VMS

digital

Introduction to VMS System Management

Order Number AA–LA24A–TE

Introduction to VMS System Management

Order Number: AA–LA24A–TE

April 1988

This manual introduces the concepts of VMS system management for new system managers and operators.

Revision/Update Information:	This is a new manual.
Software Version:	VMS Version 5.0

digital equipment corporation
maynard, massachusetts

April 1988

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

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

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

Copyright ©1988 by Digital Equipment Corporation

All Rights Reserved.
Printed in U.S.A.

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
DEC/CMS
DEC/MMS
DECnet
DECsystem-10
DECSYSTEM-20
DECUS
DECwriter

DIBOL
EduSystem
IAS
MASSBUS
PDP
PDT
RSTS
RSX

UNIBUS
VAX
VAXcluster
VMS
VT

digital ™

ZK3387

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

USA & PUERTO RICO*

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 and Puerto Rico call 800-258-1710.

In New Hampshire, Alaska, and Hawaii call 603-884-6660.

In Canada call 800-267-6215.

*Any prepaid 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 trademark of Adobe Systems, Inc.

Contents

INDEX

FIGURES

TABLES

Preface

The *Introduction to VMS System Management* presents an overview of the concepts needed to set up and manage a VMS operating system. This manual serves as an introductory guide for finding information in the VMS System Management documentation subkit. New system managers and operators should not attempt to perform system management tasks until they are familiar with the detailed procedures described in the subsequent manuals of the system management subkit.

Intended Audience

This manual addresses users who have a basic understanding of the VMS operating system and who are responsible for managing the daily operations of the system.

Document Structure

The *Introduction to VMS System Management* is organized into seven chapters, each introducing a different component of VMS system management.

The following chapters are included:

- **Chapter 1 -** Overview, describes system management tasks, the tools for carrying out these tasks, and VMS operating system components. Also included is a brief overview of the three types of software installation procedures.
- **Chapter 2 -** Setting Up the System, describes the concepts for customizing a VMS system, setting up user accounts, and controlling system resources.
- **Chapter 3 -** Security Management, introduces the basic concepts for controlling access to information and protecting system objects.
- **Chapter 4 -** Maintaining the System, introduces the concepts for maintaining public volumes, performing backups, managing disk space, performing batch and print operations, and analyzing error conditions.
- **Chapter 5 -** Performance Management, gives an overview of workload management concepts and provides guidelines for evaluating performance problems.
- **Chapter 6 -** VMS Clusters, gives an overview of the different types of VMS cluster configurations and special considerations for managing clusters.
- **Chapter 7 -** Networking, gives an overview of DECnet networking on a VMS system and the types of tasks that you can perform over the network.

Associated Documents

- For general background information about the system, see the *Introduction to VMS*.
- For information on system installation and other processor-specific operations, see your VAX processor installation and operations guide.
- For information on tasks for maintaining daily operations (for example, backing up public volumes and maintaining system log files), see the *Guide to Maintaining a VMS System*.
- For security management information, see the *Guide to VMS System Security*.
- For information on creating and maintaining volumes using the volume shadowing option, see the *VAX Volume Shadowing Manual*.
- For hardware operating instructions, see the appropriate hardware owner's manual.
- For managing network operations, see the *VMS Networking Manual*.
- For information on performance tuning, see the *Guide to VMS Performance Management*.
- For detailed information on utilities, see the individual VMS utility manuals.
- For supplemental reference information, see the *VMS DCL Dictionary* and the *VMS System Messages and Recovery Procedures Reference Volume*.

Conventions

Convention	Meaning
$\boxed{\text{RET}}$	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.

1 Overview

As system manager, you are responsible for managing daily operations and ensuring the efficient use of system resources. In order to make good management decisions, you should have a basic understanding of the users' needs and the system's capabilities.

1.1 System Management Responsibilities

System management tasks may be performed by one individual, or by many individuals. This manual does not make job distinctions between system managers, operators, and other specialists, because responsibilities vary from site to site.

System management tasks fall into the following general categories:

- Installing and upgrading the system
- Setting up the system for site-specific operations
- Performing periodic maintenance tasks
- Tuning the system for optimal performance
- Setting up and managing special configurations (such as clusters and networks)

See your VAX processor installation and operations guide for instructions on installing the system and other processor-specific procedures. Instructions for performing an upgrade are included in the *VMS Release Notes Manual.*

After you install the VMS operating system, you can customize the system for site-specific operation by performing the following tasks:

- Selecting a bootstrap procedure
- Creating site-specific command procedures that execute when the system is started
- Running AUTOGEN to adjust the system for special configuration or workload needs
- Setting up user accounts
- Establishing resource controls
- Establishing access controls

The VMS system normally runs with minimal operator intervention. In many installations, however, operators keep the system running smoothly by performing the following tasks periodically, or as user needs dictate:

- Physically mounting magnetic tapes and disks at the request of the users
- Initializing and mounting system volumes
- Backing up files and volumes

- Sending messages to users
- Monitoring queues and output devices
- Responding to emergencies
- Maintaining system log files
- Responding to security alarms

The following section describes some of the tools provided with the VMS operating system to help system managers do their job more efficiently.

1.2 System Management Tools

DIGITAL supplies the following software tools to monitor and control system operations and resources:

- Operator Communication Manager (OPCOM)
- DIGITAL Command Language (DCL) commands and procedures
- VMS utilities
- Access controls (see Chapter 3)
- Resource controls (see Chapter 2)

Operator Communication Manager (OPCOM)

As system manager, you can perform many of your duties from user terminals, however, some tasks must be performed from an operator's terminal. For example, you use the operator's terminal to send messages to system users and respond to user requests, using the operator communication manager (OPCOM).

Usually, the console terminal (OPA0) is designated as the operator's terminal. Functions such as bootstrapping the system and communicating with the VAX processor's console subsystem must be performed at the console terminal.

You designate a terminal as the operator's terminal by using the privileged DCL command REPLY/ENABLE. OPCOM itself is automatically enabled at system startup time, except on standalone workstation processors. To minimize physical memory and disk block usage, the OPCOM process is not created by default on a standalone workstation.

Messages that pass between you and system users are displayed on the operator's terminal (OPA0) and are recorded in the operator's log file (SYS$MANAGER:OPERATOR.LOG).

DCL Commands and Procedures

You interact with the VMS operating system by entering DCL commands and interpreting system messages. Most of the DCL commands used by system managers require special privileges (such as OPER privilege). See the specific DCL command description for the required privileges and syntax for each command.

The general format of a DCL command is as follows:

```
command-name[/qualifier[,...]] [parameter[,...]] [/qualifier[,...]]
```

Because a command can be continued on more than one line, the term "command string" is used to define the entire command that is passed to the system. A command string is the complete specification of a command, which includes the command name, command qualifiers, parameters, and parameter qualifiers.

A command procedure is a file containing DCL commands and data that the command interpreter can accept instead of the user entering each command interactively. Command procedures can greatly enhance efficiency in performing routine system management tasks. You should become familiar with the DIGITAL-supplied command procedures, described in *Guide to Setting Up a VMS System*. Also, you may want to design your own command procedures to automate some of the routine system management tasks specific to your site.

System Management Utilities

DIGITAL supplies a number of VMS *utilities* designed specifically to perform system management functions. A utility is a program that performs a set of related operations; for example, setting up user accounts or backing up files. Many of these utilities require special privileges, which are assigned by default when you log in to the system manager's account.

Some utilities, such as SYSGEN and AUTHORIZE, are invoked from the SYS$SYSTEM directory by using the following command format:

```
$ RUN SYS$SYSTEM:utility_name
```

Other utilities, such as MOUNT and ANALYZE/DISK_STRUCTURE, are invoked by entering a DCL command, for example:

```
$ ANALYZE/DISK_STRUCTURE
```

Table 1–1 lists the utilities most commonly used by system managers and gives a brief description of their functions.

Table 1–1 System Management Utilities

Utility Name	Function
ACCOUNTING	Produces reports and summaries of system usage.
ANALYZE/DISK_STRUCTURE	Checks the validity of Files–11 Structure Level 1 and Structure Level 2 disk volumes, and reports errors and inconsistencies.
AUTHORIZE	Adds and modifies records in the existing user authorization and network authorization files or creates new files; adds and modifies records in the rights database.
BACKUP	Performs backup operations on disk and tape volumes and files.
ANALYZE/MEDIA	(Bad Block Locator Utility) Analyzes block-addressable devices and records the location of blocks that cannot reliably store data.
ANALYZE/ERROR_LOG	(Error Logger Utility) Reports the contents of the system error log file.

Table 1–1 (Cont.) System Management Utilities

Utility Name	Function
EXCHANGE	Transfers data to and from mass storage volumes that are written in formats other than standard formats recognized by VMS.
INSTALL	Installs and maintains known images.
MOUNT	Makes a disk or magnetic tape volume available for processing.
NCP	A DECnet–VAX utility, Network Control Program (NCP) accepts terminal commands to configure, control, monitor, and test a DECnet network.
SYSGEN	Performs tasks associated with system generation such as loading and connecting drivers, creating or extending page and swap files, and displaying or modifying the values of the system parameters.
SYSMAN	Allows you to define a system management environment (such as a node or a cluster) so that operations performed from a local node are executed on all other member nodes in the environment. Other system management utilities (such as DISKQUOTA and SYSGEN) can be executed from within the SYSMAN Utility.

