the DCL command SET TERMINAL at your terminal. If there is more than one terminal attached to your VMS system, you must specify a SET TERMINAL command for each terminal line that will be used for a static asynchronous DECnet connection.

   - **Nondialup line:** For a nondialup configuration, enter the following SET TERMINAL command to convert a terminal line to a static asynchronous line:

     ```
     $ SET TERMINAL/PERMANENT/PROTOCOL=DDCMP device-name:
     ```

     In this command, **device-name** is the name of the terminal port that is connected to the line that you want to make a static asynchronous DECnet line. (All references to a device in this section refer to the terminal port.)

   - **Dialup line:** For a dialup configuration, enter the following SET TERMINAL command to convert the terminal line to a static asynchronous DECnet line with modem control.

     ```
     $ SET TERMINAL/PERMANENT/MODEM/NOAUTOBAUD -
     _$ /NOTYPE_AHEAD/PROTOCOL=DDCMP device-name:
     ```

   You can ensure that these SET TERMINAL commands will be executed automatically each time the network is started. Modify your SYS$MANAGER:SYSTARTUP_V5.COM command procedure to include all required SET TERMINAL commands before the command @SYS$MANAGER:STARTNET.

4. After configuring your node, configure the asynchronous lines and circuits in the network database. Use NCP commands to define each asynchronous line and accompanying circuit as being in the ON state. (The line and circuit are turned on when SYS$MANAGER:STARTNET.COM is executed.) Enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE LINE dev-c-u STATE ON RECEIVE BUFFERS 4 -
     _LINE SPEED  baud-rate
NCP>DEFINE CIRCUIT dev-c-u STATE ON
NCP>EXIT
```

Baud-rate is the speed at which the line sends and receives data. For an asynchronous line or circuit, **dev-c-u** is defined as follows:

dev The first two letters of the asynchronous device name (possible values are TT and TX).

c A decimal number (0 or a positive integer) designating a device's hardware controller. If the third letter of the device name is A, **c** equals 0. If the third letter of the device name is B, **c** equals 1, and so on.

u The unit number of the device name; **u** is always equal to 0 or a positive integer.

(An example is the device identifier TT-0-0, which represents the asynchronous device name TTA0.)

A minimum of four buffers should be allocated for data reception over the line.

If the line speed at the other end of the connection is changed after the initial static asynchronous connection is made, you can use the DEFINE LINE command specified in step 4 to change the line speed for your end of the connection to match the line speed at the other end. The line speed will be changed the next time the line is turned on.

5. For security over a dialup connection, you can run NCP and establish optional transmit and receive passwords for the local end of the static asynchronous dialup link. The transmit password is the password sent to the other node during connection startup; the receive password is the password expected from the other node during connection startup. You must also use NCP to specify that your asynchronous circuit is to verify the password supplied by the other node. If the correct passwords are not supplied, the asynchronous connection cannot be made.

   Although transmit and receive passwords are not mandatory for static asynchronous dialup links, they add to the security of your DECnet connection. Passwords can contain from one to eight alphanumeric characters and must be delimited with quotation marks if they contain spaces. Specify the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE CIRCUIT dev-c-u VERIFICATION ENABLED
NCP>DEFINE NODE node-id TRANSMIT PASSWORD transmit-password -
     _RECEIVE PASSWORD receive-password
NCP>EXIT
```

Node-id is the name of the remote node to which your node will be connected.

Note that if you have defined passwords for the local end of the link, you must notify the remote node system manager to establish transmit and receive passwords for the remote end of the static asynchronous DECnet dialup link.

6. If the network is not already on, turn on the network at your node by entering the following command:

```
$ @SYS$MANAGER:STARTNET
```

This command starts the network and causes the permanent database entries defined in the previous steps to be entered in the volatile database on the running network.

If the network was already running before you began the static asynchronous connection procedure, enter the following commands to cause the permanent database entries to be entered in the volatile database.

```
$ RUN SYS$SYSTEM:NCP
NCP> SET LINE dev-c-u ALL
NCP> SET CIRCUIT dev-c-u ALL
NCP> SET NODE node-id ALL
NCP> EXIT
```

If the line and circuit could not be set on in the volatile database, causing DECnet to fail to gain control of the line, the following error message is displayed:

```
% NCP-I-NMLRSP, LISTENER RESPONSE - Operation failure
```

If the static asynchronous connection cannot be made, refer to the section on asynchronous connection problems.

7. If you want to turn off the asynchronous lines temporarily, run NCP and enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP> SET LINE dev-c-u STATE OFF
NCP> SET CIRCUIT dev-c-u STATE OFF
NCP> CLEAR LINE dev-c-u ALL
NCP> CLEAR CIRCUIT dev-c-u ALL
NCP> EXIT
```

To turn the static asynchronous DECnet line back on, enter the following NCP commands:

```
$ RUN SYS$SYSTEM:NCP
NCP> SET LINE dev-c-u ALL
NCP> SET CIRCUIT dev-c-u ALL
NCP> EXIT
```

8. If you want to switch an asynchronous DECnet line back to a terminal line with DECnet running, you must clear the line and circuit entries from the network volatile database. To clear the entries, enter these commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LINE dev-c-u STATE OFF
NCP>SET CIRCUIT dev-c-u STATE OFF
NCP>CLEAR LINE dev-c-u ALL
NCP>CLEAR CIRCUIT dev-c-u ALL
NCP>EXIT
```

To switch the line for which modem control was not enabled back to a terminal line, enter the following command:

```
$ SET TERMINAL/PERMANENT/PROTOCOL=NONE device-name:
```

To switch the line for which modem control was enabled back to a terminal line, enter the following command:

```
$ SET TERMINAL/PERMANENT/MODEM/AUTOBAUD -
_$ /TYPE_AHEAD/PROTOCOL=NONE device-name:
```

Establishing a Dynamic Asynchronous Connection

A dynamic asynchronous DECnet connection is a temporary connection between two nodes, normally over a telephone line through the use of modems. The line at each end of the connection can be switched from a terminal line to a dynamic asynchronous DECnet line. Configuration of dynamic asynchronous lines is performed automatically by DECnet during establishment of a dynamic connection. A dynamic asynchronous connection is normally maintained only for the duration of a telephone call.

NOTE: A dynamic asynchronous connection to a VMS node can be initiated from any VMS or non-VMS node that supports the DECnet asynchronous DDCMP protocol.

On a VMS node, you have the option of performing the initial steps of the dynamic asynchronous connection process (steps 1 and 2 as follows) before you turn on the network at your node (step 3). The later steps of the process (starting with step 4) must occur when the line is being switched to DECnet.

Follow the steps listed in this section to establish a dynamic asynchronous DECnet connection. This procedure assumes the local VMS node is originating the connection and switching on the terminal line for DECnet use. The connection must be to a VMS node on which you have an account with NETMBX privilege. The steps that the system manager at the remote VMS node must perform in order for the dynamic asynchronous DECnet link to be established successfully are also included in this section.

1. Log in to the SYSTEM account and enter the following commands interactively (or include them in the SYS$MANAGER:SYSTARTUP_V5.COM command procedure before you boot the system). These commands load the asynchronous driver NODRIVER (NOA0) and install DYNSWITCH software on your system.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  CONNECT NOAO/NOADAPTER
SYSGEN>  EXIT
$ INSTALL:=$SYS$SYSTEM:INSTALL
$ INSTALL/COMMAND
INSTALL> CREATE SYS$LIBRARY:DYNSWITCH/SHARE -
_ /PROTECT/HEADER/OPEN
INSTALL> EXIT
```

The system manager of the remote VMS node must also enter these commands.

Additionally, the system manager at the remote VMS node must enter the commands that follow. These commands enable the use of *virtual terminals* for the terminal line that is to be switched, and set the DISCONNECT characteristic for the terminal line. (The virtual terminal capability permits the process to continue running if the physical terminal you are using becomes disconnected.)

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  CONNECT VTAO/NOADAPTER/DRIVER=TTDRIVER
SYSGEN>  EXIT
$ SET TERMINAL/EIGHT_BIT/PERMANENT/MODEM/DIALUP -
_$ /DISCONNECT device-name:
```

Device-name is the name of the terminal port to which the dynamic asynchronous connection is made.

2. You must establish the required transmit password at the originating end of the dynamic asynchronous dialup link. The transmit password is the password sent to the remote node during connection startup. Use NCP to enter a command to define the transmit password for the remote node. The password can contain one to eight alphanumeric characters and should not contain any spaces. Specify the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP> DEFINE NODE node-id TRANSMIT PASSWORD password
NCP> EXIT
```

Node-id is the name of the remote node with which your node is forming a connection.

For each remote node with which you will create a dynamic asynchronous DECnet dialup link, you must define a transmit password in a separate command.

The system manager for the node at the other end of the connection must define that same password as a receive password for your node (the password expected to be received from your node). The remote system manager should also specify the parameter INBOUND ROUTER or INBOUND ENDNODE, to indicate the type of node (router or end node) that is expected to initiate the dynamic connection. The remote manager should enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP> DEFINE NODE node-id RECEIVE PASSWORD password INBOUND node-type
NCP> EXIT
```

3. DECnet must be running on both nodes for the remaining steps. If you have not already done so, turn on the network by entering the following command (and request that the remote system manager do so also):

```
$ @SYS$MANAGER:STARTNET
```

   If the network was already running before you began the dynamic asynchronous connection procedure, enter these commands to cause the permanent database entry to be entered in the volatile database:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET NODE node-id ALL
NCP>EXIT
```

4. The remaining steps can be performed by any VMS user with NETMBX privilege. Log in to your local VMS system and enter the following DCL command on your terminal to cause your process to function as a *terminal emulator* (which makes the remote terminal appear to be a local terminal connection):

```
$ SET HOST/DTE device-name:
```

   Device-name is the name of your local terminal port that is connected to the modem. If both systems use modems with autodial capabilities (for example, DF03, DF112 or DF224 modems that support an autodial protocol), you can optionally include the /DIAL qualifier on the SET HOST/DTE command to cause automatic dialing of the modem on the remote node, as follows:

```
$ SET HOST/DTE/DIAL=number device-name:
```

5. If you are not using automatic dialing, dial in to the remote node manually.

6. Once the dialup connection is made and you receive the remote VMS system welcome message, log in to your account on the remote node.

7. While logged in to your account on the remote node, enter the following command to cause the line to be switched to a DECnet line automatically:

```
$ SET TERMINAL/PROTOCOL=DDCMP/SWITCH=DECNET
```

   The following message indicates that the DECnet link is being established:

```
%REM-S-END - control returned to local-nodename::
$
```

   To check whether the communications link has come up, specify the following command on the local system:

```
$ RUN SYS$SYSTEM:NCP
NCP>SHOW KNOWN CIRCUITS
NCP>EXIT
```

   The resulting display should list a circuit identified by the mnemonic TT or TX, depending on the asynchronous device installed on the line, and indicate that it is in the ON state.

When the DCL prompt ($, by default) appears on your terminal screen, you can begin to communicate with the remote node over the asynchronous DECnet connection.

If the dynamic connection is not made successfully, refer to the section on asynchronous connection problems.

8. As an alternative to switching the terminal line to a DECnet line automatically (as described in the previous step), you can switch the line manually. If you originate a dynamic connection to a VMS node from a non-VMS system, manual switching is required; from a VMS system, it is optional. If you are originating the connection from a non-VMS node, follow system-specific procedures to log in to the remote VMS node by means of terminal emulation.

   Once you are logged in to the remote node, two steps are required to perform manual switching:

   a. Using your account on the remote VMS node, specify the SET TERMINAL command described in step 7, but add the /MANUAL qualifier:

      ```
      $ SET TERMINAL/PROTOCOL=DDCMP/SWITCH=DECNET/MANUAL
      ```

      You will receive the following message from the remote node indicating the remote system is switching its line to DECnet use:

      ```
      %SET-I-SWINPRG The line you are currently logged over is becoming
                     a DECnet line
      ```

   b. You should exit from the terminal emulator and switch your line manually to a DECnet line. The procedure depends on the specific operating system on which you are logged in. The following example shows how a VMS user originating a dynamic connection would perform this procedure.

      - Exit from the terminal emulator by pressing the backslash (\) key and the CTRL key simultaneously on your VMS system.
      - Enter the following command to switch your terminal line to a DECnet line manually:

        ```
        $ SET TERMINAL/PROTOCOL=DDCMP TTA0:
        ```

        TTA0 is the name of the terminal port on the local node.

      - Enter NCP commands to turn on the line and circuit connected to your terminal port TTA0 manually, as in the following example:

        ```
        $ RUN SYS$SYSTEM:NCP
        NCP>SET LINE TT-0-0 RECEIVE BUFFERS 4 LINE SPEED 2400 STATE ON
        NCP>SET CIRCUIT TT-0-0 RECEIVE BUFFERS 4 STATE ON
        NCP>EXIT
        ```

        Asynchronous DECnet is then started on the local VMS node.

9. You can terminate the dynamic asynchronous link in one of two ways:

   a. Break the telephone connection.

b. Run NCP and turn off either the asynchronous line or circuit. The two commands you can use are as follows:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LINE dev-c-u STATE OFF
NCP>SET CIRCUIT dev-c-u STATE OFF
NCP>EXIT
```

If either of the above NCP commands is entered at the remote node, the line returns to terminal mode immediately. If the command is entered at the local (originating) VMS node, the remote line and circuit remain on for approximately four minutes and then the line returns to terminal mode.

6.2.2.4 Shutting Down and Restarting the Network

The network shuts down automatically as part of the normal VMS system shutdown procedure. If your VMS system is running, you can shut down the network at your local node without destroying any active logical links by entering the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET EXECUTOR STATE SHUT
NCP>EXIT
```

When this command sequence is issued, no new links are allowed; when all existing links are disconnected, the network is turned off.

While your VMS system is running, you can stop the network at your node by entering the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET EXECUTOR STATE OFF
NCP>EXIT
```

All logical links are terminated immediately and the network is stopped.

To turn the network on manually, specify the following:

```
$ @SYS$MANAGER:STARTNET
```

To start the network if it is not currently active, you must be logged in to the SYSTEM account or have the privileges listed at the beginning of the STARTNET.COM command procedure.

To cause the network to be started each time the VMS operating system is booted, enable the following command in the SYS$MANAGER:SYSTARTUP_V5.COM command procedure:

```
$ @SYS$MANAGER:STARTNET
```

The command is supplied in the command procedure; to enable it, use a text editor to delete the exclamation point at the start of the command line. The network will be turned on automatically as part of the VMS system startup. You will not have to turn on the network again unless you should explicitly shut down the network or remove the network startup invocation from the site-specific startup command procedure.

6.2.2.5 Using NCP to Create and Tailor the Configuration Database

The system manager is responsible for configuring the node for network operation by supplying information in the DECnet–VAX configuration database about the following network components:

- The local (executor) node
- Remote nodes with which the local node can communicate
- Local circuits
- Local lines
- Network objects
- Network event logging

The configuration database is actually two databases: a permanent database that establishes the default parameter values for node startup, and a volatile database that contains the current parameter values in a functioning network.

You can use the Network Control Program (NCP) Utility to build the network configuration database manually or to modify its contents. If you are configuring the node for the first time, you can use the automatic configuration command procedure, NETCONFIG.COM, to establish parameters needed to get DECnet running. The procedure for using NETCONFIG.COM is described in an earlier section.

When you run NCP and enter a command, NCP will prompt you for selected parameters if you do not supply them. NCP also provides a HELP facility with information about each command, which you can access as follows:

```
$ RUN SYS$SYSTEM:NCP
NCP>HELP [topic...]
```

Use NCP SET commands to establish the contents of the volatile database. Use NCP DEFINE commands to establish the contents of the permanent database. You must have OPER privilege to change the volatile database and SYSPRV privilege to change the permanent database.

The permanent database information is supplied to the volatile database when the network is started (that is, the STARTNET.COM command procedure is executed). You can also use the ALL parameter with the SET command to cause all permanent database entries for a network component to be loaded into the volatile database.

The basic NCP commands required to define the network components in the permanent configuration database are as follows:

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE EXECUTOR
NCP>DEFINE NODE node-id
NCP>DEFINE CIRCUIT circuit-id
NCP>DEFINE LINE line-id
NCP>DEFINE OBJECT object-name
NCP>DEFINE LOGGING MONITOR STATE ON
NCP>DEFINE LOGGING MONITOR EVENTS event-list
NCP>EXIT
```

NCP commands also recognize the plural forms of the network component names: KNOWN NODES, KNOWN CIRCUITS, KNOWN LINES, KNOWN OBJECTS.

To modify the current configuration of your node, you can enter SET commands for any network component. For example, to add circuit and line entries for the Ethernet UNA device (the DEUNA), enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LINE UNA-0 STATE ON
NCP>SET CIRCUIT UNA-0 STATE ON
NCP>EXIT
```

To determine the contents of your network configuration database, use the NCP commands LIST and SHOW. The LIST command displays information in the permanent database; the SHOW command displays volatile database entries. To delete entries from the configuration database, use the PURGE and CLEAR commands. The PURGE command deletes permanent database entries; the CLEAR command deletes or resets volatile database entries.

For example, to list the permanent name and address of a node, enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>LIST NODE node-id
NCP>EXIT
```

To delete a node from the permanent database, enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>PURGE NODE node-id ALL
NCP>EXIT
```

Node-id can be either the node name or the node address. You can also delete an individual parameter for a node.

Because the PURGE command does not affect the volatile (memory-resident) copy of the DECnet database, you can access a node deleted with the PURGE command until DECnet is started again. If you use the CLEAR command to delete a node entry, the node entry will reappear in the volatile database after DECnet is started again.

6.2.2.6 Providing Security for Your DECnet-VAX Node

Some of the security measures that you can use to protect your files and system in a network environment are summarized in this section.

As manager of a VMS node, you can protect your system against unauthorized access by users on other nodes in the network by setting passwords for any accounts that you may create. Otherwise, users on other nodes could gain full access to your system by using the SET HOST command to log in to one of the accounts on your node.

Protecting Files and Using Proxy Accounts

As a user on a VMS node, you can protect the files in your directory against access over the network. To set limits on who can access the files in your account, specify the DCL command SET PROTECTION. If your file is protected, a VMS user on a remote node who wants to access your file must be able to specify the user name and password of a local account that has the appropriate privileges to access the file. A remote user to whom you have given this information must then include the authorization information in the form of an access control string, *"username password"*, in the VMS file specification used to access your file:

```
node"username password"::device:[directory]filename.type;version
```

Establishing proxy accounts. As system manager of your node, you can maintain the security of passwords by preventing their transmission over the network. You can permit selected outside users to access particular non-privileged accounts on your node without having to send any explicit access control information over the network. To do this, you must create a proxy account that allows a remote user to have access privileges on your node without having a private account on your node. If the remote user is assigned a proxy account on your local node that maps into a local user account, the remote user assumes the same access privileges as the owner of the local account.

The system manager controls the use of proxy accounts at the local node. Use the Authorize Utility to create and modify the permanent proxy database, NETPROXY.DAT, at your node. Each NETPROXY.DAT entry can map a single remote user to multiple proxy accounts on the local node (one default proxy account and up to 15 additional proxy accounts). The proxy database entry identifies the user by *nodename::username* or *nodename::(group,member)*.

When DECnet is started up, the information in NETPROXY.DAT is used to construct a volatile proxy database. If changes are made to the permanent proxy database by means of the Authorize Utility, the volatile proxy database is updated automatically.

Similarly, the system manager at a remote node can create and maintain a proxy database of network users having proxy access to specific accounts on that node.

Controlling proxy login access. For proxy login to be successful, one node must be able to initiate proxy login access and the other node must allow proxy login access. To control proxy login for your local (executor) node, use Network Control Program commands to modify the proxy parameters in the executor and object databases. The NCP parameters that specify whether a node can initiate proxy login are the outgoing proxy parameters; the parameters that specify whether a node allows proxy login access are the incoming proxy parameters. By default, both the local node and the remote node can initiate proxy login and allow proxy access.

Defaults for DIGITAL-supplied objects are set in the object database. For example, the object MAIL has outgoing proxy access set by default. To specify or modify the proxy parameter for a network object, use the NCP command SET OBJECT. Use this command to permit outgoing proxy access for a network object:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET OBJECT object-name PROXY OUTGOING
NCP>EXIT
```

Controlling Access to Your Node

In general, the system manager can control access to the local node at three levels:

- **Circuit-level access control:** For point-to-point connections, especially over dialup lines, you can use passwords to verify that the initiating node is authorized to form a connection with your node. Passwords are usually optional for point-to-point connections but are required for dynamic asynchronous connections.

  Each end of a point-to-point circuit can establish a password to transmit to the other node, and specify a password expected from the other node. Before the link is established, each node verifies that it received the expected password from the other node.

  Added security is provided for a dynamic asynchronous connection (which is normally maintained only for the duration of a telephone call): the node requesting the dynamic connection is required to supply a password, but the node receiving the login request is prevented from revealing a password to the requesting node.

- **Node-level access control:** To control the establishment of logical links with remote nodes, you can specify in your network database access control parameters that indicate which of the following logical link connections are permitted: INCOMING, OUTGOING, BOTH, or NONE. Use the NCP commands that follow to specify access parameters for a specific node, and the executor parameter DEFAULT ACCESS that applies to any node for which a specific access parameter is not specified:

  ```
  $ RUN SYS$SYSTEM:NCP
  NCP>DEFINE NODE node-id ACCESS option
  NCP>DEFINE EXECUTOR DEFAULT ACCESS option
  NCP>EXIT
  ```

- **System-level access control:** When a remote user requests access to the system, the following means of authorization are checked:
  - Is an explicit access control string available?
  - Does the user have a proxy account on the local node?
  - Is there a default nonprivileged DECnet account at the local node?

  If no explicit access control information or proxy account is available, DECnet–VAX will attempt to use a default nonprivileged DECnet account to access the system. The default DECnet account allows users to perform certain network operations, such as the exchange of electronic mail between users on different nodes, without having to supply a name and password. The default DECnet account is also used for file operations when an access control string is not supplied. For example, it permits remote users to access local files on which the file protection has been set to allow WORLD access. If you do not want remote users accessing your node, do not create a default DECnet account.

  You can request the DECnet–VAX configuration command procedure, NETCONFIG.COM, to establish the default nonprivileged DECnet account and directory for you automatically or you can establish the account and directory manually, as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF>ADD NETNONPRIV/PASSWORD=NONPRIV/DEVICE=device-name:-
_ /DIRECTORY=[NETUSER]/UIC=[200,200]/PRIVILEGE=(TMPMBX,NETMBX)-
_ /FLAGS=(CAPTIVE)/NOBATCH/NOINTERACTIVE/LGICMD=NL:
UAF>EXIT
$ CREATE/DIRECTORY device-name:[NETUSER]/OWNER_UIC=[200,200]
```

  Device-name is the name of the device on which you have your directory.