1.3 VMS Operating System Components

To manage a VMS system effectively, you must be familiar with its principal components, which are contained in system directories on the VMS system disk. The logical names of these directories and brief descriptions of their contents are listed as follows:

- SYS$ERRORLOG—Contains the error log file (ERRLOG.SYS)
- SYS$EXAMPLE—A subdirectory of SYS$HELP, which includes sample driver programs, user-written system services, and other example source programs
- SYS$HELP—Contains text files and help libraries for the Help Utility
- SYS$INSTRUCTION—Contains EDT computer-aided instruction (EDTCAI) files
- SYS$LIBRARY (also SYS$SHARE)—Contains various macro and object libraries and shareable images
- SYS$LOADABLE_IMAGES—Contains the set of images that are loaded during a bootstrap of the system
- SYS$MAINTENANCE—Contains system diagnostic programs
- SYS$MANAGER—Contains files used in managing the operating system (SYS$MANAGER is the default directory for the system manager's account)
- SYS$MESSAGE—Contains system message files

- SYS$STARTUP—Contains command procedures for starting the system and optional products
- SYS$SYSTEM—Contains the executable images of most operating system components
- SYS$TEST—Contains files used in testing operating system functions
- SYS$UPDATE—Contains files used in applying system updates

1.4 Installation, Upgrades, and Updates

The three methods of installing operating system software on a VAX processor include a new installation, an upgrade, and an update.

If your processor is new, you must install the most recent version of the operating system software. When you perform a new installation, the installation procedure does the following:

- Initializes the target disk, erasing its entire contents
- Creates a system directory structure on the target disk
- Transfers the VMS operating system from the distribution media to the target disk

Caution: **Use the new installation procedure only on newly purchased processors or to destroy the previous contents of the target system disk.**

An **upgrade** is a version of the operating system software in which all new system files replace those of the previous version. If your processor is already running a version of the VMS operating system, you must use either the upgrade procedure or the update procedure, depending on whether the most recent release of the software was an upgrade or an update. Unlike the procedure for a new installation, the upgrade procedure does not initialize the target disk. Instead, it replaces old system files with new. To prevent loss of data resulting from power failures, DIGITAL recommends that you back up your system disk before performing the upgrade.

An **update** replaces only selected files on your system. The update procedure applies patches to some system files and replaces other system files. The *VMS Release Notes Manual* contains a step-by-step description of the update procedure.

Before upgrading or updating your system software, perform these preliminary steps:

1. Restore the operating system to a standard system. The upgrade or update procedures will not work correctly if you have changed the names of system directories in your current operating system or if you have deleted operating system files from them.

2 Back up and restore the system disk following the instructions provided in your VAX processor installation and operations guide. You should back up your current system disk before you install an optional software product, upgrade, or update your system. This provides you with a backup copy of your system in case a problem occurs during the installation. Files are stored contiguously on newly restored disks, leaving a large number of contiguous free blocks in which the new files of the upgrade or update can be stored.

DIGITAL recommends that you use standalone BACKUP, which employs a subset of the Backup Utility qualifiers, to back up and restore your system disk. See your VAX processor installation and operations guide for detailed instructions on using standalone Backup.

3 Set the appropriate switches on the processor control panel, as described in your VAX processor installation and operations guide.

2 Setting Up the System

After you have installed the VMS operating system, you are ready to set it up for site-specific operations. These tasks include

- Creating site-specific command procedures that execute at startup time
- Adding user accounts
- Allocating and controlling system resources

The concepts for performing these tasks are introduced in this chapter.

2.1 Startup and Login Command Files

The command procedure SYS$SYSTEM:STARTUP.COM is invoked when the system is booted. STARTUP.COM configures and initializes the system and executes several site-specific command procedures. Do not modify the STARTUP.COM file. Instead, add or modify commands in the site-specific template files supplied with your VMS distribution kit. There are two versions of each template file in the system manager's directory (logical name SYS$MANAGER): an executable version with the file extension COM, and a non-executable version with the file extension TEMPLATE.

Caution: **Do not modify or delete files with the extension TEMPLATE. The VMSKITBLD.COM procedure uses these files to create a new system disk.**

The following table lists the files in SYS$MANAGER that are supplied as templates and can be modified for site-specific use:

Command Procedure	Function
SYPAGSWPFILES.COM	A file to which you can add commands for installing page and swap files.
SYCONFIG.COM	A file to which you can add commands for site-specific device configuration.
SYLOGICALS.COM	A file to which you can add your site-specific system logical name assignments. (This file contains a command procedure for adding system logical names on a MicroVAX that is not in a cluster. If you do not have a MicroVAX, add your own logical name assignments to the end of the file.)
SYSTARTUP_V5.COM	A file to which you can add or modify commands for performing various operations that execute at startup time.
SYLOGIN.COM	A file to which you can add commands that are executed whenever a user logs in.
LOGIN.COM	An initial login command file that is executed only when the system manager logs into the SYSTEM account.
EDTINI.EDT	A template for the EDT editor initialization file.

See Chapter 2 of the *Guide to Setting Up a VMS System* for guidelines for creating site-specific command procedures.

To modify the site-specific template files and to perform various other system management functions, you must log in to the system manager's account (SYSTEM).

When you boot the system (except on a standalone MicroVAX), it displays a message similar to the following:

```
  VAX/VMS Version n.n <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>
```

Use the following procedure to log in to the system manager's account:

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.

DIGITAL recommends that you change the system manager's account password frequently to maintain system security. Because the account has full privileges by default, 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-1987 15:13:21.07
```

The startup and login procedures provided by DIGITAL should always work. However, if an error is introduced in the startup or login procedures, you may find yourself accidently locked out of the system. See Chapter 3 of the *Guide to Setting Up a VMS System* for emergency startup procedures.

2.2 Accounts and System Resources

The VMS operating system provides several tools for controlling who has access to the system and what resources an individual is authorized to use. These controls are established primarily by using the Authorize Utility to assign specific attributes to each user account when you add the account record to the user authorization file (UAF). See Chapter 4 of the *Guide to Setting Up a VMS System* for a detailed description of a UAF record and procedures for setting up user accounts.

2.2.1 The User Authorization File (UAF)

User account records are maintained in the system UAF file named SYS$SYSTEM:SYSUAF.DAT. Each record consists of fields providing information about the accounts identification, login characteristics, login restrictions, and resource control attributes. The username field is specified as a parameter to Authorize Utility commands; the other fields are specified as qualifier values of Authorize Utility commands.

The VMS system uses the UAF to validate login requests and to set up processes for users who successfully log in. You create, examine, and modify UAF records with the Authorize Utility (AUTHORIZE).

The following resource control attributes are assigned in the UAF record:

- Priority
- Limits and quotas
- Privileges

These resource control attributes are discussed briefly in the following sections.

2.2.2 Priority

A user's priority is the base software priority used in scheduling computer time for the process associated with the user's account. Priorities range in value from a low of 0 to a high of 31; 1 through 15 are time-sharing priorities and 16 through 31 are real-time priorities.

Processes with real-time priorities are scheduled strictly according to base priority—the executable real-time process with the highest base priority executes first. Processes with time-sharing priorities are scheduled according to a slightly different principle, to promote equitable service to all users.

You should leave the base priority at the default of 4 for time-sharing accounts.

Batch queues are usually given lower priority than time-sharing accounts to provide better response time to interactive users.

2.2.3 Limits and Quotas

Limits are set on system resources that can be reused, for example, the amount of memory that a process can use for I/O requests. Most limits restrict the use of physical memory. You set limits for processes associated with accounts through the appropriate UAF fields. Some of these limits can be changed later with DCL commands or by calling system services from programs.

A process passes on its resources to a subprocess (for example, when you create a subprocess with the SPAWN command) in one of the following three ways, depending on the particular resource:

- Pooled—A process and its subprocesses share the resource on a first-come, first-served basis until the limit is reached.

- Nondeductible—A subprocess receives the same limit on the resource as the creator receives. The creator's limit is not affected.
- Deductible—A subprocess receives a portion of the creator's resource. That portion is deducted from the creator's limit.

Normally, you should leave limits at their default values. See the sample SYSTEM and DEFAULT user authorization file records supplied with the Authorize Utility on your VMS distribution kit, for the default values for the system and user accounts. Also see Chapter 5 of the *Guide to Setting Up a VMS System* for a full description of limits and quotas.

2.2.4 Privileges

Privileges determine what functions users are authorized to perform on the system. System manager functions require privileges that are denied to most users. The system manager's account is granted full privileges by default. Therefore, you must exercise caution in using the system manager's account, because the system allows you to modify and delete any file regardless of its protection. See Chapter 5 of the *Guide to Setting Up a VMS System* for a description of privileges.

2.2.5 Accounting

In addition to the Authorize Utility, the accounting facility can be used to control the use of system resources. The accounting facility collects statistics on the use of system resources in an accounting log file. You can use this information to monitor system activity and charge for the use of system resources.

On most VAX processors, the accounting log file automatically begins logging the activities of users when your system is started. This function is enabled by the SET ACCOUNTING command in the SYS$MANAGER:SYSTARTUP_V5.COM template. You can disable the accounting facility by adding the /DISABLE qualifier to the SET ACCOUNTING command. If your processor is a standalone MicroVAX, however, the accounting facility is turned off by default to conserve disk space.

The log file, SYS$MANAGER:ACCOUNTNG.DAT, records the use of system resources and is the source of the accounting reports generated by the ACCOUNTING command. You can close the current accounting log file and open a new version with the SET ACCOUNTING/NEW_FILE /LOG command (see your on-line SYS$MANAGER:SYSTARTUP_V5.COM template).

See Chapter 7 of the *Guide to Maintaining a VMS System* for more information on the accounting log file.

3 Security Management

In the VMS operating system, the management of system security is concerned with the following two basic groups:

- Individual users or processes who gain access (or attempt to gain access) to the system (for example, interactive users, batch processing jobs, network access jobs)
- Information and resources that are kept on the system (for example, data files, application programs, system utilities)

Users and Process Gaining Access to the System

For the VMS operating system, a *process* is the means by which the system is accessed. When a user logs into the system interactively or when a batch or network job starts, the system creates a process for that user or job.

Each process is associated with the user (or job) for whom the process was created. For each process, the system also has information that determines access control information for the process. The user authorization file (UAF) and the rights database file (RIGHTSLIST) are the primary databases for storing this information; the information in these files specifies the access and system resource usage that are allowed for the processes created for individual users. As previously discussed in Section 2.2, information such as the UIC, privileges, password, and permitted access times are supplied in the UAF record for each user.

Information and Resources Kept on the System

Any VMS system has information (for example, databases, logical name tables, executable images) and system resources (for example, disk space and CPU availability). The second aspect of system security is the control of such information and resources.

A *system object* is a file, directory, logical name table, global section, queue or any other source on the system that a process might access. A level of security is associated with every system object, even if that level of security allows unrestricted access to any process. The UAF and RIGHTSLIST databases contain information that define the access that each process will have to various objects. Additional authorization information is stored with the object to be protected.

The VMS operating system provides two security mechanisms for protecting objects: UIC-based protection and ACL-based protection. Almost all objects have UIC-based protection. In addition, objects can be assigned a level of protection known as ACL-based protection. An access control list (ACL) refines the protection scheme for sharing system objects by using both UIC-based access codes and ACL-based *identifiers*, which further define the rights associated with a process.

The following sections describe these two security mechanisms in more detail and the components they share in common. See *Guide to VMS System Security* for more information on security management.

3.1 UIC-Based Protection

UIC-based protection consists of the following components:

- User identification code (UIC)
- Ownership category
- Access category
- Protection mask

It is important to understand the basic components of protection before distinguishing the types, as described in the following sections.

3.1.1 User Identification Code

Each user process in the system is assigned a user identification code (UIC) in the user authorization file (UAF) with the Authorize Utility. Each object on the system is also associated with a UIC (typically the UIC of its creator). A UIC consists of two parts, group and member, specified in the following format:

```
[group,member]
```

A UIC can be either numeric or alphanumeric. A Numeric UIC consists of a group number in the range 0 through 37776 (octal) and a member number in the range 0 through 177776 (octal).

An alphanumeric UIC consists of a member name (the user name parameter specified with the Authorize Utility command ADD) and, optionally, a group name (the name specified with the Authorize Utility command ADD/ACCOUNT); both member and group names must contain at least one alphabetic character and up to a maximum of 31 alphanumeric characters (including A–Z, 0–9, underscore, and dollar sign characters).

You can generally specify a numeric UIC and its equivalent alphanumeric UIC interchangeably. The member component of an alphanumeric UIC must be unique to the system. The following examples show several UICs in proper syntax.

UIC	Translation
[200,10]	Group 200, member 10
[3777,3777]	Group 3777, member 3777
[USER,FRED]	Group USER, member FRED
[EXEC,JONES]	Group EXEC, member JONES

When you log in to a VMS system, your process UIC is derived from information in your UAF account. Typically, your process UIC does not change, although it can be changed with the SET UIC command (which requires CMKRNL privilege). By default, detached processes (created by the DCL commands SUBMIT or RUN) and subprocesses (created by the DCL command SPAWN) take the same UICs as their creators. By default, an object (such as a file or logical name table) receives the UIC of the process creating it.

3.1.2 Ownership and Access Categories

The relationships between the UIC of a process and the UIC of an object fall into the following four ownership categories:

- System—The UIC of the process is in the range 1 through 10 (octal) or the process has SYSPRV (or GRPPRV) privilege. (The range of system UICs is determined by the SYSGEN parameter MAXSYSGROUP, which defaults to 10 octal.)
- Owner—The UIC of the process and the UIC of the object are identical.
- Group—The group number of the process and the group number of the object are identical.
- World—The UIC of the process and the UIC of the object can have anything or nothing in common.

A process and object may have any number of the above relationships. For example, a process that owns an object necessarily has group and world relationships with the object.

The VMS operating system provides the following four access categories, which determine the ways in which a process can access an object:

- Read (allocate)—Read a file; read from a disk volume; allocate nonfile devices.
- Write—Write a file; write to a disk volume.
- Execute (create)—Execute an image file; look up entries in a directory if you explicitly specify the file name (without using wildcard characters); create files on a disk volume.
- Delete—Delete files.

3.1.3 Protection Masks

The ownership and access categories are combined into a protection mask, which conveys the relationship that a subject has to an object. A protection mask associated with an object determines the type of access allowed to a user, based on the relationship between the user UIC and the object UIC.

A protection mask consists of four ownership fields, and four access fields. Each access field applies to one category of ownership, as shown in the following table:

Ownership	Access			
SYSTEM	READ	WRITE	EXECUTE	DELETE
OWNER	READ	WRITE	EXECUTE	DELETE
GROUP	READ	WRITE	EXECUTE	DELETE
WORLD	READ	WRITE	EXECUTE	DELETE

Use the following syntax to specify a protection mask:

```
(ownership[:access],...)
```

Specify ownership as one of the following (each may be abbreviated to one character): SYSTEM, OWNER, GROUP, or WORLD. Specify access as one or more of the following: R (read), W (write), E (execute), D (delete). Omission of the colon and access indicators means no access for that category of ownership.

The protection mask in the following example allows system users full access to an object, the owner full access except delete, and group and world users no access:

```
SET PROTECTION=(S:RWED,O:RWE,G,W) [JONES]MY_FILE.TXT
```

The operating system provides each process with a default UIC-based protection of (S:RWED,O:RWED,G:RE,W). To change the default protection, enter the SET PROTECTION command with the /DEFAULT qualifier as shown in the following example:

```
$ SET PROTECTION = (S:RWED,O:RWED,G:RE,W:RE)/DEFAULT
```

3.2 ACL-Based Protection

For most interactive user accounts, the default UIC-based protection mask is adequate. However, in some cases (such as project accounts) you may want to set up an additional level of protection by using access control lists. ACL-based protection provides a more refined level of security in cases where different groups or members of overlapping groups share access to an account such as a project account. ACLs should be used only as needed because they consume additional amounts of paged system dynamic memory when files are open. They also require additional processing time.

ACLs consist of *identifiers* and *ACES* or access control entries. These specify the type access and the users or user groups to be granted access to system objects. Users are associated with identifiers in the *rights database,* accessed through the Authorize Utility. The rights database is a file that contains all the identifiers defined for the system. You can establish ACLs for the following system objects: files, directory files, global sections, devices, logical name tables, and queues.

3.2.1 Identifiers

An identifier is a value that represents an individual user, a group of users, or an aspect of the user's environment. There are three types of identifiers:

- UIC
- General identifiers
- System-defined identifiers

UIC identifiers can be in both numeric and alphanumeric form and are useful in identifying individual users. UIC identifiers must be enclosed in brackets and may have wildcard characters in either the group or member fields (for example, [EXEC,*]).

General identifiers are those you explicitly associate with users in the rights database. These general identifiers are useful in identifying multiple groups of users outside the bounds of UIC groups. For example, you could create the identifier SECRET and assign it in the rights database to a selected group of users, some of whom could be in different UIC groups.

System-defined identifiers identify users by their mode of using the system. The system-defined identifiers are described in the following table:

System-defined Identifiers	Type of user
BATCH	Batch user
DIALUP	User logged in on a dial-up terminal
INTERACTIVE	Interactive user
LOCAL	User logged in on local terminal
NETWORK	Network process
REMOTE	User logged in over the network

Generally, you should treat the preceding system-defined identifiers as being mutually exclusive. However, you can combine them with UIC identifiers or general identifiers by connecting them with plus signs (for example, [FRED]+BATCH). Access is granted only if both identifiers are true. (In the example [FRED]+BATCH, the user identified as [FRED] must be running a batch job for the system to grant the specified access.)

The following are examples of user-defined identifiers that are valid for ACL-based protection:

- PAYROLL—Specifies all users holding the identifier PAYROLL.
- [USER,JONES]—Specifies the user whose alphanumeric UIC is group USER and member JONES.
- [200,10]—Specifies the user whose numeric UIC is group 200, member 10.
- [FRED]+BATCH—Specifies the batch user whose alphanumeric UIC is FRED.
- DIALUP—Specifies all users logged in on a dial-up terminal.

3.2.2 Access Control Entries (ACE)

An entry in an access control list (ACL) specifies the access to a system object that is to be granted or denied to a user. This access is specified by the identifier. Different kinds of access include: NONE, READ, WRITE, EXECUTE, DELETE and CONTROL. Access control lists for each object can hold numerous entries, limited only by overall space and performance considerations.

The three types of ACEs are as follows:

- Identifier ACE - controls the types of access allowed to specific users based on the user's identification.
- Default protection ACE - allows you to specify a UIC-based protection code to be propogated throughout the directory tree.
- Security alarm ACE - allows you to request that a security alarm message be sent to the operator's terminal if an object is accessed in a particular way.

The following example shows an ACL containing four ACES:

```
(IDENTIFIER=[200,201],ACCESS=READ+WRITE+EXECUTE)
(IDENTIFIER=[FRED]+BATCH,ACCESS=WRITE+EXECUTE)
(IDENTIFIER=PAYROLL,ACCESS=READ)
(IDENTIFIER=DIALUP,ACCESS=NONE)
```

The identifiers in the previous example are described as follows:

1 The first ACE grants the user identified by UIC [200,201] read, write, and execute access to the system object.

2 The second ACE grants batch users with the alphanumeric UIC [FRED] write and execute access to the system object.

3 The third ACE grants users who hold the identifier PAYROLL read access to the system object.

4 The fourth ACE denies holders of the system-defined identifier DIALUP any access to the system object.

You can override default UIC protection for specified directories or subdirectories by placing a default_protection ACE in the ACL of the appropriate directory file. The default protection specified in the ACE is applied to any new file created in the specified directory or any subdirectory of the directory. The following ACE, which must be in the ACL of a directory file, specifies that the default protection for that directory and the directory's subdirectories allow system and owner processes full access, group processes read and execute access, and world users no access.

```
 (DEFAULT_PROTECTION,S:RWED,O:RWED,G:RE,W:)
```

To specify a default identifier ACE to be copied to the ACL of any file subsequently created in the directory, specify the DEFAULT option in the directory file's identifier ACL.

3.2.3 Rights List

The system determines protection by checking the object's ACL against the list of identifiers held by the user to find a matching entry. This list, called a *rights list,* is maintained in the *rights database.*

The rights list is the portion of the rights database associated with a specific user. The rights list is created for each user at login and contains all the identifiers and attributes held by the user. The rights database is a file that contains all the identifiers defined for the system.

3.3 How the System Grants Access

The system determines whether to grant a user access to an object in the following steps:

1. ACL—If the user matches an identifier in the object's ACL, the system grants or denies access based on the ACL. However, even if an entry in the ACL denies access, the system may still grant access based on the SYSTEM and OWNER fields of the UIC-based protection.
2. UIC—If the user does not match an identifier in the object's ACL or the object has no ACL, the system grants or denies access based on the relationship between the user's UIC and the object's UIC as qualified by the object's protection mask.
3. Privileges—If the system denies the user access, the user may be granted access by using one of the following privileges: BYPASS, GRPPRV, READALL, or SYSPRV.

In summary, UIC-based protection is useful for denying or granting access to a specified group of users or to all users on the system. The optional ACL-based protection allows further control over the protection of an object. You can grant or deny access to individual users, and further identify users by certain aspects of their usage (such as whether they are interactive, batch, local, remote, or dial-up users). The combination of UIC and ACL protection provides a way to specify multiple subsets and overlapping groups of users. See the *Guide to VMS System Security* for detailed information on security management.

4 Maintaining the System

At many sites, the system manager keeps the system running smoothly by performing the following tasks on a periodic basis, or as user needs dictate:

- Maintaining public files and volumes
- Performing backup operations
- Managing disk space
- Managing batch and print operations
- Handling error conditions

These tasks are briefly described in the sections that follow.

4.1 Maintaining Public Files and Volumes

Public volumes, also called system volumes, are file-structured disk volumes that contain public files. Such files are made available to most, if not all, users. Public volumes can also contain files that users create for their own private use.

As a system manager, you must balance users' needs and the system's available mass storage resources by determining the following:

- How mass storage devices on your system will be configured
- Which devices will hold public system volumes
- Which devices will be available for users' private volumes
- How the public volumes will be configured

See Chapter 2 of the *Guide to Maintaining a VMS System* for guidelines on planning public volumes.

You permit users to create and store files on a public volume by creating a default directory on a public volume for each authorized user. You control access to public files and volumes by the protection codes that you establish. A user is free to create, write, and manipulate files on a public volume only if the following conditions are met:

- Volume and file protection allow access
- User has a valid account on the system
- User has write access to a directory on the volume
- Disk quota permits usage

4.1.1 Preparing and Manipulating Volumes

Users may prepare and manipulate their own volumes or they may rely on the system manager or operator to assist them, depending on the size of the installation and the type of volume to be accessed. Users may require assistance in the following instances:

- The processor and its peripheral devices are off limits to or remotely located from some or all users.
- The magnetic tape system has requested that a tape volume be mounted.
- A system or public disk needs to be mounted by the operator or system manager, because most users do not have sufficient privileges.

On most processors, users requiring assistance can use the operator communication manager (OPCOM) to communicate with an operator. OPCOM is a system process that receives input from a process that wants to inform an operator of a particular status or condition; OPCOM passes the message to the operator, and tracks the message.

The operator can perform one or more of the following tasks:

- Physically mounts the volume on the device. Physically mounting a volume means placing it on a specific drive and starting the drive.
- Initializes new volumes, using the INITIALIZE command. Initializing a volume deletes all information from the volume and imparts a Files–11 standard file structure recognized by VMS.
- Mounts the volume, using the MOUNT command. Mounting a volume establishes a logical relationship between the device on which the volume is physically mounted, and one or more processes that can gain access to the volume.

For smaller processors with removable media, the user may perform the following tasks unassisted:

- Load—Insert the medium into the drive compartment.
- Initialize—Use the INITIALIZE command to remove all previous contents from the medium.
- Mount—Use the MOUNT command to logically mount the medium and allocate the device.
- Perform operations—Use the medium to access files and perform the desired operations on them.
- Dismount—Use the DISMOUNT command to logically dismount the medium and deallocate the device.
- Unload—Remove the medium from the drive compartment.

See your VAX processor installation and operations guide for specific procedures for manipulating removable media on your system.

4.1.2 Responding to Operator-Assisted Mount Requests

To use OPCOM to communicate with system users, you must define a terminal as an operator terminal. Normally, the console terminal is designated as the operator terminal. If the system crashes, you return to the console prompt (> > >). (On a workstation, the console prompt appears in the OPA0 window; you can toggle back and forth between the regular workstation windows and OPA0 by pressing the F2 key.)

You define an operator terminal by entering the DCL command REPLY/ENABLE. When you enter this command at the specified terminal, the following OPCOM message is displayed at the terminal.

```
%%%%%%%%%%%  OPCOM, dd-mmm-yyyy hh:mm:ss,cc, %%%%%%%%%%%
Operator _nodename$terminal-name: has been enabled, username USERNAME
```

This message indicates which terminal has been established as an operator terminal (OPA0) and when it was established.

To communicate with the operator, a user enters the REQUEST command with the /TO qualifier and a message string. If a user enters the /REPLY qualifier, the operator is alerted that the user is waiting for a response.

If the user enters a REQUEST/REPLY command, the request is displayed at the operator terminal in the following format:

```
%%%%%%%%%%  OPCOM,dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%%%
request request-id from user USERNAME __terminal-name:,
"message-text"
```

This message indicates 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 text.

If the user enters a REQUEST command without the /REPLY qualifier, the request is displayed at the operator terminal in a format similar to the preceding one, but without a request identification number. The message has the following format:

```
%%%%%%%%%%%  OPCOM,dd-mmm-yyyy hh:mm:ss.cc  %%%%%%%%%%%
request from user USERNAME __terminal-name:, "message-text"
```

When a user enters a REQUEST/REPLY command and the operator has disabled all operator terminals with the appropriate REPLY/DISABLE commands, OPCOM returns a message to the user indicating that no operator coverage is available, but OPCOM continues to record all subsequent user requests in the operator log file. (See Chapter 7 of the *Guide to Maintaining a VMS System* for more information on the operator log file.)

You respond to user requests by entering REPLY commands and qualifiers, for example:

```
$ REPLY/TO=24 "SUBSTITUTE  DMA1"
```

See the *VMS DCL Dictionary* for a complete list of REPLY qualifiers and their functions.

As an operator handling a user request for mounting a volume, you do the following:

- Place the volume on the specified device.
- Perform the necessary start-up procedure for the device. On disk drives, the start-up procedure requires pressing the START or RUN button; on magnetic tape drives, it requires pressing the LOAD button.

If the user intends to write on a magnetic tape volume to be mounted, make sure the magnetic tape contains a write ring before you load the volume on the drive. A volume that does not contain a write ring is write-locked and thus cannot be accessed for write operations.

If the user wants a particular disk checked for bad blocks, make sure that the volume has been mounted with the /FOREIGN qualifier.

4.2 Performing Backup Operations

Regularly backing up your files and volumes is an effective way to maintain data integrity. Protection is not the only reason for backing up files, however. If you have limited disk space, it is often practical to store seldom used disk files on magnetic tape until you need to retrieve them.

An efficient backup operation depends on the nature of your work, and the sensitivity of the data in your files. Many times, it is more efficient to perform a *selective* backup operation. In a selective operation, BACKUP selects files that meet specific criteria, such as version number, file type, UIC, date and time of creation, expiration date, or modification date. When only a small percentage of the files on a volume have been modified, you can save time and storage space by performing selective backups.

An *incremental* backup operation selects only those files that have been modified since the last backup. In a typical operating environment, incremental backups might be done daily, or every other day, with a full backup of the entire system volume done at the end of each week.

BACKUP has the following five basic types of operations:

- Copy
- Save
- Restore
- Compare
- List

The type of media you have affects the type of backup operation you select. For example, a Copy operation copies a Files–11 structured disk in standard VMS format; a Save operation creates a backup *save set* on magnetic tape, a standard Files–11 structured disk, or a sequential disk. A save set is a file with a unique BACKUP format. A save set must be restored to a Files–11 structured disk in standard VMS format.

See the *VMS Backup Utility Manual* and Chapter 4 of the *Guide to Maintaining a VMS System* for more information on performing backups.

4.3 Managing Disk Space

Part of your job as system manager is to make efficient use of available disk space. This section briefly describes some methods of conserving and controlling disk space:

- Purge old versions of files

  You should periodically purge outdated versions of system log files, such as the operator's log and the accounting log files. You can also use wildcard characters to perform global purges.

- Use the /VERSION_LIMIT=n qualifier

  You can restrict the number of file versions that can be generated in a user's directory by entering the /VERSION_LIMIT qualifier with the SET DIRECTORY or CREATE DIRECTORY command. When you are creating a default directory for a user, use the /VERSION_LIMIT qualifier with the CREATE/DIRECTORY command as in the following example:

  ```
  $ CREATE/DIRECTORY $DISK1:[JONES]/OWNER_UIC=[200,1]/VERSION_LIMIT=3
  ```

- Set file expiration dates

  File expiration is a file system feature (available on Files–11 Structure Level 2 disks only) that uses the expiration date of each file to track the file's use. The expiration dates aid in the disposal of seldom used files. To enable the setting of expiration dates, use the DCL command SET VOLUME:

  ```
  $ SET VOLUME device-name: /RETENTION=(min,max)
  ```

See Chapter 5 of the *Guide to Maintaining a VMS System* for more information on managing disk space.

4.4 Managing Batch and Print Operations

Setting up queues and managing batch and print jobs is of primary importance to maintaining efficient daily operations on your system. As system manager, you must monitor the types of jobs that run on your system and develop ways to submit, schedule, and execute jobs to reap maximum performance benefits.

After the system is booted, the DCL command START/QUEUE/MANAGER must be executed before any other queue commands can be entered. You must initialize and start queues before users can enter SUBMIT and PRINT commands to send jobs to the queues for processing. The queue manager and queues are usually initialized and started as part of the system startup procedure. See the DIGITAL-supplied template file, SYS$MANAGER:SYSTARTUP_V5.COM, for sample commands for initializing and starting queues.

Many batch/print operations require that you interact with the running system by entering commands at the DCL command level. For example, you may need to delete a job entry, or pause a queue temporarily to modify queue attributes or repair a defective printer. Most of the commands for managing queues and jobs require privileges. For example, you must have OPER privilege or EXECUTE access to a queue to modify its attributes.

See Chapter 6 of the *Guide to Maintaining a VMS System* for detailed information on the procedures for performing batch and print operations.

4.5 Handling Error Conditions

The VMS operating system provides several utilities for recording and reporting errors and other system events. The following DCL commands invoke utilities that analyze system errors conditions:

- ANALYZE/CRASH_DUMP—invokes the System Dump Analyzer (SDA) for analysis of a system dump file.
- ANALYZE/DISK_STRUCTURE—invokes the Analyze/Disk_Structure Utility. This utility verifies the Files–11 structure on disk volumes, reports errors, and repairs errors (when you specify the /REPAIR qualifier).
- ANALYZE/ERROR_LOG—invokes the Error Log Report Formatter (ERF), to report selected contents of an error log file. The error reports generated by the Error Log Utility are useful in two basic ways:
  - They aid preventive maintenance by identifying areas within the system that show potential for failure.
  - They aid the diagnosis of a failure by documenting the errors and events that caused it.
- ANALYZE/MEDIA - invokes the Bad Block Locator Utility (BAD), which analyzes block-addressable devices and records the location of blocks that cannot reliably store data.
- ANALYZE/SYSTEM - invokes the System Dump Analyzer Utility (SDA) for analysis of a running VMS system.

See Chapter 7 of the *Guide to Maintaining a VMS System* for more information on system log files.

5 Performance Management

This chapter introduces the basic concepts of performance management. It is not meant to be used as a tutorial for tuning your system. Refer to the *Guide to VMS Performance Management* for detailed information on performance tuning.

Performance management of a VMS system means optimizing your hardware and software resources for the current workload. This task entails several distinct but related activities:

- Acquiring a thorough familiarity with your workload and an understanding of how that workload exercises 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 behavior 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.

5.1 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 behavior 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 behavior; 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 dedicate time to observing system operation using the following tools:

- Monitor Utility
- Accounting Utility
- SHOW commands (available through DCL)

The *Guide to VMS Performance Management* provides detailed procedures for using the Monitor Utility and, to a lesser extent, other VMS tools to observe and evaluate system performance. 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. You will begin to see how certain activities affect system performance and how the number of users or the time of day affects some of the values. 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 use these same tools, together with your knowledge, 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.

You should also understand that system resources are used when you use the tools that are available to the system manager. 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.

5.1.1 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 above. 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.

5.1.2 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 knowing which images are heavy consumers of 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.

5.1.3 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? (See the *Guide to VMS File Applications.*)
- Are you making the best use of the code-sharing ability that the VMS system offers? 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.

5.1.4 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. See Chapter 6 in the *Guide to Maintaining a VMS System* for guidelines on batch and print operations.
- 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. See the *VMS System Generation Utility Manual* for detailed information on system parameters.

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 bottlenecks 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). (See the *Guide to VMS File Applications.*)

5.2 Understanding System Tuning

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. You goal is to establish values that, on 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. See Chapter 6 of the *Guide to Setting Up a VMS System* for more information on AUTOGEN functions. 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.

The *Guide to VMS Performance Management* describes how to select the parameters and new values that are likely to produce the desired changes.

5.3 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.

Experience has shown that the most common cause of disappointment in system performance is 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.

5.4 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. See the *VMS DCL Dictionary* for detailed information on the SHOW command, and the *VMS Monitor Utility Manual* for more information on the Monitor Utility.

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.

5.5 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 the 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. (See the *VMS System Generation Utility Manual* and Chapter 6 of the *Guide to Setting Up a VMS System*.)
- 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 *VMS Install Utility 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.

Before you undertake any tuning operation, however, be sure you are familiar with the VMS resource management mechanisms described in the *Guide to VMS Performance Management.*

6 VAXcluster Overview

A cluster is a highly integrated organization of VAX or MicroVAX[1] systems (or a combination of these systems) that communicate over a high-speed communications path. A cluster environment provides both a single VMS security and management domain, and the ability to share processing resources, queues, and disk storage. It is this ability to share resources under a single security and management perimeter that distinguishes clusters from tightly coupled multiprocessor systems and loosely coupled networks.

This chapter describes the key components and distinctive features of the VAXcluster environment. Topics include the following:

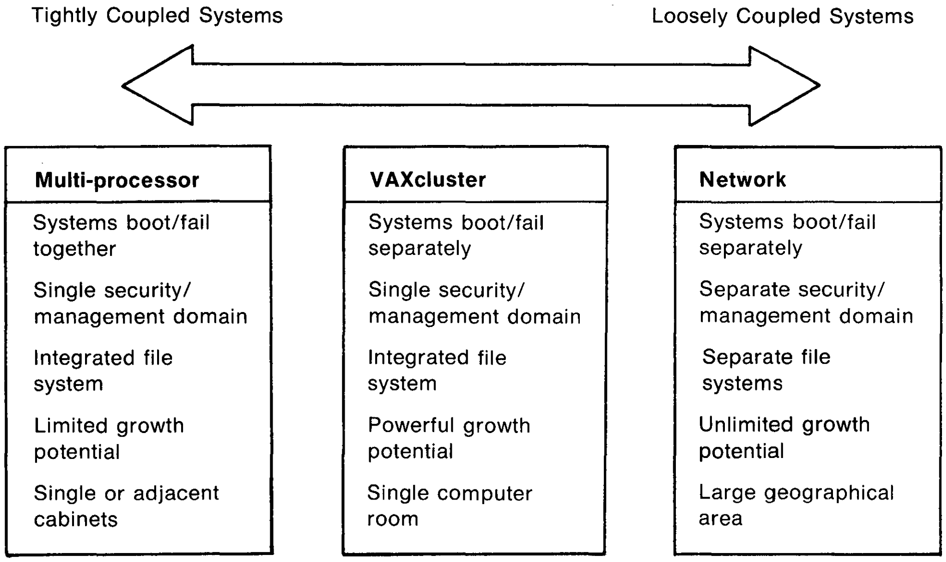
- Clusters and other multiprocessor environments
- Cluster software and hardware components
- Cluster configuration types
- DECnet–VAX connections
- Cluster connection management
- Shared cluster resources

6.1 Clusters and Other Multiprocessor Environments

Clusters are best understood when compared to tightly coupled multiprocessor systems and loosely coupled networks. Figure 6–1 shows how clusters fit into the range of multiprocessor environments.

[1] MicroVAX II or MicroVAX 3000 series; MicroVAX I systems are not supported in VAXcluster configurations.

Figure 6–1 Clusters and Other Multiprocessor Configurations



Tightly Coupled Systems ⟷ Loosely Coupled Systems

Multi-processor	VAXcluster	Network
Systems boot/fail together	Systems boot/fail separately	Systems boot/fail separately
Single security/ management domain	Single security/ management domain	Separate security/ management domain