  If a remote node requests access to an object on the local node but does not supply access control information, any access control information specified for the object in the local network database will be used.

6.3 Keeping the Network Running

Once you have brought up your system as a network node, you can use a variety of software techniques to monitor and test the network. You can also use troubleshooting techniques to resolve problems related to keeping the network running. The tools you can use to monitor the network and the types of tests you can perform on the network are summarized in the following sections. Common problems encountered during network operation are indicated, along with advice on troubleshooting.

6.3.1 Monitoring the Network

You can monitor network activity using software tools. Analyzing the information you collect can help you to determine whether the network is running properly or whether any changes are required to resolve problems or improve performance. Major network monitoring tools include the following:

- NCP display commands you can use to determine the status and characteristics of components in the network.
- NCP counters you can read to obtain error and performance statistics on current network operations.
- Network events logged by DECnet that can be reported to you as they happen.
- Other software tools, such as the Ethernet configurator and the DECnet Test Sender/DECnet Test Receiver (DTS/DTR) Utility, that permit you to learn more about network operation.

6.3.1.1 Using NCP to Display Information About Network Components

You can use the NCP commands SHOW and LIST to monitor network activity by displaying the following:

- Information about the current condition of network components (using SHOW commands) and the startup values assigned to the components (using LIST commands)
- Counter information about circuits, lines, remote nodes, and the local node (using SHOW COUNTER commands)
- Information about the range of network events being logged by the DECnet event logging facility (using SHOW LOGGING commands)

You do not need any privileges to issue SHOW commands, but you need the privilege SYSPRV to issue LIST commands.

Use the SHOW command to monitor the operation of the running network. You can display the characteristics and current status of network circuits, lines and nodes, including the local (executor) node. This information is useful in detecting any changes in the network configuration or operation. For example, if a circuit failure causes some nodes to become unreachable, you can use SHOW commands to check the status of the circuit and the nodes.

In general, the SHOW and LIST commands permit you to indicate what type of information you want NCP to display about the particular component you specify. The display types include the following:

- CHARACTERISTICS—Static information that does not normally change during network operations (for example, the identification of the local node and the circuits connected to the local node, and relevant routing parameters such as circuit cost).

- STATUS—Dynamic information that usually indicates network operation for the running network (for example, the operational state of the local node, circuits, lines and remote nodes).
- SUMMARY—Only the most useful information from both static and dynamic sources; usually a condensed list of information provided for the CHARACTERISTICS and STATUS display types. SUMMARY is the default if you do not specify a display type.
- COUNTERS—Counter information about circuits, lines, remote nodes, and the local node.
- EVENTS—Information about which network events are currently being logged to which logging collection point.

When you display information about network components, you can specify either the singular or plural form of the component in the NCP command. Plural forms of component names are KNOWN (all components available to the local node), ACTIVE (all circuits, lines and logging not in the OFF state), and ADJACENT (all nodes directly connected to the local node).

Typical examples of NCP display commands follow:

```
$ RUN SYS$SYSTEM:NCP
NCP>SHOW EXECUTOR CHARACTERISTICS
NCP>SHOW KNOWN LINES STATUS
NCP>SHOW ACTIVE CIRCUITS
NCP>SHOW ADJACENT NODES STATUS
NCP>LIST KNOWN NODES
NCP>EXIT
```

All NCP display commands optionally allow you to direct the information displayed to an output file you specify.

You can display information about network components on remote nodes using the TELL prefix in the NCP command. The format of the command is TELL *node-id* SHOW *component*. For example, to look at remote node counters, enter the following command sequence:

```
$ RUN SYS$SYSTEM:NCP
NCP>TELL node-id SHOW EXECUTOR COUNTERS
NCP>EXIT
```

6.3.1.2 Using NCP Counters

You can use NCP commands to display error and performance statistics about network components at any time while the network is running. DECnet software uses counters to collect statistics for the executor node, remote nodes, circuits and lines automatically. To display the contents of counters, use NCP SHOW COUNTER commands, as in the following typical examples of the commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SHOW EXECUTOR COUNTERS
NCP>SHOW NODE node-id COUNTERS
NCP>SHOW KNOWN CIRCUITS COUNTERS
NCP>SHOW KNOWN LINES COUNTERS
NCP>SHOW LINE line-id COUNTERS
NCP>EXIT
```

For the local node and remote nodes, counter statistics cover such subjects as connection requests, user data traffic, timeouts, and errors. Circuit counters cover such topics as the transmission of data packets over the circuit, timeouts, and errors. Line counters cover such information as the transmission of bytes and data blocks over the line and relevant errors.

Use NCP commands to control counter usage. You may want to reset counters to zero if you are establishing a controlled environment for test purposes. To reset counters to zero, use the NCP command ZERO COUNTERS (the ZERO command requires the OPER privilege). You can zero counters for the executor node and individual nodes, circuits and lines, or all nodes, circuits and lines. In the examples of typical commands that follow, note that the word COUNTERS is optional:

```
$ RUN SYS$SYSTEM:NCP
NCP>ZERO EXECUTOR COUNTERS
NCP>ZERO NODE node-id
NCP>ZERO KNOWN CIRCUITS
NCP>ZERO LINE line-id COUNTERS
NCP>EXIT
```

You can regulate the frequency with which specific counters are logged by entering the following command sequence:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET component COUNTER TIMER nn
NCP>EXIT
```

The variable *nn* is in seconds. Expiration of the counter timer causes the contents of the counter to be logged and the counter reset to zero.

6.3.1.3 Using DECnet Event Logging

Use the DECnet event logging facility to monitor significant network events, such as circuit failures or lost packets, on a continuous basis. Whenever a network error or other meaningful event occurs, the DECnet event logger will log an event message to a terminal or file that you specify. Examples of network events that are logged as they happen include the following:

- Changes in circuit and line states (for example, a circuit failure)
- A node becoming reachable or unreachable
- Circuit and node counter values, logged before the counter is automatically reset to zero
- Errors in data transmission
- Use of invalid data link passwords

Collection and analysis of network events can provide insight into why a particular error condition exists or why network performance may vary.

As events are detected, the event logger sends them to a collection point for analysis. Collection points are called *logging sinks;* they can be located on the local node or at a remote node. Event data can go to one or more sinks. Each of the following types of event sinks handles event data in a slightly different way:

- **Logging monitor.** A program that receives and processes events. Events sent to the logging monitor are displayed on the screen of any terminal declaring itself a "network operator" by means of the Operator Communication (OPCOM) facility. Directing events to the OPCOM terminal is very useful for applications where the operator needs to know what is happening on the network as it happens. For example, it may be useful to know that a circuit is going down as it happens.

  The automatic configuration command procedure enables the logging monitor. The OPCOM process is started when the command procedure SYS$MANAGER:SYSTARTUP_V5.COM is executed. You can enable a terminal as a network operator terminal by specifying the DCL command REPLY /ENABLE=NETWORK. Usually the operator console (OPA0) is one of the OPCOM terminals.

- **Logging console.** A terminal or file that receives events in a readable format. The default logging console is the operator console.

- **Logging file.** A user-specified file that receives events in binary format, possibly for later analysis.

In order for logging to occur at your node, logging must be enabled and the events to be logged must be identified. If you use the automatic configuration command procedure, NETCONFIG.COM, logging will be established automatically. Otherwise, you can use the NCP command SET or DEFINE LOGGING to set the logging sink state to be ON. To identify a remote location for a logging sink, specify the SINK *node-id* parameter in the command. Use one or more separate commands to define

the events to be logged. For example, enter the following commands to cause all network events to be logged to OPCOM and displayed at your network operator terminal:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LOGGING MONITOR STATE ON
NCP>SET LOGGING MONITOR KNOWN EVENTS
NCP>EXIT
```

Alternatively, for each event class, you can specify the specific events to be logged, as follows:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET KNOWN LOGGING EVENTS event-list
NCP>EXIT
```

Events in the event list are identified by class and type (in the form *class.type*). An event *class* refers to the DECnet software functional layer in which the event occurred. Event classes logged by DECnet include those listed in Table 6–2. The event *type* is a decimal number representing a unique event within the class. You can use the asterisk (*) wildcard character for event types, and you can specify a single event type or a range of types.

Table 6–2: DECnet Event Classes

Event Class	DECnet Functional Layer
0	Network Management
1	Application
2	Session Control
3	End Communication
4	Routing
5	Data Link
6	Physical Link
7	X.25 packet-level events
128–159	VMS system-specific

An example of the command to specify event types 5 through 7 in event class 4 is as follows:

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE LOGGING MONITOR EVENTS 4.5-7
NCP>EXIT
```

The event message displayed by OPCOM is in the following form:

```
EVENT TYPE  class.type, event-text
From node-address (node name) Occurred (date and time)
component type and identifier
descriptive text
```

You can use the SHOW LOGGING command to learn what logging is being performed. For example, to display complete information on all logging being conducted at all nodes, use these commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SHOW ACTIVE LOGGING KNOWN SINKS
NCP>EXIT
```

To stop monitoring at the network operator terminal temporarily, enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LOGGING MONITOR STATE OFF
NCP>CLEAR LOGGING MONITOR ALL
NCP>EXIT
```

Then enter these commands to turn monitoring back on:

```
$ RUN SYS$SYSTEM:NCP
NCP>SET LOGGING MONITOR STATE ON
NCP>EXIT
```

To disable logging at the network operator terminal permanently, enter the following commands:

```
$ RUN SYS$SYSTEM:NCP
NCP>PURGE LOGGING MONITOR ALL
NCP>EXIT
```

6.3.2 Common Problems Encountered on the Network

Once you have brought up your system as a network node, you may receive messages related to networking errors. Other problems that can occur at any time during network operation may not result in messages being displayed. This section explains the causes of error messages that may be displayed.

6.3.2.1 Common Error Messages and Meanings

When you are using DECnet–VAX, you may receive network-related messages indicating software or hardware problems, transient conditions, or errors in your input. The following list displays some common network-related messages, explains what condition may be causing each message, and suggests actions you can take.

- **NCP-I-INVPVA, invalid parameter value**

  This message is displayed if you specify a parameter value in an NCP command that is not a valid value for the specified parameter. The name of the parameter for which the value was invalid is displayed at the end of the error message. Re-enter the command with the correct value for the parameter.

- **SYSTEM-I-LINKEXIT, network partner exited**

  This message is displayed if the process on the remote node exited before confirming the logical link to your node. The remote process might have exited prematurely, a timeout may have occurred at the remote node, or there may be a problem in the log file on the remote node. You could either retry the operation or try to read the NETSERVER.LOG file in the directory of the account you are attempting to access at the remote node. (DECnet–VAX automatically creates a NETSERVER.LOG file and places it in the directory for the appropriate account when it receives a connect request.)

- **SYSTEM-F-UNREACHABLE, remote node is not currently reachable**

  This message is displayed when you attempt to connect to a node that is unreachable. You can try to access the remote node again at a later time.

  The message is also displayed even if the remote node does not exist, as long as you have indicated a node address or a node name that you previously defined in your node database.

  You also receive notice that the node is unreachable if the value of the executor parameter MAXIMUM ADDRESS in your network database is lower than the address of the remote node you are attempting to access. Increase the value of the NCP executor parameter MAXIMUM ADDRESS in your database to be at least as high as the highest address of any node that you want to contact.

- **SYSTEM-F-INVLOGIN, login information invalid at remote node**

  This message is displayed if you attempt to access a remote node using an access control string that contains an invalid user name or password, or if you do not specify any access control information and no default DECnet account or proxy account is available at the remote node. Retry the file operation with the correct login information.

- **SYSTEM-F-NOSUCHNODE, remote node is unknown**

  This message is displayed if you attempt to enter a command to access a remote node and the remote node represented by **node-id** is not identified in the local volatile database. Verify that the node identifier is correct, enter the node name in your node database, and retry the operation.

- **SYSTEM-F-PATHLOST, path to network partner lost**

  This message is displayed if you logged in to another node over the network (for example, using the DCL command SET HOST) and the path to the remote node is lost.

  The path may be lost because of too much network activity or communications problems, or because DECnet was turned off at the remote node. Wait, then check to see if the node is still reachable. If so, try again to log in.

- **SYSTEM-F-SHUT, remote node no longer accepting connects**

  This message is displayed if you attempt to access the remote node using a DCL command (such as the SET HOST command) under either of these conditions:

  a. The executor parameter DEFAULT ACCESS on the remote node has been set to NONE. The default access at the remote node must be set to permit incoming and outgoing access before you can connect to the node.

  b. The command SET EXECUTOR STATE SHUT was executed on the remote system. The network must be restarted on the remote node.

- **SYSTEM-F-NOLINKS, maximum network logical links exceeded**

  This message is displayed if the maximum number of links that the remote node allows has been exceeded. Wait and try again later.

- **SYSTEM-F-NOSUCHOBJ, network object unknown at remote node**

  This message is displayed if you attempt to access a network object at a remote node and the object is not specified in the remote node database. For example, if you attempt to use the Phone Utility to reach a node that does not have an entry for the network object PHONE in its configuration database, you receive the above message.

6.3.2.2 Problems Related to Network Operation

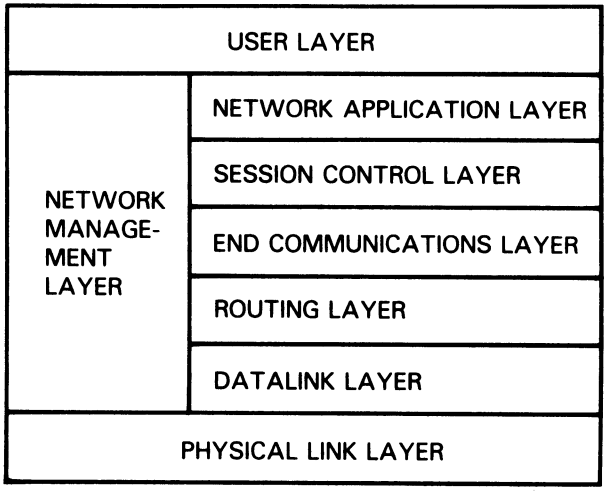
Problems in maintaining the proper functioning of the running network can be difficult to resolve. This section describes the technique for isolating a problem to a particular DECnet software functional layer or layers, and provides troubleshooting suggestions to determine the specific network problem. As system manager of the local node, you may want to consult with the network manager (if one is available for your network) as necessary to resolve these problems.

Troubleshooting Techniques Based on DNA Layers

Techniques for troubleshooting DECnet–VAX problems are based on the layered network design of DECnet–VAX as specified in the DIGITAL Network Architecture (DNA). The DNA layers are illustrated in Figure 6–1. Each layer performs particular services as part of the overall network capability provided at the node.

During troubleshooting, it is useful to distinguish among the network layers in localizing the cause of a particular problem. For example, some problems are characteristic of the Data Link layer, while others are related to the Routing layer or to the End Communications layer (which provides logical link services).

Figure 6-1: DECnet-VAX Software Design as Based on DNA Layers



Network Problems and Suggested Actions

The following discussion of network difficulties identifies typical problems originating at the various layers, and it describes actions you can take to locate the source of the problem. The problems are grouped into those related to data links, routing, and logical links.

Data link problems. Inability to reach an active node is a common problem on the network. The problem could be either a data link problem or a routing problem.

To determine whether the problem is a data link problem, examine both the remote node and the circuit. The data link layer causes data to be moved over physical devices, so it affects only adjacent nodes (an adjacent node is connected to the local node by a single physical line). You can learn whether the unreachable node is an adjacent node and whether the node is available by checking with the network manager or the system manager of the unreachable node.

Also check the state of the circuits (the data link protocol causes a circuit to be in the ON-SYNCHRONIZING state). The problem may be with the data link if the circuit does not start up correctly or is up but the adjacent node is not reachable. (Note that circuit startup may also be affected by incorrect setting of the transmit and receive passwords, as described in the following section on routing problems.)

To locate a data link problem, examine the appropriate counters, line and circuit parameters, and network events.

- Use NCP SHOW commands to display the contents of the circuit and line counters to see if they are reporting errors.
- Use NCP SHOW commands to check the values of line and circuit parameters in the network configuration database.
- Then look at the network events DECnet logged for event class 4 (for the Routing layer) and event class 5 (for the Data Link layer) to determine whether any events affecting the data link have occurred.

Routing problems. Routing layer problems can involve nodes that are not reachable or circuits that are not stable. The circuit may be up and the adjacent node may be reachable, but one or more intermediate nodes (along the communications path) that should be reachable are not.

To isolate such Routing layer problems, examine the appropriate counters and passwords, and try to check the nodes along the routing path.

- Check the contents of the node and circuit counters at your node and, if possible, arrange for the node and circuit counters at the remote node to be examined.
- Examine network events logged for event class 4 (for the Routing layer).
- Check the settings of the transmit and receive passwords for the local node and the adjacent node to see if they match (these passwords affect circuit startup).
- Finally, you can use NCP commands with the TELL prefix to try to trace the routing path from one node to another, to determine if an intermediate node is down and to examine the parameter values for all nodes on the communications path.

If erratic routing behavior occurs (for example, constant changes in the reachability of nodes, or connection to a node other than the one you expect to reach), check whether two or more nodes with the same node address are connected to the network. If routing seems to be functioning properly, you can look at executor parameters related to routing (such as cost and hops).

Logical link problems. The End Communications layer, which provides logical link services, can also be the source of common problems. Typical symptoms of logical link problems include

- Link timeout
- Network partner exited
- Invalid account
- Problems with performance and response time

In general, for logical link problems, you can examine the following:

- The default DECnet nonprivileged account and directory on the remote node, to determine if they have been created properly.
- Incoming and outgoing timers at both ends of the logical link, to ensure the links are not timing out prematurely because the timers are set too low.
- The accounting log (using the VMS Accounting Utility), to determine whether the correct process was created or whether a correct process exited prematurely.
- The load on the local and remote nodes, to determine whether the load is preventing the link from being created.
- The path over the network to the remote node. If the circuit is an Ethernet circuit, check the line buffer size parameter to ensure the proper setting.
- The Netserver timeouts, by getting someone to examine the NETSERVER.LOG file at the remote node.
- The proxy settings for your node and for the objects being accessed. (To determine the default proxy access setting for your executor node, specify the NCP command SHOW EXECUTOR CHARACTERISTICS. To examine the proxy access setting for network objects, use the NCP command SHOW KNOWN OBJECTS CHARACTERISTICS.)
- The disk quota, to ensure it is sufficient to create the NETSERVER.LOG file.
- The SYS$LOGIN file, to determine whether the file protection is set to WORLD:READ.

If a logical link connection is unsuccessful, the link may have timed out for one of the following reasons:

- A heavily loaded node can cause creation of a logical link to take a long time.
- Incoming and outgoing timers may be set too low.

To prevent link timeouts, you can reset the executor parameters INCOMING TIMER and OUTGOING TIMER to higher values at both nodes.

A logical link problem may cause the message "network partner exited" to be displayed. This message indicates that the remote node exited before the logical link was established. Check the following:

- The networking load on the nodes at each end of the logical connection
- The accounting log on the remote node
- Netserver timeouts on the remote node

If you receive a message indicating an invalid account, check that you have the proper authorization to make the logical link connection. However, an invalid account condition may also be reported by the message "network partner exited." Consequently you should try to to have someone check the NETSERVER.LOG file on the remote node.

If performance and response time over the logical link become degraded, the cause may be too much traffic on a path to the target node. If you encounter this problem, consult with the network manager.

Configuration problems. The main reason for network errors may be improper configuration of the system. Check your DECnet–VAX configuration, and check the communications cables and connections.

6.3.2.3 Asynchronous Connection Problems

Attempts to establish asynchronous DECnet connections with other nodes can fail for a variety of reasons. This section describes some reasons why you may fail to make a static or dynamic connection.

A static asychronous connection has failed if the static asynchronous DECnet line is started but remains in the ON-STARTING state. To isolate the cause of the problem, check whether the following conditions exist:

- Are the line speeds at both ends of the connection set to the same value?
- If you are using a dialup line, is the modem characteristic set on the terminal? (This must be done before the line is set to asynchronous DDCMP use.)
- Are the two nodes being connected located in the same area in the network (that is, do their node addresses have the same area number) or are both nodes area routers?
- Is the parity on the asynchronous DECnet line set to NONE? If your system is not a VMS system, is the terminal line set to the correct parity?
- Is the terminal line set up to use 8-bit characters?
- If the node already has an active circuit, is the node a routing node?
- If verification is enabled for the circuit, do the passwords set at the two nodes match?

If you are unsuccessful in setting up your terminal line as a static asynchronous DDCMP line, check the following:

- Is the /NOTYPE_AHEAD qualifier specified for your terminal before you attempt to set up the static asynchronous line? If a type-ahead buffer is associated with your terminal, you may not be able to bring up your terminal line as an asynchronous DECnet line until you terminate any process started at the remote node that may own your terminal line.

If dynamic switching is being performed and the asynchronous DECnet connection is not made, first check the following:

- Is DECnet started on both nodes?
- Is the asynchronous DDCMP class driver (NODRIVER) loaded by means of SYS$SYSTEM:SYSGEN at each node?
- Is the dynamic switch image (DYNSWITCH) installed by means of SYS$SYSTEM:INSTALL at each node?
- Are virtual terminals enabled on the remote node and, in particular, for the terminal over which you are logged in to the remote node?

If the dynamic asynchronous lines are started but are left in the "ON—STARTING" state, make the following checks:

- Are the two nodes that are being connected located in the same area (that is, do their addresses have the same area number) or are they both area routers?
- Are the routing initialization passwords (transmit and receive passwords) set appropriately at each node?
- Is the INBOUND parameter for the initiating node set correctly in the node database at the node receiving the connection request?
- Is the parity on the asynchronous DECnet line set to NONE? If your system is not a VMS system, is the terminal line set to the correct parity?
- Is the terminal line at the remote node set up to use 8-bit characters?
- If the node already has an active circuit, is the node defined as a routing node?

Chapter 7

Setting Up a Local Area VAXcluster Environment

This chapter discusses how to set up a small local area VAXcluster configuration. For the purposes of this manual, a small local area VAXcluster configuration consists of one processor called a *boot server* that serves as the hub of the cluster, and one or more MicroVAX or VAXstation processors that are connected to the boot node. If you want to learn how to set up this type of VAXcluster configuration, then you should read the rest of this chapter. If you manage a VAXcluster environment other than the type described in this chapter, then you should refer to the *VMS VAXcluster Manual.*

7.1 What Is a Cluster?

A *cluster* is a group of two or more processors that share some or all of their resources. When a group of VAX processors shares resources in a VAXcluster environment, the storage and computing resources of all of the processors are combined, which can increase the processing capability, communications, and availability of your computing system.

7.1.1 VAXcluster Types

Three types of VAXcluster configurations are possible:

- Local Area VAXcluster configuration
- CI-only VAXcluster configuration
- Mixed-interconnect VAXcluster configuration

Local Area VAXcluster Configuration

A Local Area VAXcluster configuration is made up of a single VAX processor that serves as the management center of the cluster, plus one or more VAX processors that are connected to this hub. A local area VAXcluster configuration always includes the following parts:

- **A Boot Server**

  A boot server is a VAX or MicroVAX processor, and it serves as the management center of a local area VAXcluster environment. The system disk of the boot server contains management files for the entire cluster, including startup files, the boot server's system disk, user authorization information, and the capability of letting other processors join the cluster. The boot server must be available and running for the cluster to operate.

  Boot servers should be the most powerful machines in the cluster. They should also use the highest bandwidth Ethernet adapters available. You can use any VAX or MicroVAX system except VAX-11/725, VAX-11/730, or MicroVAX I as a boot server.

  (Note that if your boot server is a MicroVAX II class system with an RD54 system disk, you can have a maximum of three satellite nodes in your VAXcluster configuration. In either of these cases, DIGITAL recommends that the satellites use local RD series disks for paging and swapping. Refer to the VAXcluster Software Product Description for complete information about supported configurations.)

- **Satellite Nodes**

  A satellite node is a MicroVAX processor that is a member of the cluster. A processor becomes a satellite node when the CLUSTER_CONFIG.COM procedure is executed from the boot server to add the processor to the cluster.

  You can use any of the following as satellite nodes:

  - MicroVAX II or MicroVAX 2000 systems
  - VAXstation II or VAXstation 2000 systems
  - MicroVAX 3000 series systems

CI-only VAXcluster Configuration

A CI-only VAXcluster configuration is a cluster in which only the computer interconnect is used for communications between the processors in the cluster. In a CI-only VAXcluster configuration, the star coupler is used as the common connection point for all nodes in the cluster, including both VAX processors and Hierarchical Storage Controllers (HSCs).

Nodes in a CI-only VAXcluster configuration can be either

- VAX processors listed in the VAXcluster SPD, or
- HSCs

Mixed-Interconnect VAXcluster Configuration

A *mixed-interconnect* cluster may include both CI-connected VAX processors and MicroVAX systems.

This chapter concentrates on setting up a Local Area VAXcluster configuration with a single boot server. Although some of the management tasks for other VAXcluster types are similar, you should refer to the VAXcluster documentation that is available in the full VMS documentation set for information about managing a CI-only or mixed-interconnect cluster.

7.2 Shared Resources

A major benefit of a VAXcluster configuration is the ability to share resources. A *shared resource* is a resource (such as a disk or a queue) that can be accessed and used by any node in a cluster. Data files, application programs, printers, are just a few of the items that can be accessed by users on a cluster with shared resources, without regard to the particular node on which the files or program or printer might physically reside.

When disks are set up as shared resources in a VAXcluster environment, users have the same environment (password, privileges, access to default login disks, and so on) regardless of the node that is used for logging in. You can realize a more efficient use of mass storage with shared disks, because the information on any device can be used by more than one node—the information does not have to be rewritten in many places.

Print and batch queues can also be set up as shared resources. In a VAXcluster configuration with shared print and batch queues, a single job controller queue file manages the queues for all nodes on the cluster. The job controller file makes the queues available from any node. For example, suppose your VAXcluster configuration has fully shared resources and includes nodes ALBANY, BASEL, and CAIRO. A user logged in to node ALBANY can send a file that physically resides on node BASEL to a printer that is physically connected to node CAIRO, and the user never has to specify (or even know) the nodes for either the file or the printer. For more information about setting up and using print and batch queues in a VAXcluster environment, see Chapter 5.

7.3 Preparing a System for a Local Area VAXcluster Environment

In a VAXcluster environment with a single system disk, you need to install the VMS operating system only once, regardless of the number of nodes in the cluster.

To install the operating system, follow the instructions in your processor's installation guide. Before beginning the installation procedure, you must determine the configuration type for your cluster (CI-only, local area, or mixed-interconnect). During the installation of the operating system, you will be asked a series of questions. Table 7–1 lists the questions and answers for Local Area VAXcluster configurations.

NOTE: While rebooting at the end of the installation procedure, the system displays messages warning that you must install required licenses. Be sure to install these licenses, as well as the DECnet–VAX license, as soon as the system is available. Procedures for installing the licenses are described in the release notes distributed with the software kit.

Table 7–1: Installation Questions for Local Area VAXcluster Configurations

Question	Response
Will this node be a cluster member (Y/N)?	Enter Y.
What is the node's DECnet node name?	Enter DECnet node name—for example, ALBANY. The DECnet node name may be from 1 to 6 alphanumeric characters in length and may not include dollar signs or underscores.
What is the node's DECnet node address?	Enter DECnet node address—for example, 2.2.
Will the Ethernet be used for cluster communications (Y/N)?	Enter Y. The Ethernet is required for cluster (SCS internode) communications in local area configurations.
Enter this cluster's group number:	Enter a number in the range from 1 to 4095 or 61440 to 65535.
Enter this cluster's password:	Enter the cluster password. The password must be from 1 to 31 alphanumeric characters in length and may include dollar signs and underscores.
Reenter this cluster's password for verification:	Reenter the password.
Will ALBANY be a disk server (Y/N)?	Enter Y. In local area configurations, the system disk is always served to the cluster.
Will ALBANY serve HSC disks (Y/N)?	Enter N.
Enter a value for ALBANY's ALLOCLASS parameter:	Enter a value of 0 for Local Area VAXcluster configurations covered by this manual.
Does this cluster contain a quorum disk [N]?	Enter N for Local Area VAXcluster configurations covered by this manual.

7.3.1 Building a VAXcluster Configuration

Once you have installed the VMS operating system, you can start to build your cluster. This section describes how to build a simple Local Area VAXcluster configuration using the command procedure SYS$MANAGER:CLUSTER_CONFIG.COM. If you find that your cluster configuration is more complex than the type described in this manual, be sure to consult the *VMS VAXcluster Manual.*

The command procedure CLUSTER_CONFIG.COM is the primary tool that you use for adding a node to your VAXcluster configuration, removing a node from the cluster, or changing the characteristics of a node. This section describes how to use CLUSTER_CONFIG.COM to add or remove a satellite node in a Local Area VAXcluster configuration.

7.3.1.1 Using CLUSTER_CONFIG.COM

Before using CLUSTER_CONFIG.COM, log in to the SYSTEM account on the system that will be your boot server and make sure that DECnet–VAX is up and running. Be sure that your default is set to SYS$MANAGER; then enter the following command:

```
$ @CLUSTER_CONFIG
```

7.3.1.2 Setting Up the Boot Server

The first step in setting up your Local Area cluster for the first time is to establish the boot server. You must establish the local system as a boot server before you can add any satellites to the cluster.

To establish a node as a boot server, run CLUSTER_CONFIG.COM and select the CHANGE option from the menu. Then, select the option to enable the local system as a boot server.

7.3.1.3 Adding Satellite Nodes

To add satellites nodes to your Local Area VAXcluster configuration, you use the ADD option from the CLUSTER_CONFIG.COM menu. When you execute CLUSTER_CONFIG.COM to add a satellite node, you will be asked a series of questions for which the command procedure will supply most default values. For Local Area clusters that are the subject of this manual, the default values are sufficient. (If your cluster has special requirements and you want to learn more about values other than the defaults, you should consult the VAXcluster documentation in the extended VMS documentation set.)

There are some values that you must supply. These include the following:

- DECnet node name and node address for each satellite—The node name has up to 6 alphanumeric characters. The node address should be supplied by your network manager.
- Satellite's Ethernet hardware address—The Ethernet hardware address has the form xx-xx-xx-xx-xx-xx. You must include the hyphens when specifying the hardware address.

To obtain the Ethernet hardware address for MicroVAX II and VAXstation II satellites, enter the following commands at the satellite's console:

```
>>> B/100 XQ

Bootfile: READ_ADDR
```

For MicroVAX 2000 and VAXstation 2000 satellites, enter the following commands at successive console-mode prompts:

```
>>> T 53
2 ?>>>  3
>>> B/100 ES
Bootfile:  READ_ADDR
```

(In this example, if the second prompt appears as 3 ?> > > , press RETURN.)

For 3xxx series satellites, enter the following command at the satellite's console:

```
>>> SHOW ETHERNET
```

- Workstation windowing system—The windowing system (for example, UIS) if your satellite is a workstation.

7.4 DECnet–VAX Connections

In any cluster configuration, DECnet–VAX connections are required for all processor nodes. Use of DECnet–VAX facilities ensures that cluster managers can access each node in the cluster from a single terminal, even if terminal-switching facilities are not available.

In local area clusters, DECnet is required both for system management functions and interprocessor communication. For example, DECnet is used for remote booting operations (downline loading of satellite nodes).

In these configurations, DECnet and System Communication Services coexist on the same Ethernet. They share the same data link and physical link protocols, which are implemented by the Ethernet data link drivers, the Ethernet adapters, and the Ethernet itself.

Chapter 8

BACKUP Procedures

By duplicating files, the Backup Utility (BACKUP) protects data from loss or corruption. If a file is accidentally deleted or a disk becomes corrupted, and the data has been backed up regularly, you can use BACKUP copies to restore files or to re-create the contents of a disk volume or volume set.

This chapter describes *online* BACKUP, which runs under the control of the VMS operating system. *Standalone* BACKUP, which is bootstrapped into main memory and does not require the direction of the operating system, is described in your VAX processor installation and operations guide.

By saving files on a regular basis using BACKUP, you will be able to restore files if a disk is damaged or if files are deleted accidentally. The sensitivity of the data and the frequency of modifications determine how frequently you should complete BACKUP tasks, as well as the types of BACKUP tasks you should perform.

You enter BACKUP commands at the DCL command level. For most BACKUP tasks, the command line includes an *input specifier* and an *output specifier*. The input and output specifiers identify the input to and output from BACKUP. In addition, you can specify qualifiers in the command line to modify BACKUP's default behavior.

8.1 An Overview of BACKUP Tasks

The most common BACKUP tasks are save, restore, and list operations. The input and output specifiers in the BACKUP command line, along with any qualifiers specified, determine which task BACKUP performs. Table 8–1 lists the BACKUP tasks described in this chapter.

Table 8-1: BACKUP Tasks

Task	Explanation
Save	BACKUP save operations safeguard data against accidental deletion or disk corruption. Save operations create save sets and place the contents of selected disk files, directories, volumes, or volume sets into the save set. A save set is a file created by BACKUP and written in a format that only BACKUP can interpret.
Restore	BACKUP restore operations return data saved during a BACKUP save operation to its original VMS file format.
List	BACKUP list operations list the date and time a save set was created, the user name of the person who created it, and the names of the files in the save set. Because BACKUP save sets are written in a unique format that only BACKUP can interpret, the list operation is the only way to determine the contents of a save set without restoring it.

Section 8.4 contains detailed explanations and examples of these BACKUP tasks.

8.2 The BACKUP Command Line

You use the DCL command BACKUP to perform BACKUP operations. A BACKUP command usually includes two parameters, as follows:

```
BACKUP input-specifier output-specifier
```

BACKUP evaluates the input and output specifiers to determine which type of operation it is to perform. These parameters specify the input to and output from BACKUP.

You can use several types of qualifiers to modify the default behavior of BACKUP. You can modify the action of the entire command, or you can change the way BACKUP processes the input and output specifiers. *Command qualifiers* modify the command itself, *input-specifier qualifiers* modify the processing of the input specifier, and *output-specifier qualifiers* modify the processing of the output specifier.

There are two types of input-specifier qualifiers: *input file-selection qualifiers* and *input save-set qualifiers*. Input file-selection qualifiers allow you to select specific files from the input specifier in a save or list operation. Input save-set qualifiers change the way BACKUP processes an input save set during a restore operation. There are also two types of output-specifier qualifiers: *output file qualifiers* and *output save-set qualifiers*. Output file qualifiers affect the way BACKUP restores files to a Files–11 structured disk volume. Output save-set qualifiers affect the processing of an output save set during a save operation.

The position of qualifiers in the BACKUP command line is important. Although command qualifiers can be placed anywhere in the command line, input- and output-specifier qualifiers are position-dependent. That is, input-specifier qualifiers must be placed immediately after the input specifier, and output-specifier qualifiers must be placed immediately after the output specifier. Additionally, several BACKUP

qualifiers can be used in more than one way. Therefore, be sure to understand the use of position-dependent qualifiers, in order to achieve the desired results from a BACKUP command.

For example, the /REWIND qualifier can be used as an output save-set qualifier in a BACKUP save operation, and as an input save-set qualifier in a BACKUP restore operation.

8.3 Using BACKUP Media

Magnetic tape is the most commonly used medium for storing BACKUP save sets. It is less expensive than disk media, and its compact size makes it easy to store. You can use more than one tape device at a time to save or restore data; this allows processing to continue on another tape while the tape used most recently is rewinding.

NOTE: This chapter describes the use of magnetic tape devices to save and restore BACKUP save sets. If your hardware is configured without a magnetic tape device, you can use a disk device to perform the same BACKUP operations by including the /SAVE_SET qualifier in the command, as shown in the examples throughout this chapter.

BACKUP treats all magnetic tape files as BACKUP save sets. You cannot use save-set specifications as both the input and output specifiers in a BACKUP command line. Therefore, you cannot specify a magnetic tape in both the input and output specifier.

Save-set specifications on magnetic tape are limited to 17 characters, including the period delimiter (.) and file type. Do not specify a directory or a version number in a magnetic tape save-set specification. The following is a valid save-set specification:

WKLY23SEPDLY.BCK

8.3.1 Tape Label Processing

By default, BACKUP processes information stored in the volume header record of the tape before writing to a magnetic tape. The volume header record contains volume protection information, an expiration date, and a volume label. By processing the volume protection information, BACKUP ensures that you have the right to access the volume in the manner you requested. By processing the tape expiration date, BACKUP prevents you from initializing a magnetic tape that has not yet expired. By comparing the volume label specified in the BACKUP command line to the volume label of the tape, BACKUP prevents you from creating a save set on the wrong magnetic tape.

Refer to the description of the BACKUP qualifier /LABEL in the reference section of this manual for help in specifying magnetic tape labels in BACKUP commands.

8.3.2 Initializing Magnetic Tapes

You must initialize a new magnetic tape to prepare it to receive data and to write a volume label, tape expiration date, and volume protection data to the volume header record. You can initialize a used magnetic tape to remove access to data stored on the tape, change the volume label, change the tape expiration date, change the volume protection data on the tape, and prepare the tape to receive new data. If a magnetic tape contains a non-ANSI or non-ISO label, initialize the tape to write an ANSI label to the volume header record.

You can use either the DCL command INITIALIZE or the BACKUP output save-set qualifier /REWIND to initialize a magnetic tape. To initialize a volume that was previously initialized with the output save-set qualifiers /REWIND and /PROTECTION, you must either own the volume (your UIC matches the owner UIC of the volume), or you must have VOLPRO privilege.

For more information on initializing magnetic tapes, refer to the description of the DCL command INITIALIZE in the *VMS General User's Manual* or the description of the BACKUP output save-set qualifier /REWIND in the reference section of this manual.

8.3.3 Protecting a Magnetic Tape Volume

By default, BACKUP applies no protection to magnetic tapes. You can assign a UIC protection code to a magnetic tape by specifying the /PROTECTION qualifier with the /REWIND qualifier. All save sets that you create on the tape will be protected with this protection code. If you specify no code with the /PROTECTION qualifier, BACKUP assigns the default protection of the current process to the tape. The following command specifies a protection code that allows full access to SYSTEM and OWNER and no access to GROUP and WORLD for the save set FRIDAY.DAT and all save sets created subsequently on the magnetic tape labeled DLY101:

```
$ BACKUP
_FROM: []
_TO: MTA0:FRIDAY.BCK/REWIND/LABEL=DLY101-
_$ /OWNER_UIC=[301,211]/PROTECTION=(S:RWED,O:RWED,G:,W:)
```

As shown in the preceding example, you can also use the output save-set qualifier /OWNER_UIC to assign a UIC value to the output tape. If you do not specify the /OWNER_UIC qualifier, the output tape receives the UIC value of the process that created the save set.

8.3.4 Using Tape Expiration Dates

You can specify an expiration date for a magnetic tape volume by using the output save-set qualifiers /REWIND and /TAPE_EXPIRATION. For example, your daily incremental BACKUP tapes should expire in seven days, and your weekly incremental BACKUP tapes should expire in one month. If you do not specify /TAPE_EXPIRATION, today's date is used. In the following example, the output tape is given an expiration date one week from the current date:

```
$ BACKUP/SINCE=BACKUP/TAPE_EXPIRATION=TODAY+7-
_FROM: DB2:[*...]
_TO: MTA0:102388.BCK/REWIND/LABEL=DLY101
```

8.3.5 Assigning Volume Labels to Magnetic Tapes

Magnetic tape volume labels can contain a maximum of six characters. You can use any ANSI "a" character in a magnetic tape volume label. The ANSI "a" characters include numbers, uppercase letters, and any one of the following nonalphanumeric characters:

```
! " % ' ( ) * + , _ . / : ; < = > ?
```

If you use any of the preceding nonalphanumeric characters, you must enclose the volume label with quotation marks.

Label your magnetic tapes according to the data contained on the tapes. The following table presents some suggestions for labeling tapes.

Label	Type of Backup	Expiration Date
DLY101	Daily, group 1, volume number 1	Expires in 7 days
DLY102	Daily, group 1, volume number 2	Expires in 7 days
WKY101	Weekly, group 1, volume number 1	Expires in 4 weeks
WKY201	Weekly, group 2, volume number 1	Expires in 4 weeks
MTH101	Monthly, group 1, volume number 1	Expires in 12 months
YRY101	Yearly, group 1, volume number 1	Expires in 5 years

8.4 Performing BACKUP Tasks

This section describes how to perform the BACKUP tasks commonly used in daily system operations, which include saving files to magnetic tape, restoring files from a magnetic tape save set, and listing the contents of save sets.

8.4.1 Saving Files

Because many users' files are stored on public volumes, it is important to regularly save files stored on public volumes. Public volumes are volumes mounted with the /SYSTEM, /GROUP, or /CLUSTER qualifiers. All system users granted access to a public volume share the volume.

Enter a command in the following format to save files to a magnetic tape save set:

```
BACKUP input-specifier save-set-specification
```

To write a save set to a disk, you must include the output save-set qualifier /SAVE_SET in the BACKUP command line, as shown:

```
BACKUP input-specifier save-set-specification/SAVE_SET
```

Use a combination of incremental and full (image) save operations to safeguard data stored on public volumes. An incremental save operation saves only those files that have been created or modified since the last save operation. A full backup (usually an image save operation) saves all files on a volume. Periodic full backups are necessary to provide the basis for the reconstruction of a lost volume. If a volume is lost, you must first restore the most recent full backup and then restore incremental save sets performed since the last full backup. The most efficient way to restore incremental save sets is in reverse chronological order.

Perform incremental save operations more frequently than full backups. After consulting with users of the system, the system manager can decide how frequently to save files and volumes and how long to retain BACKUP media.

The following sample schedule for backing up public disk volumes provides adequate data protection for many installations:

- Daily—A daily incremental save set retained for 7 days. This schedule requires 7 daily sets of magnetic tapes that are rotated on a weekly basis.
- Weekly—A weekly incremental save set retained for 4 weeks. This schedule requires 4 weekly sets of magnetic tapes that are rotated every 4 weeks.
- Monthly—A monthly full backup (usually an image save set) retained for a year. This schedule requires 12 monthly sets of magnetic tapes that are rotated annually.

To follow this BACKUP schedule, use a combination of incremental save and image save operations.

8.4.1.1 Incremental Save Operations

An incremental save operation saves only those files that have been created or modified on a volume or volume set since the last save operation. Incremental save operations take less time to perform than operations that save all files, and they require fewer magnetic tape or disk volumes to store each save set.

To perform incremental operations to save files created or modified since the last save operation, use the command qualifier /RECORD and the input file-selection qualifier /SINCE=BACKUP. The command qualifier /RECORD directs BACKUP to record the current date and time in the BACKUP date field of each file's header record. (To use the command qualifier /RECORD, you must own the files or have the user privilege SYSPRV.) The /SINCE=BACKUP qualifier directs BACKUP to select only those files that have been created or modified since the last BACKUP/RECORD operation.

NOTE: If you use the command qualifier /RECORD to perform incremental save operations on a disk volume, do not allow other users to use /RECORD in their BACKUP operations on the same disk volume. If other users specify the /RECORD qualifier, the dates in the BACKUP date fields of file header records will change. This will make it impossible for you to save all files created or modified since you last performed a save operation.

Daily Incremental Save Operations

To perform daily incremental save operations, follow the instructions in the previous section for incremental save operations. You may also want to include the command qualifier /IGNORE=INTERLOCK. The /IGNORE=INTERLOCK qualifier instructs BACKUP to save files even if they are open for writing; this means that you may produce a copy of a file in a partial state. If you do not use /IGNORE=INTERLOCK, any files that are open at the time of the save operation are not saved. Note that you must have the user privilege SYSPRV, a system UIC, or you must own the volume to use the IGNORE=INTERLOCK qualifier.

The /IGNORE=INTERLOCK qualifier is especially useful if you have files that are always open (and would otherwise never be backed up).

The following example saves all files created or modified since the last save operation to a save set named INCD12JUN.BCK on the tape labeled DLY03:

```
$ BACKUP/IGNORE=INTERLOCK/RECORD/SINCE=BACKUP
_FROM: PUBLIC:[*...]
_TO: MTA0:INCD12JUN.BCK/LABEL=DLY03
```

Weekly Incremental Save Operations

Perform weekly incremental save operations the same way you perform daily incremental save operations. However, the input file-selection qualifier /SINCE specifies the date of the last weekly incremental save operation, normally one week earlier.

The following example assumes that the current date is October 14, 1988:

```
$ BACKUP/IGNORE=INTERLOCK/RECORD/SINCE=7-OCT-1988
_FROM: PUBLIC:[*...]
_TO: MTA0:INCW14JUN/LABEL=WKLY03
```

8.4.1.2 Image Save Operations

An image save operation, also called a full backup, creates a save set containing an entire volume or volume set. This section describes how to perform image save operations.

Use the command qualifier /IMAGE to perform an image save operation. To use /IMAGE in a save operation, you either need write access to the volume index file (INDEXF.SYS) and the bit map file (BITMAP.SYS), or the input medium must be write-locked. BACKUP opens the index file to synchronize with the file system (no update is made). Finally, you must have read access to all files on the input medium.

An image save operation saves all files on the input disk, including files marked for deletion and lost files (files without a directory entry). Therefore, you cannot use input file-selection qualifiers in an image save operation. Also, the input specifier of an image save operation must be a device name. You cannot use directory or file specifications in the input specifier of an image save operation.

During an image save operation, BACKUP creates a *save-volume summary record* on the volume to which it writes the save set. This record contains data necessary for initializing the disk volume to which the image save set is restored.

Specify the command qualifier /RECORD if you are performing the image save operation in conjunction with incremental save operations that use /RECORD. The /RECORD qualifier writes the date and time of the save operation into each file's header record. When you perform the next incremental save operation, only files created or modified since the image backup are saved.

The following example shows an image save operation from a Files–11 disk to magnetic tape:

```
$ BACKUP/IMAGE/RECORD DRA1:  MTA0:1JUN.BCK/REWIND/LABEL=DLY101
```

You may need to mount additional magnetic tapes, depending upon the size of the disk being saved and the size of the magnetic tape. By default, BACKUP applies the /NOREWIND qualifier to a magnetic tape save operation and does not initialize the first tape in the save set. Subsequent tapes in a multivolume save set are initialized, however.

8.4.2 Restoring Files

A BACKUP restore operation returns data from a BACKUP save set to its original VMS file format on the specified output disk. Restore files, directories, volume, or volume sets if they have been inadvertently deleted or if the disks containing the original data have been lost or corrupted.

Enter a command in the following format to restore all files in a save set:

BACKUP save-set-specification output-specifier

The output specifier can be a device, directory, file name, or wildcard character.

To restore a save set from a disk, you must include the input save-set qualifier /SAVE_SET in the BACKUP command line, as shown:

BACKUP save-set-specification/SAVE_SET output-specifier

When restoring a save set from a streaming tape device (like a TK50), you improve the performance by increasing the number of I/O buffers used in the restore operation to 5. Enter the command qualifier /BUFFER_COUNT=5 to increase the number of I/O buffers. When restoring a save set from a nonstreaming tape device, leave the buffer count at its default value of 3.

This section describes how to perform the following types of restore operations from save sets stored on magnetic tapes:

- Selective restore—You perform this backup operation to recover specific files from a magnetic tape save set.
- Full restore—You perform this backup operation to re-create a disk that may have been lost or corrupted.

8.4.2.1 Restoring Selected Files from a Save Set

You can restore specific files from a save set by using the input save-set qualifier /SELECT. In the following example, the file STRAT1.DAT in the directory [LYKINS.GLENDO] was accidentally deleted. The most recent backup of the deleted file is saved in the magnetic-tape save set NOV2SAVE.BCK. This example restores the file STRAT1.DAT to its original directory by using the input save-set qualifier /SELECT:

```
$ BACKUP
_From: MTA0:NOV2SAVE.BCK/SELECT=[LYKINS.GLENDO]STRAT1.DAT;5
_To:   STRAT1.DAT;5
$ DIRECTORY STRAT1.DAT
Directory [LYKINS.GLENDO]

STRAT1.DAT;5

Total of 1 file.
$
```

8.4.2.2 Restoring Files to the Directory from Which They Were Saved

BACKUP does not automatically restore files to the directory from which they were saved. To restore files to the directory from which they were saved, use the directory wildcard character [*...] in the output specifier. The following example restores a file named [PUBLIC]FINANCE.DAT from the save set NOVELS.BCK on tape MTA0 to the directory from which the file was saved on DUA1:

```
$ BACKUP MTA0:NOVELS.BCK/SELECT=[PUBLIC]FINANCE.DAT DUA1:[*...]
```

8.4.2.3 Restoring Files from Multivolume Save Sets

You might need to restore a small number of files from a multivolume save set. If your save set encompasses more than one tape, it is usually possible to begin the restore operation by mounting the volume that contains the files, rather than the first volume of the save set. (Because an image restore operation restores the contents of the entire volume or volume set, processing must begin with the first volume if you use the command qualifier /IMAGE in your restore operation.)

If the mounted tape is not the first volume in the save set, you receive the following warning message, which has no effect on the restore operation:

```
%BACKUP-W-NOT1STVOL, 'name' is not the start of a save set
```

8.4.2.4 Restoring Entire Disk Volumes

You can restore an entire disk volume if the disk volume was destroyed, lost, or corrupted. To restore all files from a save set, enter a command in the following format:

BACKUP save-set-specification device-specification

If you have been performing a combination of full and incremental save operations on a volume, use the following procedure to recover the volume if it has been lost, corrupted, or destroyed:

1. First, restore the volume from the last image save set using an image restore operation. The following example restores a multivolume magnetic tape save set named FULLJUN84.BCK to DRA0. The command qualifiers /RECORD and /IMAGE are required for the correct operation of this procedure.

```
$ MOUNT/FOREIGN DRA0:
%MOUNT-I-MOUNTED, mounted on _DRA0:
$ BACKUP/IMAGE/RECORD MTA0:FULLJUN84.BCK,MTA1 DRA0:
%BACKUP-I-RESUME, resuming operation on volume 2
%BACKUP-I-RESUME, resuming operation on volume 3
%BACKUP-I-RESUME, resuming operation on volume 4
    .
    .
    .
$ DISMOUNT/NOUNLOAD DRA0:
```

2. Next, mount the disk as a file-structured volume and restore all incremental save sets created since the image save set was created. The most efficient way to restore incremental save sets is in reverse chronological order. Start with the last daily incremental save set; then restore the preceding daily incremental save sets and finally the weekly incremental save sets, as follows:

```
$ MOUNT DRA0: PUBLIC
%MOUNT-I-MOUNTED, PUBLIC mounted on _DRA0:
! Mount the tape containing the save set INCD17JUN in drive MTA0
$ BACKUP/INCREMENTAL  MTA0:INCD17JUN DRA0:
! Remove the tape containing the save set INCD17JUN from drive MTA0
! Mount the tape containing the save set INCD16JUN in drive MTA0
$ BACKUP/INCREMENTAL  MTA0:INCD16JUN DRA0:
! Remove the tape containing the save set INCD16JUN from drive MTA0
! Mount the tape containing the save set INCD15JUN in drive MTA0
$ BACKUP/INCREMENTAL  MTA0:INCD15JUN DRA0:
! Remove the tape containing the save set INCD15JUN from drive MTA0
! Mount the tape containing the save set INCW14JUN in drive MTA0
$ BACKUP/INCREMENTAL  MTA0:INCW14JUN DRA0:
! Remove the tape containing the save set INCW14JUN from drive MTA0
! Mount the tape containing the save set INCW7JUN in drive MTA0
$ BACKUP/INCREMENTAL  MTA0:INCW7JUN DRA0:
```

   If you choose to exclude selectively certain files in your incremental save operations (for example, listing files or batch logs), these files will not be restored but will have directory entries in the resulting volume. You can delete these null directory entries by running a repair pass with the Analyze/Disk_Structure Utility.

If directory files were renamed during the time period covered by the incremental save sets, these directories appear on the reconstructed volume under both their old and new directory names. The files that were written since the directory was renamed appear under the new directory name; the other files, written before the directory was renamed, appear under the old directory name. You must merge the old and new directories manually.

If you renamed many directory files, your disk may become full during the course of the incremental restore operations. Merge the old and new directories and complete the incremental restore operations.

8.4.3 Listing the Contents of a BACKUP Save Set

Because BACKUP save sets are written in a format that only BACKUP can interpret, the list operation is the only way to determine the contents of a save set without restoring the save set. You can either display the list on a terminal or write it to a specified output file. You can perform a list operation in conjunction with any other BACKUP operation. Use the command qualifier /LIST to perform a list operation.

By default, a list operation supplies the same information about files in the save set as the DCL command DIRECTORY/DATE/SIZE, including the actual number of blocks saved for each file. Specify the command qualifier /FULL with the /LIST qualifier to list information about files supplied by the DCL command DIRECTORY/FULL, including the number of blocks allocated for each file.

The following command displays save-set information on your terminal about a magnetic tape save set. Before entering this command, the magnetic tape containing the save set was loaded on drive MTA0.

```
$ BACKUP/LIST MTA0:2MAR1555.BCK/SAVE_SET
Listing of save set(s)

Save set:          2MAR1555.BCK
Written by:        POLYANNA
UIC:               [000200,000207]
Date:              21-AUG-1988 09:36:14.68
Command:           BACKUP/LOG [USER.SAVE] MTA0:2MAR555.BCK/REWIND/LABEL=WK102
Operating system:  VMS version 5.0
BACKUP version:    5.0
CPU ID register:   08000000
Node name:         _SUZI::
Written on:        _MTA0:
Block size:        8192
Group size:        10
Buffer count:      3

[USER.SAVE]ANOTHER.DAT;1                          1  18-AUG-1988 14:10
[USER.SAVE]LAST.DAT;1                             1  18-AUG-1988 14:11
[USER.SAVE]THAT.DAT;1                             7  18-AUG-1988 14:10
[USER.SAVE]THIS.DAT;2                             1  18-AUG-1988 13:44

Total of 4 files, 10 blocks
End of save set
```

The following command combines a list operation with a save operation to magnetic tape:

```
$ BACKUP/LIST=MYBACK.DAT [PRAMS] MTA0:2MAR1555.BCK/LABEL=DLY201
```

BACKUP verifies that the tape label is DLY201 and copies the contents of the directory [PRAMS] to a save set named 2MAR1555.BCK. The command qualifier /LIST causes BACKUP to write save-set information to the file MYBACK.DAT as the save operation proceeds.

If you do not know the names of save sets stored on a magnetic tape volume, you only need to specify the name of the drive in which the tape is inserted as the output specifier in the BACKUP/LIST command. If you only use the device specification, BACKUP reads the next save set it encounters on the magnetic tape and then stops processing when it reaches the end of that save set. BACKUP does not automatically rewind to the beginning-of-tape marker (BOT) unless you include the input save-set qualifier /REWIND in your command. Therefore, you can proceed to the next save set (if one exists) by repeating the command.

By including the asterisk wildcard character (*) with the device specification and the /REWIND qualifier, you can direct BACKUP to rewind to the beginning-of-tape and list all save sets on the tape volume, as follows:

```
$ BACKUP/LIST MTA0:*/REWIND
```

To list only the names of the save sets on a magnetic tape volume, mount the magnetic tape volume as a Files–11 volume. Then enter the DCL command DIRECTORY, as follows:

```
$ MOUNT MTA0: SAVE01 TAPE
%MOUNT-I-MOUNTED, SAVE01 mounted on _MTA0:
$ DIRECTORY/SIZE/DATE/PROTECTION TAPE:
Directory MTA0:[]

CONTRACTS.BCK;1           5  13-JUN-1988 12:11  (RWED,RWED,RE,)
JUL20SAVE.BCK;1           3  20-JUL-1988 23:59  (RWED,RWED,RE,)
MYPHILE.BCK;1             2  28-SEP-1988 12:00  (RWED,RWED,RE,)

Total of 3 files, 10 blocks.
```

8.5 Protecting a BACKUP Save Set

Limiting access to BACKUP save sets is an important part of system security. The file system treats a BACKUP save set as a single file. Therefore, anyone who has access to a save set can read any file in the save set. BACKUP does not check protection on individual files until after they are restored to standard VMS file format.

To maintain system security, it is crucial that you protect save sets adequately. Accordingly, you should assign restrictive protection to save sets by using the output save-set qualifiers /OWNER_UIC and /PROTECTION. Sufficient protection can prevent nonprivileged users from mounting a save-set volume or reading files from a save set. You should also take physical security precautions with save sets stored offline by keeping BACKUP media in locked cabinets.

Protection information is written to the volume header record of a magnetic tape and applies to all saves sets stored on the tape. Therefore, the output save-set qualifiers /OWNER_UIC and /PROTECTION are effective on magnetic tape save sets only if you specify the output save-set qualifier /REWIND. This rewinds the tape to its beginning, writes the protection data to the volume header record, and initializes the tape. If you specify /PROTECTION, any protection categories that you do not specify default to your default process protection. If you do not specify /REWIND with the /PROTECTION and /OWNER_UIC qualifiers, the magnetic tape receives no protection. (By default, BACKUP applies no protection to magnetic tapes.)

The following example saves the directory [PAYROLL] to KNOX.BCK on the magnetic tape drive MFA2. The output save-set qualifier /LABEL provides the label BANK01 for the tape. The output save-set qualifier /OWNER_UIC assigns an owner UIC of [003,003] to the save set. The output save-set qualifier /TAPE_EXPIRATION assigns an expiration date of January 15, 1989 to the tape. The output save-set qualifier /PROTECTION assigns the owner of the volume read, write, execute, and delete access. SYSTEM users are assigned read, write, and execute access; GROUP users are assigned read and execute access; WORLD users are assigned no access.

```
$ BACKUP
_FROM: [PAYROLL]
_TO: MFA2:KNOX.BCK/LABEL=BANK01/REWIND/OWNER_UIC=[003,003]-
_$ /TAPE_EXPIRATION=15-JAN-1989/PROTECTION=(S:RWE,O:RWED,G:RE,W)
```

When a nonprivileged user wants to restore a particular file, do not lend the volume containing the save set. You could give away access to all the files on the volume. The safest way to restore a particular file is to restore the file selectively, as shown in the following example:

```
$ BACKUP MTA0:JULY.BCK/SELECT=[JONES.TEXTPROC]LASTMONTH.DAT -
_$ [*...]/BY_OWNER=ORIGINAL
```

The selected file is restored with its original directory, ownership, and protection. In this way, the file system determines if the user is permitted access to the file.

8.6 Using Command Procedures to Perform Backup Tasks

If you perform similar BACKUP tasks regularly, you can write command procedures to assist you in performing these tasks. For example, if you perform daily incremental save operations to magnetic tape, you use identical or similar command strings. You could use a command procedure to enter the necessary commands, as in the following example:

```
$ ! Command procedure DAILYBACK.COM
$ !
$ ! Execute this command procedure interactively,
$ !  by entering the command @[directory]DAILYBACK
$ !  at the DCL prompt.
$ !
$ ! The BACKUP command in this procedure contains the
$ !  output save-set qualifier /REWIND.  Therefore, this
$ !  command procedure always initializes the output tape.
$ !
$ ON CONTROL_Y THEN GOTO EXIT
$ ON WARNING THEN GOTO EXIT
$ INQUIRE DRIVE "Enter the drive name (without a colon)"
$ ALLOCATE 'DRIVE'
$ INQUIRE SAVESET_SPEC "Enter the save-set specification"
$ INQUIRE LBL "Enter the tape label"
$ INQUIRE EXP "Enter the tape expiration date"
$ BACKUP/NOASSIST/RECORD/IGNORE=INTERLOCK/SINCE=BACKUP -
  [...] 'DRIVE':'SAVESET_SPEC'/REWIND/LABEL='LBL'/TAPE_EXPIRATION='EXP'
$ EXIT:
$ DEALLOCATE 'DRIVE'
$ EXIT
```

The preceding command procedure allocates the specified tape drive. BACKUP searches the tape's volume header record for a volume label and compares the label you specified with the volume label. If the volume header record contains no volume label, BACKUP writes the label and expiration date you specified to the volume header record and initializes the tape. Otherwise, BACKUP compares the tape's volume label with the label you specified and ensures that the tape is expired. If the tape is not expired or the label does not match, the command procedure exits. If the tape is expired and the label matches, BACKUP writes the expiration date you specified to the volume header record and initializes the tape. After initializing the tape, BACKUP saves all files in the current default directory tree that have been

created or modified since the last save operation to a save set with the name you specified.

Note that if you do not have an ON WARNING statement in your command procedure and BACKUP returns a FATAL, ERROR, or WARNING status code, the command procedure exits immediately after the BACKUP command is executed. If you want your command procedure to complete, include an ON WARNING statement as shown in the preceding procedure.

DIGITAL provides two template command procedures in the SYS$EXAMPLES directory to assist system managers in designing BACKUP command procedures. These command procedures are called BACKUSER.COM and RESTUSER.COM.

Chapter 9

Maintaining Acceptable System Performance

Performance management of a VMS system means optimizing your hardware and software resources for the current workload. This task entails several related activities:

- Acquiring a thorough familiarity with your workload and an understanding of how that workload uses the system's resources. This knowledge, combined with an appreciation of the VMS resource management mechanisms, will enable you to establish realistic standards for system performance in areas such as the following:
  - Interactive and batch throughput
  - Interactive response time
  - Batch job turnaround time
- Routinely monitoring system operating conditions to determine if, when, and why a given resource is approaching capacity.
- Investigating reports of degraded performance from users.
- Planning for changes in the system workload or hardware configuration and being prepared to make any necessary adjustments to system values.
- Performing, after installation, certain optional system management operations.

This chapter introduces the basic concepts of performance management. It is not meant to be used as a tutorial for tuning your system.

9.1 Knowing Your Workload

One of the most important assets that a system manager brings to any performance evaluation is an understanding of the normal workload and operating conditions of the system. Each system manager must assume the responsibility for understanding the system's workload sufficiently to be able to recognize normal and abnormal operating conditions; to predict the effects of changes in applications, operations, or usage; and to recognize typical throughput rates. The system manager should be able to answer such questions as the following:

- What is the typical number of users on the system at each time of day?
- What is the typical response time for various tasks for this number of users, at each hour of operation?
- What are the peak hours of operation?
- Which jobs typically run at which time of day?
- Which commonly run jobs are intensive consumers of the CPU, memory, and disk space?
- Which applications involve the most image activations?
- Which parts of the system software, if any, have been modified or user-written, such as device drivers?
- Are there any known system bottlenecks? Are there any anticipated ones?

If you are new to VMS system management, you should observe system operation using the following tools:

- Monitor Utility
- Accounting Utility
- SHOW commands (available through DCL)

Over time you will learn about metrics such as the typical page fault rate for your system, the typical CPU usage, the normal memory usage, and typical modes of operation, and you will begin to see how certain activities affect system performance. As you continue to monitor your system, you will come to know what range of values is acceptable, and you will be better prepared to detect unusual conditions.

Routine evaluation of the system is critical for effective performance management. The best way to avoid problems is to anticipate them; you should not wait for problems to develop before you learn how the system performs. You can learn more about your system's operation if you use the Monitor and Accounting utilities on a regular basis to capture and analyze certain key data items. By observing and collecting this data, you will also be able to see usage trends and predict when your system may reach its capacity.

When you use the tools that measure and report system operations, keep in mind that the tools themselves use some system resources. Be careful, therefore, in selecting the items you want to measure and the frequency with which you collect the data. If you use the tools excessively, the consumption of system resources to collect, store, and analyze the data can distort your picture of the system's workload and capacity. The best approach is to have a plan for collecting and analyzing the data.

9.1.1 Using the Monitor Utility (MONITOR)

You can develop a database of performance information for your system by running MONITOR continuously as a background process. The directory with the logical name SYS$EXAMPLES includes three command procedures that you can use to establish the database. Instructions for installing and running the procedures are contained in the comments at the beginning of each one. Following is a brief summary of these procedures:

- SUBMON.COM—Starts MONITOR.COM as a detached process. You should invoke SUBMON.COM from the DCL procedure SYS$MANAGER:SYSTARTUP_V5.COM.
- MONITOR.COM—Creates a summary file from the recording file of the previous boot, then begins recording for this boot. The recording interval is 10 minutes.
- MONSUM.COM—Generates two clusterwide multifile summary reports; one for the previous 24 hours, and one for the previous day's prime-time period (9 A.M. to 6 P.M.). These are mailed to the system manager, and then the procedure resubmits itself to run each day at midnight.

While MONITOR data is recorded continuously, a summary report can cover any contiguous time segment. The command file MONSUM.COM, which is executed every midnight, generates and mails the two multifile summary reports described previously. These reports are not saved as files, so if you want to keep them, you must either extract them from your mail file or alter the MONSUM.COM command procedure to save them.

9.1.2 Using the Accounting Utility (ACCOUNTING)

The Accounting Utility can be used to generate reports that indicate how well the system is performing. Of particular interest to performance management is image-level accounting, which records information on the system resources consumed by the execution of specific images. By being aware of the images that use the most resources at your site, you can better direct your efforts toward controlling them and the resources they consume.

Images used frequently are typically good candidates for code sharing, whereas images that consume large quantities of various resources may be forced to run in a batch queue. In batch queues, the number of simultaneous processes can be controlled. Using a series of commands like those in the following example, you

can produce a report containing the resource usage information necessary to manage images.

NOTE: It is assumed in the following example that image-level accounting records have been collected previously. (You enable image-level record collection by entering the DCL command SET ACCOUNTING/ENABLE=IMAGE.)