Integrated file system	Integrated file system	Separate file systems
Limited growth potential	Powerful growth potential	Unlimited growth potential
Single or adjacent cabinets	Single computer room	Large geographical area

ZK-1344-83

6.2 Cluster Software

The software components used to implement cluster functions are as follows:

- System Communication Services
- Connection Manager
- Distributed File System
- Distributed Lock Manager
- Distributed Job Controller
- Mass Storage Control Protocol (MSCP) Server

The *System Communication Services* (SCS) software implements internode communication, according to DIGITAL's System Communication Architecture (SCA).

The *Connection Manager* is the software layer that dynamically defines and coordinates the cluster. The connection manager uses the system communication services and provides an acknowledged message delivery service for higher VMS software layers. The connection manager also conducts cluster state transitions—that is, nodes joining or leaving the cluster.

The *Distributed File System* allows all processors to share disk mass storage, whether the disk is connected to an HSC or to a processor. A local disk may be made available to the entire cluster, and all cluster available disks appear as if they are local to every processor.

The distributed file system and VAX Record Management Services (VAX RMS) provide the same access to disks and files clusterwide that is provided on a standalone system. VAX RMS files may be shared clusterwide to the record level.

The *Distributed Lock Manager* is used for synchronization functions by the file system, job controller, device allocation, and other cluster facilities. It is available to users to develop cluster applications. The distributed lock manager implements the $ENQ and $DEQ system services to provide clusterwide synchronization of access to resources.

The lock manager provides a mechanism to lock and unlock resource names. It also provides a queueing mechanism so that processes can be put into a wait state until a particular resource is available. As a result, cooperating processes can synchronize their access to shared objects such as files or records.

If a processor in the cluster should fail, all locks held by the failed processor are released. This mechanism allows processing to continue on the remaining processors.

The distributed lock manager also supports clusterwide deadlock detection.

The *Distributed Job Controller* makes queues available clusterwide. A cluster operates with a common set of batch and print queues. Users can submit jobs to any queue within the cluster, provided that the necessary mass storage volumes and peripheral devices are accessible to the system on which the job executes. Cluster managers can also set up generic batch queues that distribute batch processing workloads among nodes.

The *Mass Storage Control Protocol (MSCP) Server* implements the MSCP protocol, which is used to communicate with a controller for Digital Standard Architecture (DSA) disks, such as HSC disks. The MSCP Server implements this protocol on a processor, submits the I/O requests to local UNIBUS, MASSBUS, and Unibus Disk Adapter (UDA) disks, and accepts the I/O requests from any node in the cluster. In this way, the MSCP Server makes locally connected disks available to all nodes in the cluster.

All of these software components exist on each processor in the cluster. Therefore, if one processor fails, the cluster continues to function, because the remaining processors possess the necessary software components.

6.3 Cluster Hardware

Basic VAXcluster hardware components are listed in Table 6–1. Unlike the other components, the HSC is not required in all configurations; however, the HSC provides additional capability in some configurations.

Table 6–1 VAXcluster Hardware Components

Component	Function
VAX processor	A VAX or MicroVAX processor running the VMS operating system. Any VAX processor in the cluster is considered an *active node*.
CI	The computer interconnect (CI) is a high-speed, dual-path bus that connects VAX processor nodes and intelligent I/O subsystems (HSCs) in a computer room environment. Only one CI may be used in each cluster.
Star coupler	The star coupler is the common connection point for all nodes connected to the CI. As with the CI bus, the star coupler is dual pathed and contains separate components for each path. The star coupler connects all CI cables from the individual nodes, creating a radial or "star" arrangement that has a maximum radius of 45 meters. It supports the physical connection or disconnection of nodes during normal cluster operations, without affecting the rest of the cluster.
Controller	A microcoded, intelligent controller that connects VAX processors to the CI. Each interface connects to the CI bus, which consists of two transmitter and two receiver cables. Under normal operating conditions, both sets of cables are available to meet traffic demands. If one path becomes inoperative, then all traffic uses the remaining path. The VMS operating system periodically tests a failed path. As soon as a failed path becomes available, it will automatically be used for normal traffic.
HSC	The Hierarchical Storage Controller (HSC) is a self-contained, intelligent, mass storage subsystem that enables cluster nodes to share DIGITAL Standard Architecture (DSA) disks. Because the HSC is an intelligent controller, it optimizes physical disk operations. The HSC is considered a *passive node*.
Ethernet	The Ethernet is a bus, in the shape of a branching tree, that uses digital baseband signalling. The Ethernet is used both for DECnet–VAX transmissions, and, in some cluster configurations, for internode SCS commumications. The maximum data rate is 10 million bits per second, but in practice, transmission between a pair of nodes on an Ethernet occurs at a considerably lower rate. Each Ethernet can support up to 1023 nodes; the maximum distance between nodes is 2.8 kilometers (1.7 miles). Ethernet circuit devices supported for VAX processors are the DEUNA, DELUA, and DEBNA. MicroVAX II class systems use the DEQNA or the DESVA.

6.4 Cluster Configuration Types

Depending on your processing needs and available hardware resources, the number of configurations you can create is virtually countless. In VMS Version 5.0, however, you start with one of the following base configuration types:

- CI-only VAXcluster configuration
- Local Area VAXcluster configuration
- Mixed-interconnect VAXcluster configuration

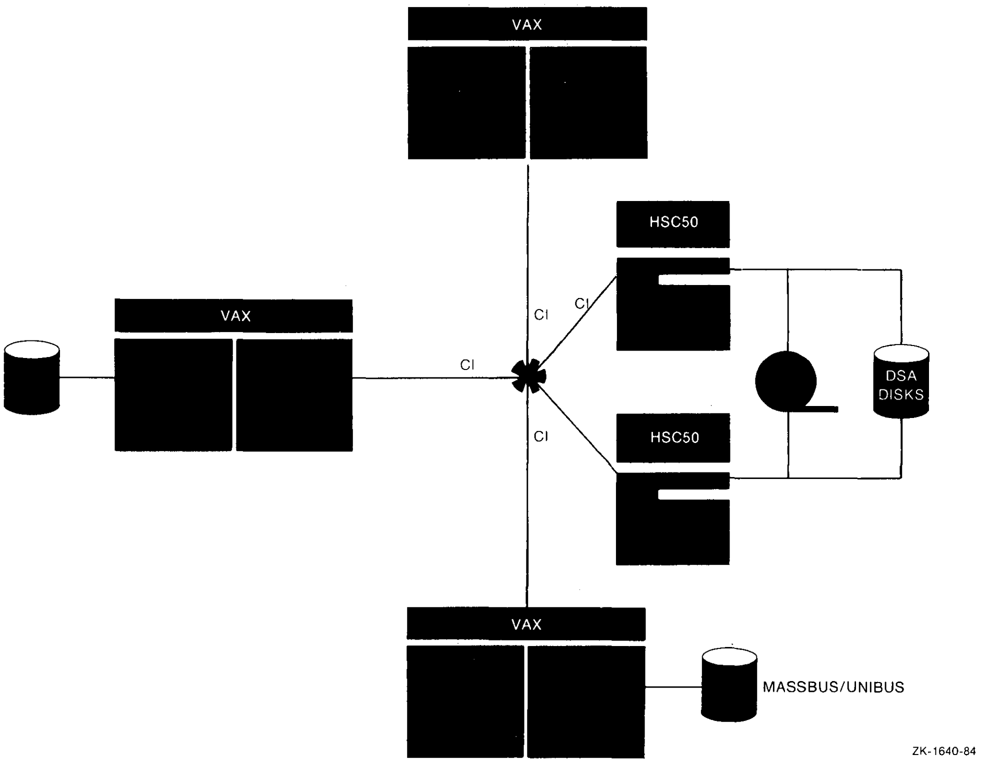
These configuration types are distinguished by the interconnect devices (CI, Ethernet, or both) and supporting software used for SCS interprocessor communication. Sections 6.4.1 through 6.4.3 describe each type of configuration and list configuration-specific rules. For more information on configuation rules, refer to the VAXcluster Software Product Description (SPD) document.

Once you have determined which type of configuration best meets your needs, you can set up your cluster using the procedures described in the *VMS VAXcluster Manual.*

6.4.1 CI-Only VAXcluster Configurations

As its name indicates, a CI-only VAXcluster configuration uses exclusively the CI for interprocessor communication, with the star coupler as the common connection point for all cluster nodes (VAX processors and Hierarchical Storage Controllers). Cluster nodes may be VAX–11/750 or any larger VAX processors, or they may be HSCs. Figure 6–2 shows how the components are typically configured.

Figure 6–2 Typical CI-Only VAXcluster Configuration



6.4.2 Local Area VAXcluster Configurations

In a local area cluster, interprocessor communication is carried out over the Ethernet by software that emulates certain CI port functions. A cluster node may be a VAX or MicroVAX processor; HSCs are not used.

A single Ethernet may support multiple local area clusters, each identified and secured by a unique *group number* and a *cluster password.* (For information on cluster security, see Section 6.4.4.)

A local area cluster configuration consists of one or two *boot servers (boot nodes)* and up to 26 *satellite nodes.*

A *boot server* is both a management center for the cluster and a major resource provider. Its system disk contains the cluster common files for startup, authorization, and queue setup, as well as the directory roots from which the satellite nodes are booted. (The cluster manager creates these directory roots—one for each satellite—using the CLUSTER_CONFIG.COM command procedure, described in the *VMS VAXcluster Manual.*)

A boot server makes available to the cluster such resources as user and application data disks, printers, and distributed batch processing facilities.

Using DECnet Maintenance Operation Protocol (MOP), a boot server responds to downline load requests from satellites. When a satellite requests an operating system load, the boot server responds to the request and sends an image to the satellite that allows the satellite to load the VMS operating system and join the cluster.