```
$ ACCOUNTING  /TYPE=IMAGE  /OUTPUT=BYNAM.LIS -
_$ /SUMMARY=IMAGE -
_$ /REPORT=(PROCESSOR,ELAPSED,DIRECT_IO,FAULTS,RECORDS)
$ SORT  BYNAM.LIS  BYNAM.ORD  /KEY=(POS=16,SIZ=13,DESCEND)
   .
   .
   .
(Edit BYNAM.ORD to relocate heading lines)
   .
   .
   .
$ TYPE BYNAM.ORD
```

You should be careful when using image-level accounting on your system. As a rule, you should enable image-level accounting only when you plan to invoke ACCOUNTING to process the information provided in the file SYS$MANAGER:ACCOUNTNG.DAT. Once you have collected enough data for your purposes, disable image-level accounting by entering the DCL command SET ACCOUNTING /DISABLE=Image. While image activation data can be very helpful in performance analysis, it can be a waste of processing time and disk storage if the data is collected but never used.

9.1.3 Managing Workload

System performance is directly proportional to the efficiency of workload management. Each installation must develop its own strategy for this key task. Before adjusting any system values, answer the following questions:

- Is there a time of day when the workload "peaks," that is, when it is noticeably heavier than at other times?
- Is there any way to balance the workload better? Perhaps some voluntary measures can be adopted by users, after appropriate discussion.
- Could any jobs be run better as batch jobs, preferably during nonpeak hours?
- Have primary and secondary hours of operation been employed with users? If not, could system performance benefit by adopting this practice? If the primary and secondary hours are in use, are the choices of hours the most appropriate for all users? (Plan to review this issue every time you either add or remove users or applications, to ensure that the desired balance is maintained.)
- Can future applications be designed to work around any known or expected system bottlenecks? Can present applications be redesigned somewhat, for the same purpose?

- Are you using to the utmost the code-sharing ability that the VMS system offers you? If not, you will find that code sharing provides an excellent means to conserve memory, thereby improving performance over the life of the system.

Do not adjust any system values until you are satisfied that all these issues are resolved and that your workload management strategy is correct.

9.1.4 Distributing Workload

You should distribute the workload as evenly as possible over the time your system is running. Although the work schedule for your site may make it difficult to schedule interactive users at optimum times, the following techniques may be helpful:

- Run large jobs as batch jobs—Establish a site policy that encourages the submission of large jobs on a batch basis. Regulate the number of batch streams so that batch usage is high when interactive usage is low. You might also want to use DCL command qualifiers to run batch jobs at lower priority, adjust the working set sizes, and/or control the number of concurrent jobs.

- Restrict system use—Do not permit more users to log in at one time than the system can support with an adequate response time. You can restrict the number of interactive users with the DCL command SET LOGINS/INTERACTIVE. You can also control the number of concurrent processes with the MAXPROCESSCNT system parameter, and the number of remote terminals allowed to access the system at one time with the RJOBLIM system parameter.

  You might also restrict use of the system by groups of users to certain days and hours of the day. You can use the Authorize Utility to define the permitted login hours for each user. In particular, refer to the AUTHORIZE qualifiers /PRIMEDAYS, /P_RESTRICT, /PFLAGS, /SFLAGS, and /S_RESTRICT. Remember you can use the DCL command SET DAY to override the conventional day of the week associations for primary and secondary days. For example, you might need to specify a primary day of the week as a secondary day when it is a holiday.

- Design applications to reduce demand on binding resources—If you know where your system bottlenccks are or where they will likely occur in the near future, you can distribute the workload more evenly by planning usage that minimizes demand on the bottleneck point(s).

9.1.5 Installing Known Images

If you have programs that are frequently used on your system, you should consider installing them as known images. In general, programs should be installed as known images if they

- Are frequently run
- Are usually run concurrently by several processes
- Require special privileges

Chapter 2 describes how to install programs as known images.

By specifying appropriate qualifiers to INSTALL commands, you can assign any of the following attributes to known images:

- **Permanently open**—Directory information on the image file remains permanently resident, eliminating the usual directory search required to locate a file. The cost of keeping an image file permanently open is approximately one page of nonpaged dynamic memory per file.
- **Header resident**—The header of the image file (native images only) remains permanently resident, saving one disk I/O operation per file access. For images with single-block file headers, the cost is less than one page of paged dynamic memory per file; for images with multiblock headers, the cost varies according to the header block count. The images must also be declared permanently open.
- **Privileged**—Amplified privileges are temporarily assigned to any process running the image (executable images only), permitting the process to exceed its user authorization file (UAF) privilege restrictions during execution of the image. In this way, users with normal privileges can run programs that require higher than normal privileges.
- **Protected**—A shareable image contains protected code, that is, code that runs in Kernel or Executive mode but that can be called by a user-level image. Protected images must be declared shared.
- **Shared**—More than one user can access the read-only and noncopy-on-reference read/write sections of the image concurrently, so that only one copy of those sections ever need be in physical memory. (Copy-on-reference sections always require a separate copy for each process.) The image is implicitly declared permanently open.
- **Writable**—When a shared noncopy-on-reference writable section is removed from physical memory (for paging reasons or because no processes are referencing it), it is written back to the image file. Any updates made by processes mapped to the section, therefore, are preserved (while the initial values are lost). The image must also be declared shared.

9.1.6 Tuning a System

Tuning is the process of altering various system values to obtain the optimum *overall* performance possible from any given configuration and workload. However, the process does not include the acquisition and installation of additional memory or devices, although in many cases such additions (when made at the appropriate time) can vastly improve system operation and performance.

Always aim for best overall performance, that is, performance viewed over time. The workload is constantly changing on most systems. System parameters that produce optimal performance at one time may not produce optimal performance a short time later as the workload changes. Your goal is to establish values that, on the average, produce the best overall performance.

Before you undertake any action, you must recognize that the following sources of performance problems cannot be cured by adjusting system values:

- Improper operation
- Unreasonable performance expectations
- Insufficient memory for the applications attempted
- Inadequate hardware configuration for the workload, such as too slow a processor, too few buses for the devices, too few disks, and so forth
- Improper device choices for the workload, such as using disks with insufficient speed or capacity
- Hardware malfunctions
- Human errors, such as poor application design or allowing one process to consume all available resources

When you make adjustments, you normally select a very small number of values for change, based on a careful analysis of the behavior being observed. These values are usually either system parameters or entries in the User Authorization File (UAF) that affect particular users.

Normally, system parameters are modified automatically by the system using AUTOGEN; AUTOGEN uses system configuration data to automatically set system parameters. You can also use the SYSGEN Utility to manually alter system parameters.

One of AUTOGEN's special features is that it makes automatic adjustments for you in associated parameters. To control the values in the UAF, you use the Authorize Utility.

9.1.7 Predicting When Tuning Is Required

Under most conditions, tuning is rarely required for VMS systems. The AUTOGEN command procedure, which is included in the operating system, establishes initial values for all the configuration-dependent system parameters so that they match your particular configuration. Additionally, the system includes features that in a limited way permit it to adjust itself dynamically during operation. That is, the system detects the need for adjustment in certain areas, such as the nonpaged dynamic pool, working set size, and the number of pages on the free and modified page lists. The system makes rough adjustments in these areas automatically. As a result, these areas can grow dynamically, as appropriate, during normal operation.

A frequent reason for disappointment in system performance is ultimately due to insufficient hardware capacity. Once the demand on a system exceeds its capacity, adjusting system values will not result in any significant improvements, simply because such adjustments are a means of trading off or juggling existing resources.

Although tuning is rarely required, you should recognize that system tuning may be needed under the following conditions:

1. If you have adjusted your system for optimal performance with current resources and then acquire new capacity, you must plan to compensate for the new configuration. In this situation, the first and most important action is to execute the AUTOGEN command procedure.
2. If you anticipate a dramatic change in your workload, you should expect to compensate for the new workload.

9.1.8 Evaluating Tuning Success

Whenever you adjust your system, you should monitor its behavior afterward, to be sure that you have obtained the desired results. To observe results, use the Monitor Utility and the various forms of the DCL SHOW command.

For example, you might consider running some programs whose results you believe are fixed and reproducible, at the same time that you run your normal workload. If you run the programs and measure their running times under nearly identical workload conditions both before and after your adjustments, you can obtain a basis for comparison.

However, when applying this technique, remember to take the measurements under very similar workload conditions. Also, remember that this test alone does not provide conclusive proof of success. There is always the possibility that your adjustments may have favored the performance of the image you are measuring—to the detriment of other images. Therefore, in all cases, continue to observe system behavior closely for a time after you make any changes.

9.1.9 Performance Options

Following is a list of optional system management operations, normally performed after installation, that often result in improved overall performance. Note, however, that not all options are appropriate at every site.

- Decompress system libraries—Most of the libraries shipped with Version 4 and later versions of the VMS operating system are in a compressed format in order to conserve disk space. The system dynamically decompresses them whenever they are accessed, and the resulting performance slowdown is especially noticeable during link operations and when requesting online help. If you have sufficient disk space, decompressing the libraries improves both CPU and elapsed time performance. To do this, invoke the command procedure SYS$UPDATE:LIBDECOMP.COM. The decompressed object libraries take up about 25 percent more disk space than when compressed; the decompressed help libraries take up about 50 percent more disk space.

- Disable file system high-water marking—This security feature guarantees that users cannot read data they have not written. It is implemented by erasing the previous contents of the disk blocks allocated every time a file is created or extended. High-water marking is set by default whenever a volume is initialized.

  Disabling this feature improves system performance by a variable amount, depending on the frequency of new file creation, the frequency of extending existing files, and the fragmentation of the volume. To disable high-water marking, you can specify the /NOHIGHWATER qualifier when initializing the volume, or you can enter the following DCL command at any time:

  ```
  $ SET VOLUME/NOHIGHWATER_MARKING device-spec[:]
  ```

  Then dismount and remount the volume. However, you should consider the security implications of disabling this feature.

- Set RMS file extend parameters—Because files extend in increments of twice the multiblock count (default 16), system defaults provide file extension of only 32 blocks. Thus, when files are created or extended, increased I/O may slow performance. The problem can be corrected by specifying larger values for SYSGEN file extend parameters or by setting the system parameter RMS_EXTEND_SIZE=80.

- Relink images—Beginning with VMS Version 4.0, the run-time library (VMSRTL) was separated into five smaller libraries. Running images linked under previous versions of the VMS operating system will therefore incur the image activation costs of mapping all five libraries, even if only one is needed. You may improve performance by relinking pre-Version 4.0 images that reference run-time library routines, so that only the required libraries are mapped and activated.

- Install frequently used images—When an image is accessed concurrently by more than one process on a routine basis, install the image with the Install Utility, specifying the /OPEN, /SHARED, and /HEADER_RESIDENT qualifiers. You will thereby ensure that all processes use the same physical copy of the image, and that the image will be activated in the most efficient way.

  Generally, an image takes about two additional physical pages when installed /OPEN/HEADER_RESIDENT/SHARED. The utility's LIST/FULL command shows the highest number of concurrent accesses to an image installed with the /SHARED qualifier. This information can help you decide whether installing an image is worth the space. For more information on the Install Utility, refer to the Install Utility section in Part II of this manual.

- Reduce system disk I/O—You can move frequently accessed files off the system disk and use logical names or, where necessary, other pointers to access them. For example:

  - SYSUAF.DAT (SYSUAF is the logical name)
  - RIGHTSLIST.DAT (RIGHTSLIST is the logical name)
  - VMSMAIL.DAT (VMSMAIL is the logical name)
  - NETPROXY.DAT (NETPROXY is the logical name)
  - JBCSYSQUE.DAT (File specification parameter for the START/QUEUE /MANAGER command)
  - ERRFMT log files (SYS$ERRORLOG is the logical name)
  - MONITOR log files (SYS$MONITOR is the logical name)
  - Default DECnet account (DECNET record in SYSUAF file)

You can also consider moving paging and swapping activity off the system disk by creating large secondary page and swap files on a less heavily used disk.

However, be sure to understand the nature of system values before adjusting them. Without the proper level of understanding, you may very well degrade, rather than improve, overall performance.

While investigating the cause of an apparent performance problem, it is wise to keep in mind that tuning is a last resort solution. This perspective is extremely important. Too many users assume incorrectly that tuning is a first rather than a last resort solution.

Chapter 10

Operator Tasks

Certain system management operations require your attention on a regular basis in order to maintain the system properly. Following are some of the activities you may perform in your role as system operator:

- Performing regular backups of user data
- Saving crash dumps following a system failure
- Printing and resetting the operator log file
- Printing and resetting the error log file
- Collecting information in the accounting log file

This chapter describes how to perform each of these tasks.

10.1 Performing Backups

One of the most commonly performed system operations is backing up files on public volumes. Backing up a volume means copying the contents of the volume to another volume or set of volumes. It is a precautionary measure to allow you to recover from the loss or destruction of valuable information. Most sites establish a policy and a schedule for regularly backing up files on public volumes.

Chapter 8 contains information on performing system backups. Refer to Section 8.4.1 for help in establishing a periodic backup schedule and using the Backup Utility.

10.2 Maintaining System Log Files

The VMS operating system provides several log files that record information about the use of system resources, error conditions, and other system events. These files include the following:

- **System dump file**

  The system dump file assists you in analyzing the cause of the system failure. In the event of a severe system failure, the VMS operating system automatically shuts down and produces a crash dump of the state of the system at the time that the error was detected. The DCL command ANALYZE/CRASH_DUMP invokes the System Dump Analyzer (SDA) for analysis of a system dump file (see Section 10.2.1).

- **Error log file**

  VMS automatically records device and CPU error messages in the error log file. The Error Log Utility invokes the Error Log Report Formatter (ERF), which selectively reports the contents of an error log file (see Section 10.2.2).

- **Operator log file**

  The Operator Communication Process (OPCOM) records system events in the operator log file (see Section 10.2.3).

- **Accounting log file**

  The accounting log file records the use of system resources and is the source of the accounting reports generated by the ACCOUNTING command (see Section 10.2.4).

10.2.1 The System Dump File

The following requirements must be met before the VMS operating system can write a complete dump file:

- The system must not be halted until the console dump messages have been printed in their entirety, and the memory contents have been written to the system dump file. Be sure to allow sufficient time for these events to take place, or make sure that all disk activity has stopped before halting the system.

- There must be a dump file in the SYS$SYSTEM directory that is named either SYSDUMP.DMP or PAGEFILE.SYS. AUTOGEN automatically creates the SYSDUMP.SYS file if there is enough disk space available.

  If SYS$SYSTEM:SYSDUMP.DMP is present, the system writes dumps to SYSDUMP.DMP. If SYS$SYSTEM:SYSDUMP.DMP is not present, the system writes dumps to SYS$SYSTEM:PAGEFILE.SYS. In this case, PAGEFILE.SYS must be at least 1004 blocks larger than physical memory, and the system parameter SAVEDUMP must be set to 1 (the default is 0). If neither file exists, the system will not generate any dumps.

The size of SYSDUMP.DMP is equal to the physical memory size plus the number of error log buffers plus 1. The number of error log buffers is controlled by the value set for the system parameter ERRORLOGBUFFERS. The range for ERRORLOGBUFFERS is from 2 to 64 with the default set to 4.

- The system parameter DUMPBUG must be set to 1 (the default is 1).

10.2.2 The Error Log File

The system automatically writes error messages to the latest version of a file named SYS$ERRORLOG:ERRLOG.SYS. You can display the information in this file by entering the DCL command ANALYZE/ERROR_LOG.

The Error Logging Facility consists of three parts:

1. A set of executive routines that detect errors and events and write relevant information into error log buffers in memory
2. A process called ERRFMT, which periodically empties the error log buffers, transforms the descriptions of the errors into standard formats, and stores the formatted information in a file on the system disk. (The ERRFMT process is started when the system is booted.)
3. The Error Log Utility, which you invoke by entering the DCL command ANALYZE/ERROR_LOG; it is used to selectively report the contents of an error log file

The executive routines and the ERRFMT process operate continuously without user intervention. The routines fill the error log buffers in memory with raw data on every detected error and event. When one of the available buffers becomes full, or when a time allotment expires, ERRFMT automatically writes the buffers to ERRLOG.SYS.

Sometimes a burst of errors can cause the buffer to fill up before ERRFMT can empty them. You can detect this condition by noting a skip in the error sequence number of the records reported in the error log reports. As soon as ERRFMT frees the buffer space, the executive routines resume preserving error information in the buffers.

The ERRFMT process displays an error message on the system console terminal and deletes itself if it encounters excessive errors while writing the error log file. To restart the ERRFMT process, first log in to the system manager's account so that you have the required privileges to perform the operation. Then execute the startup command procedure (STARTUP.COM) specifying ERRFMT as the command parameter, as follows:

```
$ @SYS$SYSTEM:STARTUP ERRFMT
```

10.2.2.1 Using Error Reports

The error reports generated by the Error Log Utility are useful in two ways:

1. They aid preventive maintenance by identifying areas within the system that show potential for failure.
2. They aid the diagnosis of a failure by documenting the errors and events that led up to it.

The detailed contents of the reports are most meaningful to DIGITAL Field Service personnel. However, you can use the reports as an important indicator of the system's reliability. For example, using the DCL command SHOW ERROR, you may see that a particular device is producing a relatively high number of errors. You can then use the Error Log Utility to obtain a more detailed report and decide whether you should consult DIGITAL Field Service. In that case, field service personnel can run diagnostic programs to investigate the device and attempt to isolate the source of the errors.

If a system component does fail, a Field Service representative can study the error reports of the system activity leading up to and including the failure. For example, if a device fails, you can generate error reports immediately after the failure. One report might describe in detail all errors associated with the device that occurred within the last 24 hours; another report might summarize all types of errors for all devices that occurred within the same time period. The summary report can put the device errors into a systemwide context. The Field Service representative can then run the appropriate diagnostic program for a thorough analysis of the failed device. Using the combined error logging and diagnostic information, the Field Service representative can proceed to correct the device.

Error reports allow you to anticipate potential failures. In turn, Field Service personnel rely on the reports as an aid to both preventive and corrective maintenance. Overall, effective use of the Error Log Utility in conjunction with diagnostic programs can significantly reduce the amount of system downtime.

10.2.2.2 Maintaining the Error Log Files

Because the error log file (SYS$ERRORLOG:ERRLOG.SYS) is a shared file, ERRFMT can write new error log entries while other entries in the same file are being read and reported by the Error Log Utility.

ERRLOG.SYS will increase in size and remain on the system disk until it is explicitly renamed or deleted. Therefore, you must devise a plan for regular maintenance of the error log file.

One method is to rename ERRLOG.SYS on a daily basis. This action causes a new error log file to be created and allows the old file (which was renamed) to be copied to a backup volume where it can be kept as long as needed. For example, you could rename the current copy of ERRLOG.SYS to ERRLOG.OLD every morning at 9:00. To free space on the system disk, you could then back up the renamed version of the error log file on a different volume and delete the file from the system disk. Note that you should exercise caution to ensure that error log files are not deleted

inadvertently. You may also want to adopt a naming convention for your files that incorporates in the file name a beginning or ending date for the data.

10.2.2.3 Printing the Error Log Files

The following steps describe how to generate an error log report for all entries in the error log file and how to print the report:

1. Ensure that you have the SYSPRV privilege. You need this privilege to access the error log file.
2. Set your default disk and directory to SYS$ERRORLOG.
3. Examine the error log directory to see which error log file you want to analyze.
4. To obtain a full report of the current error log file, enter the command:

   ```
   $ ANALYZE/ERROR_LOG/OUTPUT=ERRORS.LIS
   ```

5. Print a copy of the report, using the file name specified with the /OUTPUT qualifier:

   ```
   $ PRINT ERRORS.LIS
   ```

Example

```
$ SET PROCESS/PRIV=SYSPRV
$ SET DEFAULT SYS$ERRORLOG
$ DIRECTORY

Directory SYS$SYSROOT:[SYSERR]

ERRLOG.OLD;2  ERRLOG.OLD;1  ERRLOG.SYS;1

Total of 3 files.

$ ANALYZE/ERROR_LOG/OUTPUT=ERRORS.LIS ERRLOG.OLD
$ PRINT ERRORS.LIS
```

The directory command lists all the files in the SYS$ERRORLOG directory. In this example the directory contains three files, two old error log files and the current error log file, ERRLOG.SYS. The ANALYZE/ERROR_LOG command requests that a full report be written to a file called ERRORS.LIS, using the most recent ERRLOG.OLD file as input.

10.2.3 The Operator Log File

The operator log file (SYS$MANAGER:OPERATOR.LOG) records system events and user requests sent to the operator terminal by the operator communication process (OPCOM), even when all operator terminals have been disabled. By default, OPCOM is started when your system is booted unless you have a standalone MicroVAX processor. You use the operator log file to anticipate and prevent hardware and software failures, and to monitor user requests for disk and magnetic tape operations. The following message types appear in the operator's log file:

- Initialization of the operator's log file
- Status reports for devices attached to the system
- Operator terminals enabled and disabled
- Volume mounts and dismounts
- User requests and operator replies
- Changes to system parameters through the SYSGEN Utility
- Security alarm messages
- DECnet–VAX status messages

Example 10–1 illustrates some typical messages found in the operator log file. By regularly examining the operator log file, you can often detect potential problems and take corrective action.

10.2.3.1 Types of OPCOM Messages

This section describes some of the messages you might find in the operator's log file.

Initialization Messages

When you enter the REPLY/LOG command, the current operator's log file is closed and a new version of that file is created and opened. All subsequent OPCOM messages are recorded in this new log file.

When a new log file is created, the first message recorded in it is an initialization message that tells when and by whom the log file was initialized. This message appears in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, logfile initialized by operator
operator-name logfile is SYS$MANAGER:OPERATOR.LOG
```

Example 10–1: Sample Operator Log File (SYS$MANAGER:OPERATOR.LOG)

```
%%%%%%%%%%  OPCOM, 15-APR-1988 22:33:54.07  %%%%%%%%%%
Operator '_ZEUS$VT333:' has been disabled, user JONES
%%%%%%%%%%  OPCOM, 15-APR-1988 22:34:15.47  %%%%%%%%%%
Operator '_ZEUS$VT333:' has been enabled, user SMITH
%%%%%%%%%%  OPCOM, 15-APR-1988 22:34:15.57  %%%%%%%%%%
operator status for '_ZEUS$VT333:'
PRINTER, TAPES, DISKS, DEVICES
%%%%%%%%%%  OPCOM, 15-APR-1988 22:38:53.21  %%%%%%%%%%
request 1, from user PUBLIC
Please mount volume KLATU in device MTA0:
The tape is in cabinet A
%%%%%%%%%%  OPCOM, 15-APR-1988 22:39:54.37  %%%%%%%%%%
request 1 was satisfied.
%%%%%%%%%%  OPCOM, 15-APR-1988 22:40:23.54  %%%%%%%%%%
message from user SYSTEM
Volume "KLATU       " mounted, on physical device MTA0:
%%%%%%%%%%  OPCOM, 15-APR-1988 22:40:38.02  %%%%%%%%%%
request 2, from user PUBLIC
MOUNT new relative volume 2 () on MTA0:
%%%%%%%%%%  OPCOM, 15-APR-1988 22:41:07.54  %%%%%%%%%%
message from user SYSTEM
Volume "KLATU       " dismounted, on physical device MTA0:
15-APR-1986 22:42:14.81, request 2 completed by operator OPA0
%%%%%%%%%%  OPCOM, 15-APR-1988 22:46:47.96  %%%%%%%%%%
request 4, from user PUBLIC
_TTB5:, This is a sample user request with reply expected.
%%%%%%%%%%  OPCOM, 15-APR-1988 22:47:38.50  %%%%%%%%%%
request 4 was canceled
%%%%%%%%%%  OPCOM, 15-APR-1988 22:48:21.15  %%%%%%%%%%
message from user PUBLIC
_TTB5:, This is a sample user request without a reply expected.
%%%%%%%%%%  OPCOM, 15-APR-1988 22:49:07.90  %%%%%%%%%%
Device DMA0: is offline.
Mount verification in progress.
%%%%%%%%%%  OPCOM, 15-APR-1988 22:49:20.22  %%%%%%%%%%
Mount verification completed for device DMA0:
%%%%%%%%%%  OPCOM, 15-APR-1988 22:49:37.64  %%%%%%%%%%
Device DMA0: has been write locked.
Mount verification in progress.
%%%%%%%%%%  OPCOM, 15-APR-1988 23:33:54.07  %%%%%%%%%%
message from user NETACP
DECnet shutting down
```

Device Status Messages

Some I/O drivers send messages to OPCOM concerning changes in the status of the devices they control. For example, when a line printer goes offline, an OPCOM message is written into the operator's log file at periodic intervals until the device is explicitly returned to online status.

The device status message appears in the operator's log file in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, device device-name is offline
```

The devices for which this message can appear are card readers, line printers, and magnetic tapes.

Terminal Enable and Disable Messages

You designate a terminal as an operator's terminal by entering the REPLY/ENABLE command from the desired terminal. OPCOM confirms the request by displaying the following message at the operator's terminal and in the operator's log file:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, operator enabled, operator terminal-name
```

This message tells you which terminal has been established as an operator's terminal and when it was established.

If a terminal has been designated as an operator's terminal for a particular function, OPCOM displays the name of that function or operator class. For example, if you enter the command REPLY/ENABLE=TAPES, OPCOM displays the following message:

```
%OPCOM, 14-JUN-1988 10:25:35.74, operator enabled, operator TTE1

%OPCOM, 14-JUN-1988 10:25:38.82, operator status for operator TTE1
TAPES
```

OPCOM confirms that the terminal is established as an operator's terminal and indicates that the terminal can only receive and respond to requests concerning magnetic tape-oriented events, such as the mounting and dismounting of tapes.

A terminal that has been designated as an operator's terminal is automatically returned to nonoperator status when the operator logs out. To return the terminal to normal (nonoperator) status without logging off, enter the REPLY/DISABLE command from the terminal. OPCOM confirms that the terminal is no longer an operator's terminal by displaying a message in the following format both at the operator's terminal and in the operator's log file:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, operator disabled, operator terminal-name
```

This message tells you which terminal has been restored to nonoperator status and when the transition occurred.

If a terminal is designated as an operator's terminal and only partial operator status is disabled, OPCOM displays a status message. This message lists which requests the terminal can still receive and respond to. This message is displayed at the operator's terminal and in the operator's log file in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, operator status for operator terminal-name
status-report
```

For example, suppose you designate a terminal as an operator's terminal that receives messages concerning magnetic tapes and disks, as well as messages intended for the special site-specific operator class known as OPER10. Later, you relinquish the terminal's ability to receive messages concerning tapes. When you enter the REPLY /DISABLE=TAPES command, OPCOM returns the following message:

```
%Opcom, 14-JUN-1988 09:23:45.32, operator status for operator TTA3
DISKS, OPER10
```

This message tells you that terminal TTA3 still receives and can respond to messages about disks and messages directed to OPER10.

Volume Mount and Dismount Messages

Perhaps the widest range of operator messages occurs with volume mounts and dismounts. See Example 10–1 for examples of messages relating to mount verification and operator-assisted mounts.

User Request and Operator Reply Messages

To communicate with you, the user enters the REQUEST command, specifying either the /REPLY or /TO qualifier.

If the user enters a REQUEST/REPLY command, the request is recorded in the operator's log file in the following format:

```
%OPCOM,dd-mmm-yyyy hh:mm:ss.cc, request request-id from user user-name
__terminal-name:, "message-text"
```

This message tells you which user sent the message, the time the message was sent, the request identification number assigned to the message, the originating terminal, and the message itself.

If the user enters a REQUEST/TO command, the request is recorded in the operator's log file in the format described above, but without a request identification number, as follows:

```
%OPCOM,dd-mmm-yyyy hh:mm:ss.cc, request from user user-name
__terminal-name:, "message-text"
```

When you respond to a user's request and specify the /TO qualifier, the response is recorded in the operator's log file in the following format:

```
response message
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, request request-id completed by
operator operator-name
```

This message indicates how the operator responded to the user's request, as well as when the response was entered and which operator responded.

When you respond to a user's request and specify the /ABORT qualifier, the response is recorded in the operator's log file in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, request request-id was canceled.
```

When you respond to a user's request using the /PENDING qualifier, the response is not recorded in the operator's log file because the request has not yet been completed (that is, the request has not been fulfilled or aborted).

When a user enters a REQUEST/REPLY command and you have disabled all terminals as operator's terminals, OPCOM records all subsequent user's requests in the log file in the format shown above, but returns a message to the user indicating that no operator coverage is available.

All other OPCOM responses to REPLY commands, except responses involving the REPLY/ENABLE, REPLY/DISABLE, and REPLY/LOG commands, are not logged in the operator's log file.

Sysgen Utility Messages

Users with CMKRNL privilege can use the Sysgen Utility to change system parameters in the running (active) system. Users with the SYSPRV privilege can use the Sysgen Utility to change system parameters in the current system. OPCOM logs all changes made to current system parameters with messages in the following format:

```
%OPCOM, dd-mmm-yyyy hh:mm:ss.cc, message from user user-name
%SYSGEN-I-WRITExxx, system-mode system parameters modified by process ID n
into file y
```

Security Alarm Messages

Security alarm messages are included in the operator log file if you enable a security operator terminal and specific alarm events with the SET AUDIT/ENABLE command. Alarm messages are sent to the security operator terminal when the selected events occur. The following is an example of a security alarm OPCOM message:

```
%OPCOM, 15-OCT-1988 12:27:52.26, security alarm on node HERA/
System UAF record modification
```

You can use the command procedure SYS$MANAGER:SECAUDIT.COM to selectively extract information from the operator's log file. Output from SECAUDIT is displayed on SYS$OUTPUT. If you want to write the records to a file, you include the file specification with the /OUTPUT qualifier. The following command writes the records to the file BREAKINS.DAT in your default directory:

```
$ @SYS$MANAGER:SECAUDIT/OUTPUT=BREAKINS.DAT
```

10.2.3.2 Maintaining the Operator Log File

The operator log file, OPERATOR.LOG, normally resides on the system disk in the [SYSMGR] directory. A new version of OPERATOR.LOG is created each time the system is rebooted (unless you have a standalone MicroVAX processor, in which case the OPCOM process is not started by default). You can also use the DCL command REPLY/LOG to create a new version of the file at any time. Note that there is one operator log file per node; it is not a shared file.

You should create new versions of the operator log file regularly and store these copies for reference. The file is in ASCII format and can be printed. The current version of the operator log file can only be accessed by creating a new log file. Section 10.2.3.3 describes how to print copies of the operator log file.

You should devise a plan for regular maintenance of these files. One way is to rename the second-highest version on a daily basis. The procedure for renaming the operator log file is the same as that described in Section 10.2.2.2 for renaming the error log file. You may want to purge outdated versions of the operator log file on a regular basis. However, you should not delete versions that have not been backed up.

If OPCOM is inadvertently deleted or suspended, or if you want to start it on a standalone MicroVAX processor, use the following method to start it manually:

First, log in to the SYSTEM account so that you have the required privileges to perform the operation. Then enter the following command to execute the startup command procedure (STARTUP.COM) specifying OPCOM as the command parameter:

```
$ @SYS$SYSTEM:STARTUP OPCOM
```

If you have a MicroVAX system, you should include this command in the site-specific startup command procedure to automatically start OPCOM each time the system reboots.

10.2.3.3 Printing the Operator Log File

Perform the following operation to produce a printed copy of the most recent version of the operator log file. (You must have OPER privilege.)

1. Use the following command to enable the terminal as an operator terminal:

   ```
   $ REPLY/ENABLE
   ```

2. Close the current log file and open a new one by entering the following command:

   ```
   $ REPLY/LOG
   ```

3. Set the default to SYS$MANAGER and enter the following command to list all versions of the file:

   ```
   $ DIRECTORY OPERATOR.LOG
   ```

4. Rename the second-highest version to OPERATOR.OLD:

   ```
   $ RENAME OPERATOR.LOG;-1 OPERATOR.OLD
   ```

   The version number, -1, specifies that the second-highest version of this file is to be renamed. The highest version number is the current operator log file.

5. Print the operator log file by entering the following command:

   ```
   $ PRINT OPERATOR.OLD
   ```

In the following example, the REPLY/LOG command closes the current log file and opens a new one; the response from OPCOM verifies that a new log file has been opened. The SET DEFAULT command sets the operator default disk to the system disk, thus enabling you to examine the files contained in the directory [SYSMGR]. You can rename the second highest version of the operator log file to

OPERATOR.OLD and then enter the PRINT command to request that this version of the operator log file (OPERATOR.OLD) be printed.

```
$ REPLY/ENABLE
$ REPLY/LOG

%%%%%%%%%%  OPCOM, 15-APR-1988 12:29:24.52  %%%%%%%%%%
logfile initialized by operator _MARS$VTA2:
logfile is SYS$MANAGER:OPERATOR.LOG

$ SET DEFAULT SYS$MANAGER
$ DIRECTORY OPERATOR.LOG

Directory SYS$SYSROOT:[SYSMGR]

OPERATOR.LOG;582          OPERATOR.LOG;581

Total of 2 files.

$ RENAME OPERATOR.LOG;-1 OPERATOR.OLD
$ PRINT OPERATOR.OLD
```

10.2.3.4 Restarting OPCOM

You can restart OPCOM if for some reason it is deleted or suspended. Simply invoke the STARTUP command procedure in the [SYSEXE] directory and specify one parameter, as in the following example:

```
$ @SYS$SYSTEM:STARTUP OPCOM
```

10.2.4 The Accounting Log File

The Accounting facility collects statistics on the use of system resources in an accounting log file SYS$MANAGER:ACCOUNTNG.DAT. This information is used to monitor system activity and charge for the use of system resources. On most VAX processors (with the exception of a standalone MicroVAX processor), the Accounting facility is enabled by default when the system is started. You can modify the SET ACCOUNTING command in the site-specific startup template (SYS$MANAGER:SYSTARTUP_V5.COM) to change the default setting.

READ access is sufficient to gain access to the accounting log file. Only a user who has the ACNT privilege can create subprocesses or detached processes in which accounting is disabled. The DCL command RUN/NOACCOUNTING disables all accounting in a created process.

A user with the OPER privilege can selectively disable various kinds of accounting throughout the system by using the DCL command SET ACCOUNTING/DISABLE.

By default, the accounting log file records each of the following activities for all users:

- Batch job termination (BATCH)
- Detached job termination (DETACHED)
- Image activation (IMAGE)
- Interactive job termination (INTERACTIVE)

- Login failures (LOGIN_FAILURE)
- User messages (MESSAGE)
- Network job termination (NETWORK)
- Print jobs (PRINT)
- Process termination (PROCESS)
- Subprocess termination (SUBPROCESS)

Use the SHOW ACCOUNTING command to display which, if any, of these activities are currently being recorded in the accounting log file.

To enable or disable the logging of one or more activities, specify the corresponding keyword in the preceding list with the /ENABLE or /DISABLE qualifier of the SET ACCOUNTING command. (If you do not specify any keywords, the /DISABLE and /ENABLE qualifiers by default disable and enable all the activities listed above.) For example, to enable the recording of login failures, specify the following:

```
$ SET ACCOUNTING/ENABLE=LOGIN_FAILURE
```

To disable the recording of print jobs, specify the following:

```
$ SET ACCOUNTING/DISABLE=PRINT
```

The following list summarizes the characteristics of the accounting log file:

- File name: ACCOUNTNG.DAT (this file is not an ASCII file; hence, it must be formatted before it is printed)
- Directory location: SYS$MANAGER
- File organization: sequential
- Record length: variable
- Record types: eight

Usually, the current version of the accounting log file is closed at the end of a billing period, and a new version is created and opened. Because the accounting file is always growing, you may want to begin a new accounting file and purge the old version regularly. To begin a new accounting file, enter the DCL command SET ACCOUNTING/NEW_FILE.

If an attempt to write to the accounting log file results in an error, the file is closed automatically and a new copy is created and opened.

10.2.4.1 Accounting Records

Accounting records contain cumulative accounts of the resources used either by processes or images set up for users, or by print symbionts that print out files for users. Each accounting record contains three fields—user name, UIC, and account name—that identify the user and establish the connection between the accounting record and a user of the system. These fields correspond to similar fields of the user's account record in the user authorization file (UAF).

As system manager, you can use the Accounting Utility to sort, select, and report the accounting records. The reports can provide valuable system management tools. Alternatively, by using the detailed accounting records provided by the system, you or perhaps a system programmer can devise programs for reporting on the use of system resources and for billing for their use.

10.2.4.2 Accounting Report Formats

The Accounting Utility uses the data from the accounting log file to produce accounting reports. Using ACCOUNTING qualifiers, you can produce a variety of report formats, choose how the reports are organized, and select specific report items. Accounting reports can serve as system management tools to indicate how the system is used, how it performs, and in some cases, how particular individuals use the system. The reports also provide a means of billing users for system resources.

By default, the output is directed to SYS$OUTPUT. However, you can specify an output file with the /OUTPUT qualifier. The three output formats used for displaying data are: brief (the default), full, and summary listings. The following example illustrates a summary output:

```
$ ACCOUNTING/SUMMARY=(ACCOUNT,USER)/REPORT=(RECORDS,ELAPSED,PROCESSOR)

From:     5-APR-1988 16:33        To:      23-APR-1988  14:18
Account       Username          Total          Elapsed          Processor
                                Records          Time               Time
-------------------------------------------------------------------------
ADMIN         JFUSCIA             128       5 19:43:47.22     0 10:03:58.09
ADMIN         JGREEN               56       0 23:14:23.01     0 00:14:55.17
DECMAIL       POSTOFFICE            2       0 00:04:01.10     0 00:00:02.89
DECNET        NETMGR                1       0 00:01:31.17     0 00:00:02.81
DECNET        NETNONPRIV         2443       2 09:01:15.10     0 01:09:42.61
FIELD         FIELD                31       0 05:18:16.50     0 00:09:41.59
MANUF         BPURPLE              37       1 02:38:45.03     0 02:23:35.42
MANUF         JBROWN              227       4 04:35:07.25     0 04:30:40.60
```

Chapter 11

System Security Issues

As the person responsible for the day-to-day system management, you play an important role in ensuring the security of your system. Therefore, you should familiarize yourself with the security features available with the VMS operating system and implement the features needed to protect systems, users, and files from damage caused by tampering. This chapter outlines the security features available with the VMS operating system and suggests procedures to reduce the threat of a break-in on your system or cluster. Topics discussed in this chapter include the following:

- Setting up a site security policy
- Managing passwords
- Controlling break-in detection
- Displaying the break-in database
- Protecting files and directories with ACLs
- Creating a project account
- Security auditing

11.1 Defining a Site Security Policy

Before you begin implementing VMS security features, formulate a clear security policy for your site. Make sure that the policy outlines physical, as well as software, security requirements. Educate your users as to the security policies in effect at your site.

Each site has unique security requirements. Some sites may need limited measures because they are able to tolerate some forms of unauthorized access with little adverse effect, while other sites cannot tolerate even the slightest intrusion.

11.1.1 Types of Computer Security Problems

The source of a security breach on a computer system can usually be traced to one of three categories: user irresponsibility, user probing, or user penetration. The site security policy should describe methods to combat each type of security breach.

User irresponsibility refers to situations where the user purposely or accidently causes some noticeable damage. An example would be a user who is authorized to access certain files making a copy of a key file to sell.

There is little that an operating system can do to protect sites from this source of security failures, since the perpetrator is often a *trusted* user. Physical security measures can, however, discourage this type of security problem. Security auditing (see Section 11.6) can also be used to detect unusual system activity by privileged users.

User probing refers to situations where a user exploits insufficiently protected parts of the system. Some users consider gaining access to a forbidden system area as an intellectual challenge, playing a game of user-versus-system. Although intentions may be harmless, theft of services is a crime. Users with more serious intent may seek confidential information, attempt embezzlement, or even destroy data by probing. Always treat user probing seriously.

The VMS operating system provides many security features to combat user probing. Based on security needs, the security manager implements features on either a temporary or a permanent basis. These features are discussed in later sections.

Penetration refers to situations where the user breaks through security controls to gain access to the system. While the VMS operating system has security features making penetration extremely difficult, it is impossible to make any operating system completely impenetrable.

A user who succeeds in penetrating a system may be both skilled and malicious. Thus, penetration is a potentially serious and dangerous type of security breach. With the proper implementation of VMS security features, however, a system manager can severely limit the opportunities for break-ins.

11.2 Managing Passwords

A site needing average security protection always requires use of passwords. Sites with more security needs frequently impose a double password scheme (see Section 11.2.3) requiring primary and secondary passwords, and possibly system passwords as well.

This section describes password management.

11.2.1 Initial Passwords

When you open an account for a new user with the Authorize Utility, you must give the user a user name and an initial password. When you assign temporary initial passwords, observe all guidelines recommended in Section 11.2.6. You may want to use the automatic password generator. Avoid any obvious pattern when assigning passwords.

To use the automatic password generator while using the VMS Authorize Utility to open an account, add the /GENERATE_PASSWORD qualifier to either the ADD or the COPY command. The system responds by offering you a list of automatically generated password choices. Select one of these passwords, and continue setting up the account.

When you add a new user to the UAF, you may want to define that user's password as having expired previously using the AUTHORIZE qualifier /PWDEXPIRED. This forces the user to change the initial password when first logging in. The system behaves just as if the password had reached its expiration date, as described in Section 11.2.4.

Pre-expired passwords are conspicuous in the UAF record listing. The entry for the date of the last password change carries the following notation:

<pre-expired>

By default, the VMS operating system forces new users to change their password the first time they log in. Your site should be encouraged to use a training program for its users that includes information about changing passwords frequently and other techniques that promote system security.

11.2.2 System Passwords

System passwords are used to control access to terminals that might be targets for unauthorized use, as follows:

- All terminals using dialup lines or public data networks for access
- Terminals on lines that are publicly accessible and not tightly secured, such as those at computer laboratories at universities
- Terminals not frequently inspected
- Terminals intended for use only as spare devices
- Terminals the security manager wants to reserve for security operations

Implementing system passwords is a two-stage operation involving the DCL commands SET TERMINAL and SET PASSWORD. First, you must decide which terminals require system passwords. Then, for each terminal, you enter the DCL command SET TERMINAL/SYSPWD/PERMANENT. When you are satisfied that you have selected the right terminals, incorporate these commands into the site-specific startup command procedure so that the terminal setup work is

done automatically at system startup time. You can remove the restriction on a terminal at any time by invoking the DCL command SET TERMINAL/NOSYSPWD /PERMANENT for that terminal.

Then choose a system password and implement it with the DCL command SET PASSWORD/SYSTEM, which requires the SECURITY privilege. This command prompts you for the password and then asks you to reenter it for verification, just as is done for user passwords. To request automatic password generation, include the /GENERATE qualifier.

To enable the use of the system password for the remote class of logins (those accomplished through the DCL command SET HOST), set the appropriate bit in the default terminal characteristics parameter using SYSGEN. This is bit 19 (hexadecimal value 80000) in the parameter TTY_DEFCHAR2. Note that if you set this bit, you must invoke the DCL command SET TERMINAL/NOSYSPWD/PERMANENT to disable system passwords for each terminal where you do not want the feature. (As before, consider placing the SET TERMINAL commands you have tested in the site-specific startup command procedure.) Follow the steps in the preceding paragraph to set the system password.

When choosing a system password, select a non-English string of characters and digits, with a minimum length of 6. The system password is not subject to expiration. Change the password frequently. Always change the system password as soon as a person who knows the password leaves. Share the system password only with those who need to know.

The system password is stored in a separate UAF record and cannot be displayed. The DCL command SET PASSWORD/SYSTEM (the normal means of setting and changing the system password) requires that you enter the old system password prior to changing it. Use the AUTHORIZE command MODIFY/SYSTEM_PASSWORD to change the system password without having to specify the old password, as shown in the following command:

```
UAF> MODIFY/SYSTEM_PASSWORD=ABRACADABRA
```

The primary function of the system password is to form a first line of defense for publicly accessible ports and to prevent potential intruders from learning the identity of the system. However, requiring system passwords can appear unfriendly when authorized users are unaware that they are required on certain terminals. To avoid false reports of defective terminals or systems, inform your users which terminals allocated for their use require system passwords.

Where system passwords are not applied to either control access through dialup lines or on publicly accessed lines, few people might know the system password. There is the possibility of encumbered operations if the personnel who know the password are unavailable, incapacitated, or forget the password. Solve this problem by invoking AUTHORIZE and entering the MODIFY/SYSTEM_PASSWORD command. SYSPRV privilege is required.

11.2.3 Primary and Secondary Passwords

The use of dual passwords is cumbersome and mainly needed at sites with high-level security concerns. Dual passwords offer three advantages: when used on a widespread basis, they facilitate the verification of the physical identity of each user at login time through visual contact; when used in limited cases, they single out accounts that can only be logged in to when two persons are present; they also prevent accounts from being accessed through DECnet using simple access control.

Sites with medium security requirements might want to use dual passwords as a tool when there are unexplained break-ins after the password has been changed and the use of the password generator has been enforced. Select problem accounts, and make them a temporary target of this restriction. If the problem goes away when you institute personal verification through the secondary password, you know you have a personnel problem. Most likely, the authorized user is revealing the password for the account to one or more other users who are abusing the account.

Implement dual passwords with the AUTHORIZE qualifier /PASSWORD. For example, to impose dual passwords on a new account, invoke AUTHORIZE and enter the following command:

```
UAF> ADD newusername /PASSWORD=(primarypwd, secondarypwd)
```

To impose a secondary password on an existing account, enter the following command:

```
UAF> MODIFY username /PASSWORD=("", secondarypwd)
```

This command does not affect the primary password that already exists for the account, but adds the requirement that a secondary password be provided at each subsequent login. The secondary password acquires the same password lifetime and minimum length values in effect for the primary password. If the /FLAGS=GENPWD qualifier has been specified for this account, the secondary password can be changed only under the control of the automatic password generator.

NOTE: While secondary passwords can be specified for accounts requiring remote access using the DCL command SET HOST, they cannot be specified for accounts requiring network file access using access control strings. Do not specify secondary passwords on accounts that require network access, or request remote security managers to set up proxy accounts for those users requiring file access to other nodes in the network.

11.2.4 Enforcing Minimum Password Standards

Security managers can use AUTHORIZE to impose minimum password standards for individual users. Specifically, qualifiers and login flags provided by AUTHORIZE control the minimum password length, how soon passwords will expire, and whether the user is forced to change passwords at expiration.

Password Expiration

With the AUTHORIZE qualifier /PWDLIFETIME, you can establish the maximum length of time that can elapse between password changes before the user will be forced to change the password or lose access to the account. By default, the value of /PWDLIFETIME is 180 days. You can change the frequency requirements for user password changes by specifying a different delta time value for the qualifier. For example, to require a user to change the password every 60 days, you would specify the qualifier as /PWDLIFETIME=60-0.

The /PWDLIFETIME qualifier applies to both primary and secondary user passwords, but not to the system password. Each primary and secondary password for a user is subject to the same maximum lifetime. However, the passwords can change at separate times. As soon as the user completes a password change, that individual password's clock is reset; the new password value can exist unchanged for the length of time dictated by /PWDLIFETIME.

The use of a password lifetime forces the user to change the password regularly. The lifetime can be different for different users. Users with access to critical files generally should have the shortest password lifetimes.

System passwords have an unlimited lifetime. Therefore, change the system password regularly.

Forcing Expired Password Changes

By default, users are forced to change expired passwords when logging in. Users whose passwords have expired are prompted for new passwords at login. This password feature is only valid when a password expiration date is specified with the /PWDLIFETIME qualifier.

To disable forced password changes, specify the following qualifier to the ADD or MODIFY command:

```
/FLAGS=DISFORCE_PWD_CHANGE
```

Once disabled, the forced password feature can be reenabled by clearing the login flag, as shown in the following example:

```
/FLAGS=NODISFORCE_PWD_CHANGE
```

Users who log in and are prompted to change expired passwords can abort the login by pressing CTRL/Y.

NOTE: If secondary passwords are in effect and both primary and secondary passwords have expired, the user is forced to change both passwords. If the user changes the primary password and presses CTRL/Y before changing the secondary password, the user is logged out, and no password change is recorded.

Minimum Password Length

With the AUTHORIZE qualifier /PWDMINIMUM, you can direct that all password choices must be a minimum number of characters in length. Users can still specify passwords up to the maximum length of 31 characters.

This length applies both to primary and secondary passwords and is only required when users change passwords with the DCL command SET PASSWORD. As security manager, you can specify initial passwords through AUTHORIZE that are shorter than the minimum. However, doing so may confuse your users unnecessarily. Furthermore, initial passwords inherently introduce security weaknesses. By selecting short initial passwords, you compound the problem. Generally, it is good practice to observe the same rules you expect your users to follow.

There is always a minimum password length in effect for each user. It is either the default of 6 or another value established by the /PWDMINIMUM qualifier. Thus, if the user specifies the DCL command SET PASSWORD/GENERATE=n to generate new password choices automatically, *n* must be a value at least as great as the minimum value in effect. If *n* is less than the current minimum enforced in the UAF, it is disregarded; no message appears. The five password choices that the VMS operating system generates for the user comply with the current minimum password length.

The password generator creates passwords that range in length between *n* and *n*+2, where *n* is the specified or minimum length. In addition, the maximum values for *n* and *n*+2 that the password generator can accommodate are 10 and 12, respectively. Longer passwords require an inordinate amount of CPU time to generate.

The system password is not subject to a minimum length. Guidelines that apply to user passwords are equally applicable to system passwords. Choose system passwords that are 6 to 10 characters long.

11.2.5 Requiring the Password Generator

The /FLAGS=GENPWD qualifier in AUTHORIZE allows you to force use of the automatic password generator when a user changes a password. At some sites, all accounts will be created with this qualifier. At other sites, the security manager may be more selective.

Criteria for requiring use of the password generator should be whether or not the user will have access to sensitive data that must not be compromised by a break-in.

If your policy is to request voluntary use of the password generator, and users are not cooperating, you can force users to use the password generator by adding the /FLAGS=GENPWD qualifier to most or all user accounts. You can also add the AUTHORIZE qualifier /FLAGS=LOCKPWD to user accounts to prevent users from changing passwords. Only you as system manager will be authorized to change passwords.

11.2.6 Protecting Passwords

Observe the following guidelines to protect passwords:

- Make certain the passwords on the standard accounts SYSTEM, USER, and USERP are secure and changed regularly.
- Disable any accounts that are not used regularly, including the USER and USERP accounts, with the AUTHORIZE qualifier /FLAGS=DISUSER.
- Do not permit an outside or an in-house service organization to dictate the password for an account they use to service your system. Such service groups tend to use the same password on all systems, and their accounts are usually privileged. On seldom-used accounts, set the AUTHORIZE flag DISUSER, and only enable the account when it is needed. You can also change the password immediately after each use, and notify the service group of the new password.
- Delete accounts no longer in use.
- If you have an account on a system that stores passwords in plaintext (unencrypted), choose a different password on all of your other accounts.
- Do not leave listings where they could be read or stolen.
- Maintain adequate protection of authorization files. Note that the system user authorization file (SYSUAF.DAT) and network proxy authorization file (NETPROXY.DAT) are owned by the system account ([SYSTEM]). There should be no other users in this group. Accordingly, the categories SYSTEM, OWNER, and GROUP are synonymous. Normally the default UIC-based file protection for these authorization files is adequate.

Following are actions not strictly for password protection, but that reduce the potential of password detection or limit the extent of the damage if passwords are discovered or bypassed:

- Avoid giving multiple users access to the same account.
- Protect telephone numbers for dialup lines connected to your system.
- Make all accounts that do not require a password captive accounts.
- Extend privileges to users carefully.
- Ensure that the files containing components of the operating system are adequately protected.

11.3 Controlling Break-In Detection

This section describes how to set up break-in detection and evasion and how to display the break-in database.

11.3.1 Controlling the Number of Retries on Dialups

You can control the number of login attempts the user is allowed through a dialup line. If the user makes a typing mistake after obtaining the connection, the user does not automatically lose the connection. This option is useful for authorized users, while still restricting the number of unauthorized attempts.

To implement control of retries, use the following two *LGI parameters* provided with SYSGEN: LGI_RETRY_TMO and LGI_RETRY_LIM. If you do not change the parameters, the default values allow the users three retries with a 20-second interval between each. This means that users will lose the connection only if they fail to specify a valid password in three tries, or they spend more than 20 seconds between two of their tries.

Note that these values apply to every user on the system who is permitted to access the system through a dialup line.

The following example illustrates setting the total number of retry attempts to six, allowing a half-minute interval between tries. Since these LGI parameters are dynamic, you could change them and test them before performing the SYSGEN command WRITE CURRENT and rebooting the system.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  SET LGI_RETRY_LIM 6
SYSGEN>  SET LGI_RETRY_TMO 30
SYSGEN>  WRITE ACTIVE

{OPCOM messages show modification has been made}

SYSGEN>  EXIT
$
```

11.3.2 Controlling Break-In Detection and Evasion

Section 11.3.1 shows how to control the number of login retries for users dialing in. By limiting the number of retries to a reasonable number on each dialup login, you make the job of dialing up and trying every password combination more difficult for outsiders.

You should keep in mind that controlling dialup retries is only a part of an overall security program and is not, in itself, sufficient to avoid break-ins. An obstacle like redialing is not going to prove an effective deterrent to a persistent intruder; moreover, this technique applies only to dialups.

The VMS operating system offers additional methods of discouraging break-in attempts. These methods also use SYSGEN parameters in the LGI category. One of the parameters (LGI_BRK_LIM) defines a threshold count for login failures. When the count of login failures exceeds the LGI_BRK_LIM value within a reasonable time interval, the system assumes a break-in is in progress. Only login failures caused by specifying invalid passwords are counted, and they must be from a specific source. That source can be any of the following combinations:

- A specific terminal and a specific valid user name. As described in a following section, you can override this default to count failures by user name only. Attempted logins using invalid user names never trigger break-in detection; however, they are counted together as a single class per terminal and are used to trigger security alarms. (See Section 11.6 for information about security alarms.)
- A specific remote node and a specific remote user name.
- The user name of the creator of a detached process.

By default, LGI_BRK_LIM permits five failed login attempts from one of these sources. (Security managers can adjust the value of LGI_BRK_LIM with SYSGEN.)

The SYSGEN parameter LGI_BRK_TERM controls the association of terminals and user names for counting failures. By default, the VMS operating system sets this parameter to 1 so that they are tracked together. If you set this parameter to 0, the terminal is not included in the association; the failures associate on user name only. This feature is useful if you use terminal servers, switches, or similar facilities in which the terminal name that VMS login records does not provide a good indication of the identity of the actual terminal.

Another key parameter, LGI_BRK_TMO, controls the time period in which login failures are detected and recorded. The initial failure on each source is given an expiration time that represents the current time plus the delta time given by LGI_BRK_TMO. Each additional failure on that source adds another delta of LGI_BRK_TMO to that entry, thus extending the length of time that breakin detection is in effect. The cumulative effect is that the more failures made by a source, the greater the window of time in which additional failures will count toward the critical number defined by LGI_BRK_LIM. If no more failures occur by the time the expiration point is reached, the number of accumulated failures for that source is reset to zero. Note, however, that the failure count is not reset by a successful login.

For example, assume the default values are in effect. LGI_BRK_LIM specifies no more than five login failures from one source. LGI_BRK_TMO is set for five minutes. Assume that an outsider starts sending user names and passwords to the system. When the first password fails, the clock starts to run and the user has four more tries in the next five minutes. When the second attempt fails about 30 seconds later, the user has three tries left that will be counted over the next 9.5 minutes. When the third attempt fails 30 seconds later, the login failure observation time extends to 14 minutes. The fourth failure occurs about one minute later; the fifth failure occurs within another 30 seconds. By this time, the observation time has reached 22.5 minutes. As a result, the next login failure from that source within 22.5 minutes will trigger evasive action.

The system tolerates an average rate of login failures that is the reciprocal of the parameter LGI_BRK_TMO. For example, if the default value of LGI_BRK_TMO (300 seconds or five minutes) is in effect, the average rate of tolerable login failures is one every five minutes. When the rate of login failures exceeds the tolerable rate, and the critical number of five failures is reached (the default value of LGI_BRK_LIM), the system concludes a break-in is in progress and initiates evasive action.

The system stops accepting logins from the offending source for a period of time. When the source is a terminal (when LGI_BRK_TERM equals 1), for a period of time no one can log in from that terminal with the user name that is under suspicion. (However, other users may log in from that terminal.) A remote user triggering break-in evasion is prohibited from logging in from that node for a period of time. Consequently, login attempts that provide valid user name and password combinations that should otherwise succeed are rejected during this interval, but only from the presumed intruder at that source. Once the interval elapses, operations return to normal. As a result of this form of evasive action, outsiders are less likely to learn the correct password by using repetitive login attempts.

The duration of the evasive action is controlled by the LGI_HID_TIM parameter. The length of time depends on an additional random number (in the range of 1 to 1.5) used as a multiplier. The product of LGI_HID_TIM and the random number yields the actual duration of evasive action. The formula could be represented as follows:

```
Evasion time = LGI_HID_TIM * (random number)
```

The inclusion of a random amount of time helps obscure the true evasion time. An outsider who learned the value of LGI_HID_TIM could not be assured that the evasive action would persist for exactly that length of time.

The parameters described in the previous sections affect all terminals, users, and nodes that access the system. Because these parameters are dynamic, you can reset them without rebooting the system.

If the values of LGI_BRK_LIM and LGI_BRK_TMO can be learned or guessed, the outsider can attempt a system break-in over sufficiently long intervals that suspicion is not triggered. The outsider can also change terminals, nodes, and user names frequently enough to avoid detection. Do not rely on these break-in techniques as the sole means of security on your system.

The technique of counting failures per terminal and user name raises the potential for break-in because the password guess rate for a particular user name is multiplied by the number of available terminals. Each terminal is counted as a separate source for break-in detection. The benefit of this approach, however, is that it sharply reduces the denial of service problem that could result from simply counting failures per terminal or per user name. (A malicious user could disable an entire terminal room or user's account for a period of time if failures are counted for each user name alone.)

By setting LGI_BRK_TERM to 0, you can detect attempts more quickly, at the expense of increasing the risk of denial of service to legitimate users.

The SYSGEN parameter LGI_BRK_DISUSER makes the effects of break-in detection more severe. If you set this parameter to 1, the VMS operating system sets the DISUSER flag in the UAF record for the account where the break-in was attempted. Thus, that user name is disabled until you manually intervene. However, the service denial effects of this option can be very severe. A malicious user can put all known accounts, including yours, out of service in a short time. To recover, you must log in on the system console where the SYSTEM account is always allowed to log in. The VMS operating system stores information in the break-in database about login failures that originate from a specific source.

11.3.3 Displaying the Break-In Database

Use the DCL command SHOW INTRUSION to display the contents of the break-in database and the DELETE/INTRUSION_RECORD command to remove entries from the break-in database. See the *VMS General User's Manual* for additional information about these commands. Entries in the break-in database have the following format:

```
Intrusion      Type      Count      Expiration      Source
```

The information provided in the fields in each entry is as follows:

Intrusion	Class of intrusion.
Type	Severity of intrusion as defined by the threshold count for login failures. The SYSGEN parameter, LGI_BRK_LIM, defines the threshold count.
Count	Number of login failures associated with a particular source.
Expiration	Absolute time when the VMS operating system stops keeping track of login failure. The SYSGEN parameter, LGI_BRK_TMO, controls this time.
Source	Origin of the login failure.

The information in the break-in database is controlled by the SYSGEN parameters in the LGI category.

11.4 Protecting Files and Directories with ACLs

The VMS operating system offers two primary protection mechanisms. The first, *standard UIC-based protection,* is based on the user identification code (UIC) and is applied to all user files. It controls access to files according to the user categories SYSTEM, OWNER, GROUP, and WORLD.

The second file protection mechanism uses *access control lists (ACLs),* which employ a more refined level of protection on files than that available with UIC-based protection. ACLs can be used to grant or deny file access to individual users or groups of users, independent of the UIC.

It is assumed that you are familiar with the default file protection available with the UIC-based protection scheme. Refer to the *VMS General User's Manual* for more information on UIC-based protection.

ACLs are important file protection tools available to all VMS users and are generally used at sites with medium to high security requirements. ACLs are also prevalent in environments with complex patterns of file sharing. As security requirements increase, so does the use of ACLs.

ACLs consist of access control list entries (ACEs) that grant or deny access to system objects, such as files and devices. Each ACE specifies a user or group of users and the type of access permitted. ACLs define access more precisely than the default UIC-based protection scheme by allowing you to create groups of users independent of the users' UICs.

The VMS operating system provides a file called a *rights database* that contains a list of special names called *identifiers* as well as a list of the users specified as *holders* of identifiers. The security manager uses the VMS Authorize Utility to maintain the rights database, adding and removing identifiers and holders of identifiers as necessary. By allowing groups of users to hold identifiers, the manager has created a group designation that differs from the one used with the user's UIC. This alternative method of grouping is more finely tailored to the uses the holders of the identifier are expected to make of the objects. This method also permits each user to be a member of multiple overlapping groups.

Each time you log in, the system creates a *process rights list* for you containing a list of the identifiers in the rights database associated with your process. When you attempt to access objects protected with ACLs, the system searches the object's ACL for an identifier granting access that matches one of the identifiers in your process rights list.

The following sections describe the relationship between ACLs and identifiers in more detail.

11.4.1 Creating and Maintaining ACLs

Use the VMS ACL Editor to create and edit an ACL on a specific object. You can also use the DCL command SET ACL to manipulate (add, delete, or copy) entire ACLs or individual ACEs on more than one object at a time.

The following DCL commands can be used to display ACLs:

- SHOW ACL
- DIRECTORY/ACL
- DIRECTORY/SECURITY
- DIRECTORY/FULL

In general, you will find the DCL commands SET ACL and SHOW ACL sufficient for creating and displaying most ACLs, although the ACL editor is an important utility for more extensive ACL work.

11.4.2 Identifiers

Identifiers in an ACL specify the users who are allowed or denied access to an object. Following are the three types of identifiers:

- *UIC identifiers*—depend on the user identification codes (UICs) that uniquely identify each user on the system. Typically the UIC identifiers are presented in numeric or abbreviated alphanumeric format. For example, a UIC identifier might adopt the numeric format of the UIC, such as [306,210], or just the member name from the alphanumeric format UIC, such as JONES, where the full alphanumeric UIC is [GROUP1,JONES].
- *General identifiers*—defined by the security manager in the system rights database to identify groups of users on the system. For example, TERM3BIO, WARD5WORKERS, DATAENTRY, and RESERVDESK would identify the third term biology students, the campaign workers for Ward 5, the data entry personnel, or the people who handle the reservations desk, respectively.
- *System-defined identifiers*—describe certain types of users based on their use of the system. For example, BATCH, NETWORK, DIALUP, INTERACTIVE, LOCAL, and REMOTE correspond directly to the type of login the user executed.

When you log in, the identifiers you hold in the rights database (including your UIC and your system-defined identifiers) are copied into a rights list that is part of your current process. The rights list is the structure that the VMS operating system uses to perform all protection checks. Additional identifiers may appear in your rights list; they were put there either by VMS Login software or by software specific to your installation. These identifiers represent qualifications about your login and the state of the system.

11.4.2.1 UIC Identifiers

While the most common types of UIC identifiers are either numeric format UICs or user names, full alphanumeric UICs or UICs in hexadecimal format are accepted as UIC identifiers. Thus, you might see the following UIC identifiers:

```
[PROGRAMMERS,J_JONES]      {alphanumeric format UIC}
J_JONES                   {username from alphanumeric format UIC}
[341,311]                 {numeric format UIC}
%X08001006                {hexadecimal format UIC}
```

Each of these formats uniquely identifies a user.

The system automatically adds a UIC identifier to the system rights database when each new account is created.