Note: **Because a boot server *must* serve its system disk to the cluster (and usually its data disks as well), a boot server is, by definition, always a *disk server*.**

Boot servers should be the most powerful machines in the cluster. They should also use the highest bandwidth Ethernet adapters available.

A boot server may be any VAX system except VAX–11/725 or VAX–11/730, or it may be one of the following:

- MicroVAX II system with an RA series system disk.
- MicroVAX II system with an RD54 system disk, or VAXstation II system with an RD54 or any larger system disk. Note that these boot servers support a maximum of three satellites. In addition, it is recommended that the satellites use local RD series disks for paging and swapping.
- MicroVAX 3000 series system.

The *satellite nodes* are booted remotely from a boot server's system disk. Generally, these nodes are consumers of cluster resources, though they may also sometimes provide disk serving and batch processing resources. If satellite nodes are equipped with RD series disks, they may, for enhanced performance, use such local disks for paging and swapping.

Satellite nodes may be any of the following:

- MicroVAX II or MicroVAX 2000 system
- VAXstation II or VAXstation 2000 system
- MicroVAX 3000 series system

Figures 6–3, 6–4, and 6–5 illustrate typical local area cluster configurations.

Figure 6–3 Local Area VAXcluster Configuration with One Boot Server and One System Disk

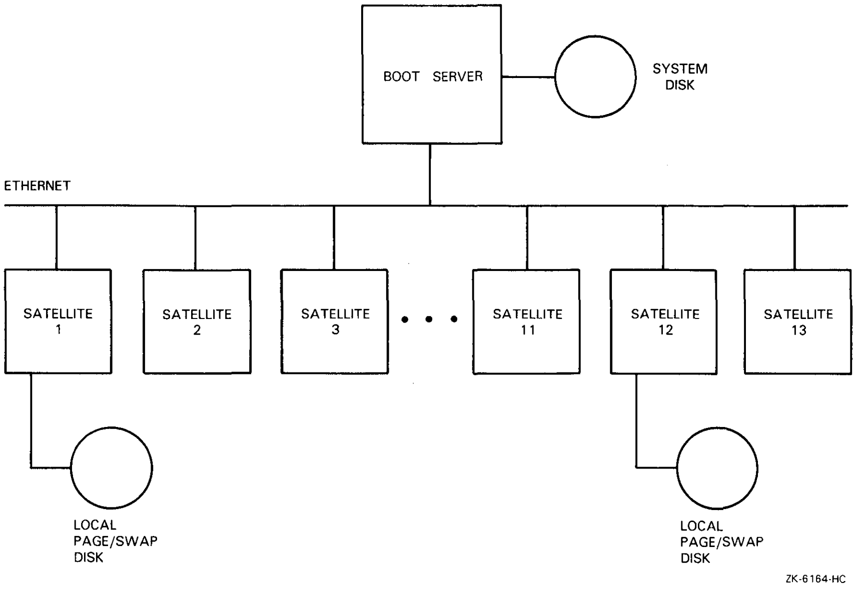


Figure 6–4 Local Area VAXcluster Configuration with One Boot Server and Two System Disks

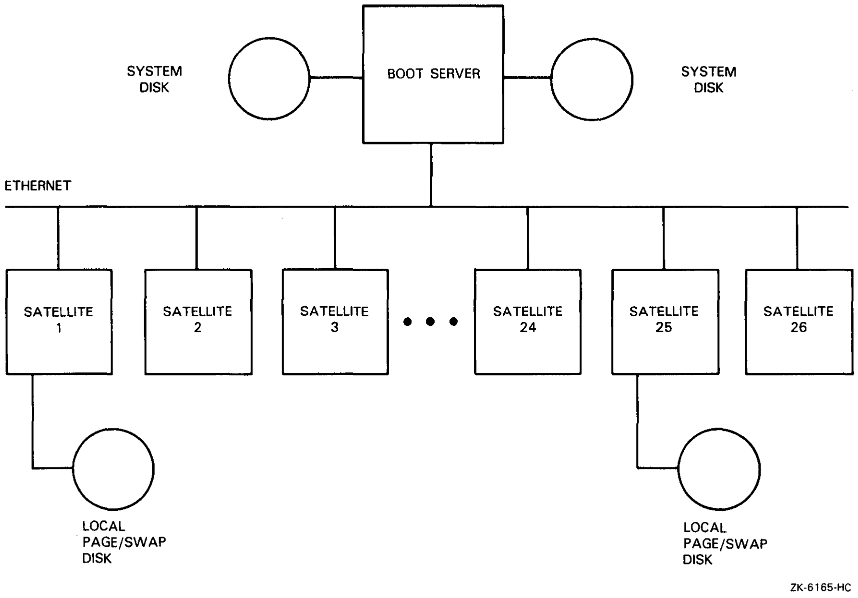
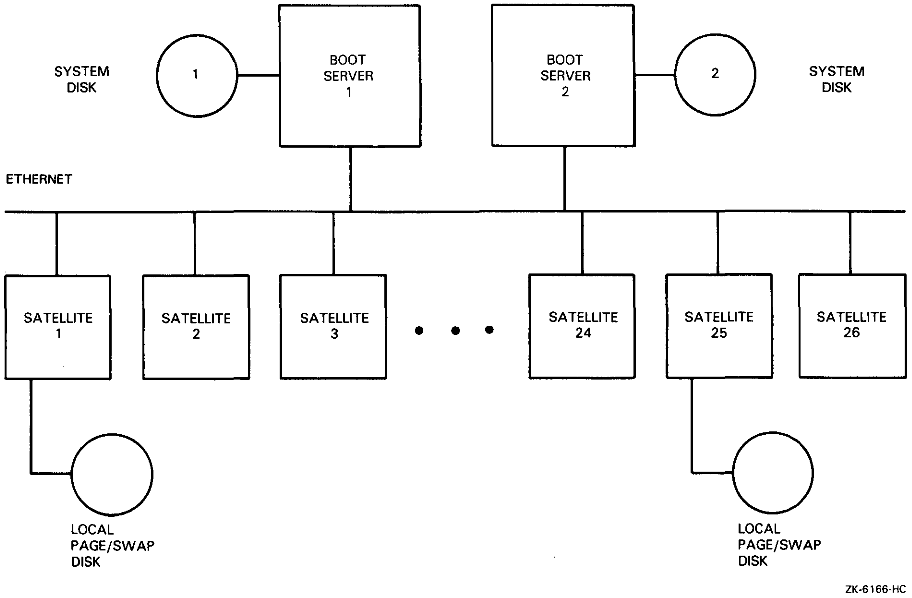


Figure 6–5 Local Area VAXcluster Configuration with Two Boot Servers and Two System Disks



6.4.3 Mixed-Interconnect VAXcluster Configurations

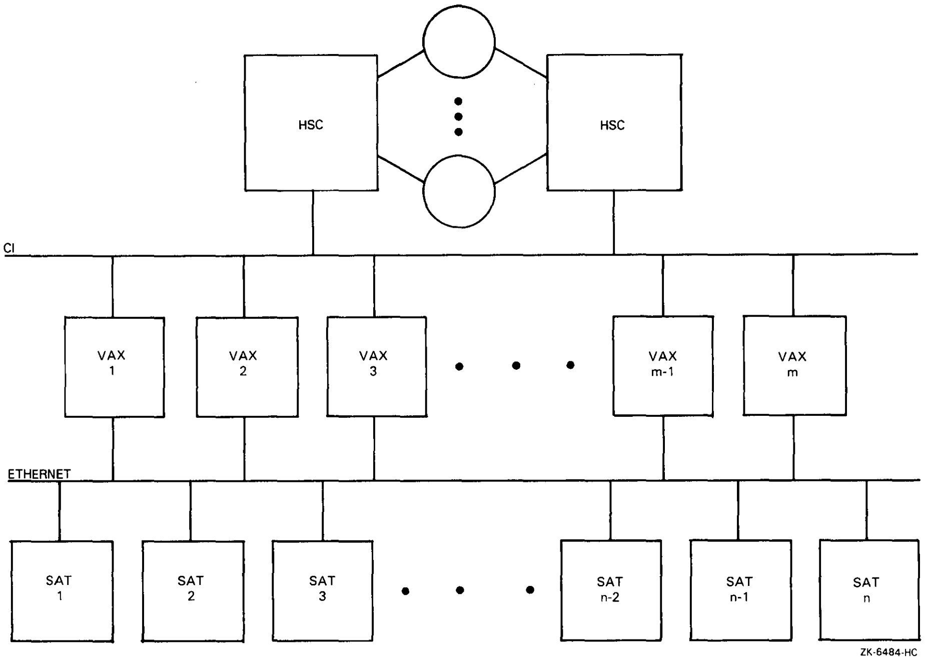
Clusters with both CI and Ethernet interconnects are available for the first time with VMS Version 5.0. Using both the Ethernet and the CI, a *mixed-interconnect* cluster may include both CI-connected VAX processors and MicroVAX systems.

Because the VMS Version 5.0 MSCP Server and disk class drivers (DUDRIVER and DSDRIVER) allow a CI-connected VAX processor to serve HSC disks, satellites can access the large amounts of storage available through HSC controllers.

HSC disk-serving functions are implemented with the SYSGEN parameters ALLOCLASS, MSCP_LOAD, and MSCP_SERVE_ALL. The ALLOCLASS (allocation class) parameter tells VMS software which disk controllers (HSCs or VAX processors running the MSCP Server) are connected to a common set of disks.

Figure 6–6 shows a typical mixed-interconnect configuration.

Figure 6–6 Typical Mixed-Interconnect VAXcluster Configuration



In Figure 6–6, a set of disks is dual-pathed to the HSC controllers named GOLF and TENNIS, and these controllers are connected to VAX processor HOBBIT. Because ALLOCLASS is set to the same value (1) on all three machines, HOBBIT can serve the disks on GOLF and TENNIS to all nodes in the cluster.

In this configuration, disks on the HSCs have *allocation class names* of the form $1$ddcu:. For example, the disk DUA17: is $1$DUA17:. Though VMS software would also recognize the disk as GOLF$DUA17:, TENNIS$DUA17:, or HOBBIT$DUA17:, an allocation class name like $1$DUA17: should always be used for configuration purposes, because it is the only name that will be unique in the cluster at all times.

Note that, for higher availability, two or more CI-connected VAX processors may serve HSC disks to the cluster. For example, if ALLOCLASS were set to 1 on node SNAIL, that node could also serve disks on the HSCs GOLF and TENNIS.

Appropriate settings for the SYSGEN parameters MSCP_LOAD and MSCP_SERVE_ALL enable a VAX processor to serve all suitable disks to other cluster members early in the boot sequence. The served disks thus become accessible to the other nodes with minimal interruption whenever the serving node reboots.

Further, the MSCP server automatically serves any suitable disk that is added to the system later. For example, if new drives are attached to an HSC controller, the disks become available within seconds after the cables are connected. For more information on automatic MSCP disk serving functions, refer to the *VMS VAXcluster Manual.*

6.4.4 Cluster Security for Local Area and Mixed-Interconnect Configurations

Local area and mixed-interconnect clusters use a *group number* and a *cluster password* to allow multiple independent clusters to coexist on the same Ethernet, and to prevent access to a cluster by unauthorized nodes.

- The *group number* uniquely identifies each mixed-interconnect and local area cluster on a single Ethernet. This number must be in the range from 1 to 4095 or from 61440 to 65535. Note that if you plan to have more than one of these clusters at your site, you must coordinate the assignment of group numbers among cluster managers.
- The *cluster password* serves as a second order check to ensure the integrity of individual clusters on the same Ethernet that accidentally use identical group numbers. (Provided that each cluster's password is unique, the clusters will form independently.) The password also prevents an intruder who discovers the group number from joining the cluster. The password must be from 1 to 31 alphanumeric characters in length and may include dollar signs and underscores.

Security data is maintained in the cluster authorization file, SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT. This file is created during installation of the Version 5.0 operating system, if you indicate that you want to set up a local area or mixed-interconnect cluster. The installation procedure then prompts you for the cluster group number and password. Cluster security functions are described in detail in the *VMS VAXcluster Manual.*

6.5 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 and mixed-interconnect 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.

6.6 Cluster Connection Management

Cluster integrity is dynamically controlled by a software component called the *connection manager,* which determines cluster membership and provides the coordination needed to manage it. It is the connection manager, for example, that creates a cluster when the first active node is booted, and that reconfigures the cluster when nodes join or leave it.

Cluster members can share various data and system resources, such as disk volumes. The integrity of shared resources, however, cannot be guaranteed unless the use of shared resources is carefully coordinated in the cluster. In the unlikely event that a pair of nodes share some resource, but are not members of the same cluster and therefore cannot coordinate the use of that resource, the integrity of the shared resource is not assured.

To achieve the coordination necessary to maintain resource integrity, the cluster nodes must share a clear sense of cluster membership. This sense of cluster membership is maintained by the connection manager, which prevents another cluster from sharing the same resources.

A cluster in which two autonomous nodes share the same resources is called a *partitioned cluster.* Cluster partitioning occurs when two active nodes that are intended to be members of the same cluster operate independently as members of two different clusters. Partitioning is undesirable, because resource sharing between two clusters is not coordinated. The connection manager prevents cluster partitioning using a scheme called *quorum.*

6.6.1 The Quorum Scheme

The quorum scheme is based on the arithmetic fact that the whole cannot be divided into multiple parts in such a way that more than one part is greater than half of the whole.

The quorum scheme functions as follows:

- Each non-satellite node in the cluster contributes a fixed number of votes toward a quorum. The votes value is specified by the SYSGEN parameter VOTES.
- Each active node in the cluster (including satellites) specifies an initial quorum value using the SYSGEN parameter EXPECTED_VOTES. This parameter is the sum of all VOTES held by cluster members.
- During certain cluster state transitions, the system dynamically computes the cluster quorum to be the *maximum* of the following:
  - The current cluster quorum value
  - The value for EXPECTED_VOTES specified by each node
  - The value calculated from the following formula, where V is the total of VOTES held by all cluster members

    ```
    (V+2)/2
    ```

  The cluster state transitions that cause cluster quorum to be recalculated occur when a node joins the cluster and when the cluster recognizes a quorum disk (see Section 6.6.2).
- If the current number of votes ever drops below the quorum (because of nodes leaving the cluster), the cluster members suspend all process activity and all I/O operations to cluster-accessible disks until sufficient votes are added (nodes joining the cluster) to bring the total number of votes to a value greater than or equal to quorum.