11.4.2.2 General Identifiers

A general identifier, defined in the system rights database, is an alphanumeric string of 1 to 31 characters that must contain at least one alphabetic character. It can include the characters A through Z, dollar signs ($), underscores (_), and the numbers 0 through 9.

Use the Authorize Utility (AUTHORIZE) to create general identifiers in the system rights database and to assign them to system users, as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD/IDENTIFIER PROJECTX
UAF> GRANT/IDENTIFIER PROJECTX WILLIAMS
```

Refer to the Authorize Utility in Part II for descriptions and examples of all AUTHORIZE commands.

11.4.2.3 System-Defined Identifiers

System-defined identifiers are automatically defined by the system when the rights database is created at system installation time. The following identifiers, which correspond directly to the possible types of login classes, are system-defined:

BATCH	All access attempts made by batch jobs.
NETWORK	All access attempts made by DECnet tasks.
INTERACTIVE	All access attempts made by interactive processes.
LOCAL	All access attempts made by users logged in at local terminals.
DIALUP	All access attempts made by users logged in at dialup terminals.
REMOTE	All access attempts made by users logged in through a network.

In addition, a system node identifier of the form SYS$NODE_*node_name* is created by the site-independent startup procedure (STARTUP.COM in SYS$SYSTEM).

A user automatically becomes a holder of one or more of these identifiers during login. The VMS Login software adds the appropriate identifiers to the process rights list.

11.4.3 Access Control List Entries

As shown in Section 11.4.2.2, you create identifiers in the rights database and assign users as holders of the identifiers. You then define which access to grant or deny holders of the identifier for each object (such as a file) requiring this level of protection. Because several identifiers may be required to represent access needs for each object, it is typical to create a list of multiple entries. Each entry defines the access rights to be granted or denied the holders of the identifier named in that entry. This list is the Access Control List, or ACL. Each entry in this list is called an access control list entry (ACE).

Like the defaults for UIC-based protection, you can set up default ACLs. As a result, some users may be unaware that their files have ACLs and may never change ACLs themselves. Other users are actively involved in creating and maintaining their own ACLs.

To summarize, ACLs can be created by the system by default, by you for specific objects, and by users to protect their own files. Users can create ACLs only for objects they own (typically files residing in their directories) or to which they have the same access as the object owner.

An ACL consists of ACEs that grant or deny access to a particular system object, such as a file, directory, or device. Because ACLs can define access more selectively than UIC-based protection, ACLs allow users to fine tune the action taken when access is requested for an object. Typically, you use ACLs to provide users from several UIC groups access to a system object without having to grant WORLD access to the object. ACLs can perform other functions, such as directing security alarms to be set off when access to an object succeeds or fails.

When the system receives a request for access to an object that has an ACL, the system searches each entry in the ACL sequentially for the first match. It stops searching at the first match. If another match exists further down in the ACL, it has no effect. Thus, ACEs that identify specific users should appear in the ACL before ACEs that identify broader classes of users, as follows:

```
(IDENTIFIER=WILLIAMS,ACCESS=READ+EXECUTE)
(IDENTIFIER=CS101,ACCESS=NONE)
```

Assume that user WILLIAMS holds the CS101 identifier. In the previous example, WILLIAMS is granted READ and EXECUTE access to the object. If the ACEs were switched, user WILLIAMS may be denied access to the object.

The use of ACLs is optional. Although the use of ACLs can enhance the security of system objects in any installation through a more detailed definition of who is allowed what kind of access, user time must be spent in creating and maintaining the ACLs, and processor time is required to perform the functions that ACLs mandate.

Each ACL consists of one or more ACEs. There is no limit to the number of ACEs that an ACL can contain or to the number of characters in an ACE; however, very long ACLs increase the amount of time necessary to gain access to an object.

The most common type of ACE is the Identifier ACE, which controls the type of access allowed to a particular user or group of users.

In general, the format of an ACE is as follows:

(type[,options][,access_to_grant])

11.4.3.1 Identifier ACE

An identifier ACE controls the types of access allowed to specific users based on user identification. Following is the format for an identifier ACE:

(IDENTIFIER=identifier[,options][,access])

Specifying Identifiers in Identifier ACEs

The first field in the identifier ACE is the keyword IDENTIFIER followed by one or more identifiers.

The system takes the access action included in the ACE only for the user who matches all the identifiers. For example, if you wanted to grant read access to user [301,25] running a batch job, you would specify the identifier ACE as follows:

(IDENTIFIER=[301,25]+BATCH,ACCESS=READ)

Although it is unusual for a number of users to share the same UIC, it is likely that a number of users will share the same general identifier. Users with the same general identifier do not need to be in the same UIC-based group. Furthermore, a single user can be associated with a number of different general identifiers as defined in the rights database. The creator of an ACL has considerable flexibility in selecting sets of users and defining access capabilities for them.

For example, the user identified by the UIC [301,25] is a member of the UIC-based group 301. That user may be the only member of group 301 who is also associated with the general identifier PERSONNEL. An ACE defining a particular type of access for the users associated with the general identifier PERSONNEL grants that type of access to that user, but not to the other members of group 301.

Specifying Options in Identifier ACEs

The options field in an identifier ACE controls whether an ACE is copied to new versions of the file, can be displayed, or can be deleted. This field in an identifier ACE begins with the keyword OPTIONS and takes one or more of the following keywords:

DEFAULT	Indicates that an ACE is to be included in the ACL of any files created within a directory. When the ACE is propagated, the DEFAULT indicator is removed from the ACL of the created file. This option is valid only for directory files. A default ACE does not grant or deny access; it just affects the ACL of new files.
HIDDEN	Indicates that this ACE should only be changed by the application that added it. The ACL editor does not permit modification or deletion. Thus, the ACL editor displays the ACE only to show its relative position within the ACL, not to facilitate editing of the ACE. The DCL DIRECTORY and SHOW ACL commands do not display hidden ACEs.
PROTECTED	Indicates that an ACE will be preserved even when an attempt is made to delete the entire ACL. A protected ACE must be deleted specifically with the ACL editor or by specifying the ACE on the command line of the DCL command SET ACL.
NOPROPAGATE	Indicates that, when copying an ACL from one version of a file to a later version of the same file, the ACE is not copied to the newer version.
NONE	Indicates that no options apply to an ACE. Although you can enter OPTIONS=NONE when you create the ACE, OPTIONS=NONE is not displayed when the ACE is displayed.

Connect multiple options with plus signs (+). If you specify any other options with the NONE option, the other options take precedence.

Identifier ACE for a Directory

The OPTIONS=DEFAULT option of an identifier ACE allows users to define one or more default ACEs for inclusion in the ACLs for files created in a particular directory. A default ACE is supplied for all new files created in that directory; any existing files are not supplied with the default ACE. Thus, if you wanted all files in the directory [MALCOLM] to have an ACE that permitted read and write access to users with the PERSONNEL identifier, you could include the following ACE in the ACL for the file MALCOLM.DIR:

```
(IDENTIFIER=PERSONNEL,OPTIONS=DEFAULT,ACCESS=READ+WRITE)
```

As a result of this ACE, any file created in the [MALCOLM] directory has the following ACE:

```
(IDENTIFIER=PERSONNEL,ACCESS=READ+WRITE)
```

Notice that the DEFAULT option does not appear in the file's ACE. However, any subdirectory created in the MALCOLM directory has the DEFAULT option as part of its ACE so that the default ACE will be propagated throughout the entire directory tree.

Specifying Access in Identifier ACEs

The third field in an identifier ACE specifies what type of access you are allowing the users identified in the first field of the ACE. This field begins with the keyword ACCESS followed by a string of access actions connected by plus signs. The following types of access are allowed in an identifier ACE:

READ	Accessor can read a file, read from a disk, or allocate a device.
WRITE	Accessor can read or write a file.
EXECUTE	Accessor can execute an image file or look up entries in a directory by explicitly specifying file names.
DELETE	Accessor can delete a file.
CONTROL	Accessor has all the privileges of the object's owner.
NONE	Accessor has no access to the object.

Sample Identifier ACEs

The most common type of ACL is one that defines the access to a file for a group of users. In the following ACL example, access to a file is based on the identity of a user. PERSONNEL, SECURITY, and SECRETARIES are general identifiers assigned to appropriate sets of users by the system manager using AUTHORIZE. NETWORK is a system-defined identifier, while [20,*] and [SALES,JONES] are examples of UIC identifiers.

```
(IDENTIFIER=SECURITY,OPTIONS=PROTECTED,ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
(IDENTIFIER=PERSONNEL,ACCESS=READ+WRITE+EXECUTE+DELETE)
(IDENTIFIER=SECRETARIES,ACCESS=READ+WRITE)
(IDENTIFIER=[20,*],ACCESS=READ)
(IDENTIFIER=NETWORK,ACCESS=NONE)
(IDENTIFIER=[SALES,JONES],ACCESS=NONE)
```

In the preceding example, the ACE providing the greatest amount of file access is listed at the top of the ACL. Any users holding both the SECURITY and PERSONNEL identifiers obtain maximum access rights through the first match, which is the SECURITY identifier. In this example, the user with UIC [SALES,JONES] is prohibited from any access to the file unless that user also happens to have one of the general identifiers (which is an oversight on the part of the creator of the ACL). If the ACL creator wants to be absolutely certain that the user with UIC [SALES,JONES] could not possibly gain access to the file, the ACE at the bottom of the ACL should be moved to the top.

The order of the ACEs in the example permits a number of users to gain types of file access over the DECnet–VAX network. The users with the identifiers of SECURITY, PERSONNEL, SECRETARIES, and UIC [20,*] can all gain some access over the network, although only those with the identifier SECURITY can gain full access. The fifth ACE prevents all other users from network access. While this might be the intent of the ACL creator, it would be an unfortunate oversight if it were not. Remember that the system searches the ACL sequentially and grants the user only the access specified in the first matching ACE. All subsequent ACEs are ignored.

The first ACE is the only ACE containing an option field (the PROTECTED option). Using this option prevents the first ACE from being deleted unless you have explicitly deleted the ACE with the ACL editor, or you have specified the ACE with the SET ACL/DELETE command.

11.4.4 Summary of ACLs

The following recommendations will help you manage ACLs:

- Do not assume that specifying ACCESS=NONE for an identifier will absolutely prohibit the holders of the identifier from accessing the object. Frequently, users in either the SYSTEM or OWNER category may still be entitled to whatever access the UIC-based protection affords that category. If the users hold special privileges, they may be granted the access requested through the privilege.
- Watch out for errors in the order that ACEs appear in the ACL. Place the ACEs that deny access to specific users at the top of your ACLs, so that the user will not obtain access by holding another identifier. Sometimes you use wildcards in the UIC-format identifiers to deny access to large groups of users. Such an ACE properly belongs at the bottom of the ACL, not at the top. Place the ACEs that grant the widest access rights immediately before the most restrictive ACEs. This technique ensures that users who hold multiple identifiers do not obtain restricted access rights on the first match when another identifier they hold could grant more generous rights. Remember that a user can only receive the access rights granted through the first matching identifier.
- Do not place ACLs on all objects. This is usually unnecessary even at medium-level security sites. Too many ACLs can cause performance penalties to appear on the system. Instead of using ACLs, group files so that only a few directories need default ACEs that propagate to many or all files.
- Use general identifiers to create practical groups of users to avoid unnecessarily long ACLs.
- Update ACLs when users leave. Always maintain the shortest and most current ACLs. Again, using general identifiers instead of individual users helps alleviate this maintenance problem.

11.5 Creating a Project Account

To allow for more flexible management and accounting of disk space, identifiers can carry the optional resource attribute. This attribute, when present on an identifier, allows file space to be owned by and charged to that identifier. Thus, when a project or department-specific identifier is the owner of a directory, the space used by files created in the directory can be charged to the appropriate department or project rather than to the individual who created them. When users work on multiple projects, they can charge their disk space requirements to the related project rather than to their personal accounts.

Another important advantage of setting up a project account is that doing so allows you, the system manager, the control the protection of the account and its files, rather than the users of the account. This assures that all files created within the project account will be adequately and uniformly protected.

Example

To set up a project identifier and directory, perform the following steps:

1. Using the VMS Authorize Utility, create the project identifier with the resource attribute in the rights database. The following example creates the identifier PROJECTX:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> ADD/IDENTIFIER PROJECTX /ATTRIBUTES=RESOURCE
```

2. Grant the identifier to the appropriate individuals with the resource attribute.

```
UAF> GRANT/IDENTIFIER PROJECTX user1 /ATTRIBUTES=RESOURCE
UAF> GRANT/IDENTIFIER PROJECTX user2 /ATTRIBUTES=RESOURCE
   .
   .
   .
```

3. Create the disk quota authorization for the project identifier. For example, the following command invokes the VMS System Management (SYSMAN) Utility and assigns the identifier PROJECTX 2000 blocks of disk quota with 200 blocks of overdraft:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> DISKQUOTA ADD PROJECTX /PERMQUOTA=2000 /OVERDRAFT=200
```

4. Create the project directory. For example, the following DCL command creates the project directory [PROJECTX] and establishes the identifier PROJECTX as the owner:

```
$ CREATE/DIRECTORY [PROJECTX] /OWNER=[PROJECTX]
```

5. Set up the necessary ACL on the project directory to allow holders of the PROJECTX identifier access to the directory. For example, the following DCL command places an ACL on the directory [PROJECTX] that permits any holder of the identifier PROJECTX to gain READ, WRITE, or EXECUTE access to the

directory. The second ACE specifies that files created in the directory will receive the same ACE as a default.

```
$ SET DIRECTORY [PROJECTX] /ACL= (-
_$ (IDENTIFIER=PROJECTX,ACCESS=READ+EXECUTE),-
_$ (IDENTIFIER=PROJECTX,OPTIONS=DEFAULT,ACCESS=READ+EXECUTE))
```

Access must be granted through ACL entries, since the owner identifier of the directory and the files does not match the UIC of any of the project members; thus, only SYSTEM and WORLD access are available through the UIC-based protection mask. The first ACE of the specified ACL gives all project members READ and EXECUTE access to the directory; the second ACE gives the same access for all files created in the directory. (The DEFAULT option in the second ACE specifies that the ACE is to be copied to each file created in the directory.)

Note that project members are not allowed to delete (or control) files created by others. However, the members each have complete access to files they have created in the directory, because the file system supplies an additional ACE that grants the file creator CONTROL access plus the access specified in the OWNER field of the UIC-based protection mask. This ACE only appears when the owner of the created file does not match the UIC of the creator, as is the case for files created in an account owned by a project identifier.

Thus, when project member CRANDALL creates files in the [PROJECTX] directory, the files receive the following access control list:

```
(IDENTIFIER=CRANDALL,OPTIONS=NOPROPAGATE,ACCESS=READ+WRITE+EXECUTE+DEFAULT+CONTROL)
(IDENTIFIER=PROJECTX,ACCESS=READ+EXECUTE)
```

This example assumes that the OWNER field grants full (RWED) access. Because this may not always be true (the systemwide default set by the SYSGEN parameter RMS_FILEPROT may have been changed, or a user may have specified a process-specific default protection mask with the DCL command SET PROTECTION /DEFAULT), you may want to ensure consistency by providing a default protection ACE in the project directory ACL, as follows:

```
$ SET DIRECTORY [PROJECTX] /ACL= (-
_$ (DEFAULT_PROTECTION,S:RWED,O:RWED,G,W),-
_$ (IDENTIFIER=PROJECTX,ACCESS=READ+EXECUTE),-
_$ (IDENTIFIER=PROJECTX,OPTIONS=DEFAULT,ACCESS=READ+EXECUTE))
```

The UIC protection specified in the default protection ACE is applied to all files created in the project directory.

11.6 Security Auditing

Security alarms are messages sent to the security operator's terminal indicating specific events. Alarms can help you detect outsiders' attempts to break into the system and can be used to monitor undesirable activity at your site. For example, you might enable an alarm that sends a message to the security operator's terminal whenever a UAF record changes.

When dealing with security alarms, carefully select and enable the events to be audited, enable a security operator terminal, and monitor and make use of the alarm information.

11.6.1 Enabling Security Alarms

Before enabling security auditing on your Local Area VAXcluster, ensure that the Operator Communications (OPCOM) process has been started in your site-specific startup command procedure. (By default, OPCOM *is not* started on MicroVAX systems.) The OPCOM process is the mechanism used to write all security alarms to the operator log file.

To enable security auditing, specify the DCL command SET AUDIT in the following format:

```
SET AUDIT /ALARM /ENABLE=keyword[,...]
```

Select the events to be audited by specifying one or more of the following keywords to the /ENABLE qualifier:

- ACL—Event requested by an ACL on a file or global section
- ALL—All possible events
- AUDIT—Execution of the SET AUDIT command
- AUTHORIZATION—Modifications to the system UAF file, network proxy authorization file, rights database, or changes to system and user passwords
- BREAKIN—Successful break-in attempt
- FILE_ACCESS—Selected types of access (privileged and non-privileged) to files and global sections
- INSTALL—Installation of images
- LOGFAILURE—Failed login attempt
- LOGIN—Successful login attempt
- LOGOUT—Logout
- MOUNT—Volume mounts and dismounts

See the *VMS General User's Manual* for more information about the SET AUDIT command.

Because security auditing affects system performance, enable security alarms only for the most important events. The following security alarm features are presented in order of decreasing priority and increasing system cost:

1. Enable security auditing for LOGFAIL and BREAKIN. This is the best way to detect probing by outsiders (and insiders looking for accounts). All sites needing security should enable alarms for these events.

2. Enable security auditing for LOGIN. Auditing successful logins from the more suspicious sources like REMOTE and DIALUP provides the best way to track which accounts are being used. An audit record is written before users logging in on a privileged account can disguise their identity.

3. Enable the FILE=FAILURE security audit. This technique audits all file protection violations and is an excellent method of catching probers.

4. Apply ACL-based file access auditing to detect WRITE access to critical system files. The most important files to audit are shown in Table 11–1. You may want to audit only successful access to these files to detect penetrations, or you may want to audit access failures to detect probing as well.

   Note that some of the files in Table 11–1 are written during normal system operation. For example, SYSUAF.DAT is written during each login, and SYSMGR.DIR is written when the system boots.

5. Audit use of privilege to access files (either WRITE or all forms of access). Implement the security audit with FILE=(SYSPRV,BYPASS,READALL,GRPPRV). Note that this class of auditing can produce a large volume of output because privileges are often used in normal system operation for such tasks as mail delivery and operator backups.

Table 11–1: System Files Benefiting from ACL-Based File Access Auditing

Device and Directory	File Name
SYS$SYSTEM	AUTHORIZE.EXE
	F11BXQP.EXE
	LOGINOUT.EXE
	DCL.EXE
	JOBCTL.EXE
	JBCSYSQUE.DAT
	SYSUAF.DAT
	NETPROXY.DAT
	RIGHTSLIST.DAT
	STARTUP.COM

Table 11–1 (Cont.): System Files Benefiting from ACL-Based File Access Auditing

Device and Directory	File Name
SYS$LIBRARY	SECURESHR.EXE
SYS$MANAGER	SYSTARTUP_V5.COM
	VMSIMAGES.DAT
SYS$SYSROOT	[000000]SYSEXE.DIR
	[000000]SYSLIB.DIR
	[000000]SYS$LDR.DIR
	[000000]SYSMGR.DIR

11.6.2 Enabling a Security Operator Terminal

Before you enable alarms for particular events, enable a security operator's terminal. Choose a terminal that provides hardcopy output and is located in a secure location. The following DCL command enables the terminal from which the command is entered:

```
$ REPLY/ENABLE=SECURITY
```

Any number of terminals can be enabled as security operators. If you designate one terminal as the security operator's terminal, add the following lines to the site-specific startup command procedure (usually SYS$MANAGER:SYSTARTUP_V5.COM) to send alarms to the terminal and disable them on the system console:

```
$ DEFINE/USER SYS$COMMAND OPA0:
$ REPLY/DISABLE=SECURITY
$ DEFINE/USER SYS$COMMAND TTA3:
$ REPLY/ENABLE=SECURITY
```

Security alarms are always written to the operator log file, even if no security operator terminal is enabled.

11.6.3 Enabling Alarm Messages

After you enable a security operator terminal, enable specific alarm events with the SET AUDIT/ENABLE command. Alarm messages are then sent to the security operator terminal when the selected events occur. Security alarms appear as follows:

```
%%%%%%%%%%  OPCOM  30-DEC-1988 12:27:52.26  %%%%%%%%%% (1)
Security alarm on LASSIE / System UAF record modification (2)
 Time:  30-DEC-1988 12:27:52.25 (3)
 PID:  23C00155 (4)
 User Name: MENACE (4)
 Rec Mod: GOWER
 Fields Mod: PRIVILEGES
```

The information included in the message depends on the type of event; in all cases, the alarm message contains the following four elements:

❶ OPCOM heading, which includes the date and time the alarm was sent

❷ Type of alarm event

❸ Date and time the alarm event occurred

❹ The user who caused the event, as identified by the user name and process identification (PID)

Other information contained in alarm messages is specific to the type of event that the alarm signaled.

11.6.4 Audit Reduction Facility

If you have enabled security alarms, the operating system writes information resulting from those alarms to the security operator's log file. Because you can enable alarms for many objects and types of access, the log file often contains a large volume of information. To extract information selectively from the operator's log file, use SECAUDIT.COM, a command procedure residing in SYS$MANAGER.

To extract all of the security alarm information from the current operator's log file (SYS$MANAGER:OPERATOR.LOG), execute the following command:

```
$ @SYS$MANAGER:SECAUDIT
```

Output from SECAUDIT is displayed on SYS$OUTPUT. If you want to write the records to a file, include the file specification with the /OUTPUT qualifier. The following command writes the records to the file BREAKINS.DAT in the user's current default directory:

```
$ @SYS$MANAGER:SECAUDIT/OUTPUT=BREAKINS.DAT
```