- As the cluster changes, the system only raises the cluster quorum value; it never lowers the value. (However, cluster managers can lower the value; for details, see the *VMS VAXcluster Manual.*)

Consider a cluster consisting of three nodes, each node having its VOTES parameter set to 1 and its EXPECTED_VOTES parameter set to 3. The connection manager dynamically computes the cluster quorum value to be 2. In this example, any two of the three nodes constitute a quorum and may run in the absence of the third node. No single node can constitute a quorum by itself. Therefore, there is no way the three cluster nodes can be partitioned and run as two independent clusters.

6.6.2 Quorum Disk

You may want to increase the availability of some cluster configurations (particularly two-node clusters) by establishing a quorum disk. A quorum disk acts as a virtual node in the cluster, adding votes (the minimum of the SYSGEN parameter QDSKVOTES settings on all nodes that have been cluster members) to the cluster votes total. For the quorum disk's votes to be counted in the cluster votes total, the following conditions must be met:

- On one or more cluster nodes you must specify the name of the quorum disk by setting the SYSGEN parameter DISK_QUORUM. The DISK_QUORUM value must be the same on each cluster node on which it is specified.
- The specified disk must have a direct (non-MSCP-served) connection to those nodes.
- The disk must contain a valid format file named QUORUM.DAT in the master file directory (MFD). The QUORUM.DAT file is created automatically after a system specifying a quorum disk has booted and run as a cluster member. On the initial booting of a system, the file will not be present; therefore, provisions for a quorum without the disk must be made.

When a quorum disk contributes to the cluster votes total, a two-node cluster with a shared HSC or MASSBUS disk can tolerate the failure of either one VAX node or the quorum disk.

Note that clusters with more than two VAX nodes may also use a disk as a virtual node; however, if your cluster is large enough so that a quorum disk would not significantly improve cluster availability, DIGITAL recommends that you do not use a quorum disk.

6.7 Shared Disk Resources

A major advantage of cluster configurations is the ability to make disk resources accessible to all cluster nodes. A *cluster-accessible* disk can be used by any active node in the cluster that successfully mounts it. A cluster disk that is not cluster accessible can be accessed only by the local node.

Cluster-accessible disks offer the following advantages:

- More efficient use of mass storage, because more than one node can use the same disk.
- Access by users to their default work disks when logging in to any node on which the disks are accessible.
- Clusterwide file sharing. Because nodes can share common versions of files, updates to a file are made only once to a single copy of the file.

- Implementation of clusterwide job controller queues. Batch and print jobs can be processed on any node that has access to the disks.

Procedures for setting up and managing cluster disks are described in the *VMS VAXcluster Manual.*

6.8 Shared Processing and Printer Resources

In any cluster configuration, nodes can share processing and printer resources. The ability to share resources allows for better workload balancing, because batch and print job processing can be distributed across the cluster.

Cluster managers control how jobs share batch processing and printer resources by setting up and maintaining clusterwide generic queues. The strategy used to set up and manage these queues will determine how well workloads are matched to available resources. Managers establish and maintain the queues with the same commands used to manage queues on a single-node system.

All clusterwide queues are controlled by a single, cluster common job controller queue file, which must be accessible to the nodes participating in the clusterwide queue scheme. This file makes queues available across the cluster and enables jobs to execute on any queue from any node—provided that the necessary mass storage volumes can be accessed by the node on which the job executes.

Procedures for setting up and managing cluster queues are described in the *VMS VAXcluster Manual.*

7 Networking

As the manager of a VMS system, you may want to connect your system to a network. This chapter explains what a DECnet network is and how a VMS system can be part of a DECnet network. It also summarizes the responsibilities of the system manager in a network environment.

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 all types of VMS systems, as well as with a wide range of non-VMS systems that support DECnet.

Systems connected to a DECnet network can communicate with each other without having to go through a central system. Any system can communicate with any other system connected to the network, not merely those systems physically attached to that system. This feature allows users to gain access to software facilities that may not exist on their particular system.

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.

7.1 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 configured 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 adheres to 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.

7.2 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 VMS. 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. Implementing network applications on VMS is straightforward, and network operations are efficient.

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 install a software key on your node.

7.3 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 reliable high-speed communications channel optimized to connect information processing equipment in a specific geographic area, such as an office, a building, or a complex 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. Ethernet provides a fast, efficient means of exchanging information and supports a high data rate.

DECnet–VAX also offers comprehensive wide-area network support and long-haul connectivity over point-to-point and multipoint 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.
- A multipoint configuration consists of two or more nodes connected by a synchronous DDCMP communications channel, with one node controlling the channel.

DECnet–VAX supports worldwide communications through packet switching networks and gateways. A DECnet–VAX node can be connected to a packet switching data network (either directly using VAX PSI software or through an X25router) to establish communication with a remote node. Packet switching networks (such as TYMNET™ and Telenet™) provide communication services between nodes on the same or different networks, often in widely dispersed geographic areas connected by satellite links. A DECnet–VAX node connected

™ TYMNET is a trademark of Tymnet Inc.

™ Telenet is a trademark of GTE Telenet Communication Corporation.

to an Ethernet can also use a DECnet/SNA gateway on the same Ethernet to communicate with IBM[TM] systems in SNA networks.

7.4 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.

[TM] IBM is a trademark of International Business Machines Corporation.

7.4 System and Network Manager Responsibilities

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.

Index

Index

Reader's Comments

Introduction to VMS
System Management
AA–LA24A–TE

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

Thank you for your assistance.

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

I would like to see more/less ____

What I like best about this manual is ____________________________________

What I like least about this manual is ___________________________________

I found the following errors in this manual:

Page Description

Additional comments or suggestions to improve this manual:

I am using **Version** ______ of the software this manual describes.

Name/Title ________________________________ Dept. ____________

Company ______________________________________ Date ____________

Mailing Address ____

____ Phone ____________

-- Do Not Tear - Fold Here and Tape --

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01–3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- Do Not Tear - Fold Here --

Reader's Comments

Introduction to VMS
System Management
AA–LA24A–TE

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

Thank you for your assistance.

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

I would like to see more/less

What I like best about this manual is

What I like least about this manual is

I found the following errors in this manual:

Page Description

Additional comments or suggestions to improve this manual:

I am using **Version** ______ of the software this manual describes.

Name/Title Dept.
Company Date
Mailing Address
Phone

-- Do Not Tear - Fold Here and Tape

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35 110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- Do Not Tear - Fold Here