VMS

digital

Guide to Setting Up a VMS System

Order Number AA–LA25A–TE

Guide to Setting Up a VMS System

Order Number: AA-LA25A-TE

April 1988

This manual provides system managers with the concepts and procedures needed to set up a VMS operating system for daily operation.

Revision/Update Information: This is a new manual.

Software Version: VMS Version 5.0

digital equipment corporation
maynard, massachusetts

April 1988

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

The following are trademarks of Digital Equipment Corporation:

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

ZK3385

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

FIGURES

TABLES

Preface

The *Guide to Setting Up a VMS System* provides system managers with the concepts and procedures needed to set up a VMS operating system for daily operation. After reading this manual, system managers should be able to perform the following tasks:

- Customize the operating system to meet site-specific requirements
- Set up user accounts and allocate resources
- Perform day-to-day operating procedures
- Use VMS utilities and DCL commands to perform routine system management functions

This manual is not intended to be a complete one-volume reference source of information. For information on operator-oriented and periodic tasks, see the *Guide to Maintaining a VMS System*. The utilities and commands used to perform specific system management tasks are described in detail in the individual VMS utility reference manuals and in the *VMS DCL Dictionary*. This manual does not include information on configuring VMS clusters; for that information, refer to the *VMS VAXcluster Manual*.

Intended Audience

This manual addresses experienced users of a VMS operating system who perform the functions of a system manager or operator.

Document Structure

The *Guide to Setting Up a VMS System* is organized into the following task-oriented chapters, each describing a different system management function:

- **Chapter 1**— Introduction
- **Chapter 2**— Customizing the Operating System
- **Chapter 3**— Starting Up and Shutting Down the System
- **Chapter 4**— Setting Up and Managing User Accounts
- **Chapter 5**— Controlling System Resources
- **Chapter 6**— Performing AUTOGEN and SYSGEN Operations
- **Chapter 7**— Connecting to a LAT Network
- **Appendix A**— Files on a VMS System Disk

Associated Documents

- For general background information about the system, see the *Introduction to VMS.*
- For instructions for performing processor-specific procedures (for example, installation, bootstrap operations, and backing up the console medium), see your VAX processor installation and operations guide.
- For information about how to use command procedures, see the *Guide to Using VMS Command Procedures.*
- For information on maintaining daily operations on your system, 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 *Guide to DECnet-VAX Networking.*
- For configuring and managing VAXclusters, see the *VMS VAXcluster Manual.*
- For information on performance tuning, see the *Guide to VMS Performance Management.*
- For information about the new System Management (SYSMAN) Utility, see the *VMS SYSMAN Utility Manual.* For detailed information on other VMS utilities, see the specific VMS utility manual.
- For supplemental reference information, see the *VMS DCL Dictionary* and the *VMS System Messages and Recovery Procedures Reference Volume.*

Conventions

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

New and Changed Features

- New System Management Utility (SYSMAN) available for facilitating the management of nodes and clusters.
- AUTOGEN V5.0 Changes and New Features
  - Feedback mechanism - AUTOGEN now uses feedback data from the running system to make its calculations.
  - Feedback report - A formatted feedback report is generated from the feedback data.
  - SAVE_FEEDBACK option - New SAVE_FEEDBACK option for the SHUTDOWN.COM system shutdown dialog. It records feedback data to be used in future AUTOGEN runs.
  - Alternate startup procedure - You can now specify an alternate startup procedure in MODPARAMS.DAT.
  - Current parameter values saved in VAXVMSSYS.OLD - AUTOGEN saves current parameter values in SYS$SYSTEM:VAXVMSSYS.OLD before updating them in SYS$SYSTEM:VAXVMSSYS.PAR.
  - OLDSITE*.DAT files obsolete - The MODPARAMS parameter passing mechanism has been enhanced to eliminate the need for OLDSITE*.DAT files.
  - File sizing - The way in which AUTOGEN sizes system page and swap files has been enhanced.
- Command Procedures That Execute at System Startup Time
  - New SYPAGSWPFILES Command Procedure for installing page and swap files
  - New SYLOGICALS Command Procedure for assigning systemwide logical names
  - The ADDUSER.COM, BACKUSER.COM and RESTUSER.COM command procedures, previously located in the SYS$MANAGER directory, are now available in the SYS$EXAMPLES directory. They can be used as templates for adding users to your system, backing up files, and restoring files.
  - New template files - The SYxxx.COM files are now supplied as templates (Two copies of the files, one with the file extension TEMPLATE and one with the file extension COM, are supplied in the distribution kit in SYS$MANAGER.)

1 Introduction

This manual provides guidelines and procedures for setting up your VMS operating system for daily operation. Your first responsibility as system manager is to get your VMS operating system up and running. At a minimum, you must install and boot the VMS operating system. For step-by-step instructions on installing your system and performing other processor-specific procedures, see your VAX processor installation and operations guide. The following is a brief overview of the tasks for setting up your system, which are described in detail in subsequent chapters:

- **Select a Boot Procedure**

  Before you can use your processor, you must boot (load) the operating system from the system disk into processor memory. You can perform either a *nonstop* or a *conversational* boot. You perform a nonstop boot if you do not want to stop to change system parameters during the boot procedure. You perform a conversational boot if you want to change system parameters during the boot procedure.

  Some VAX processors with console storage devices use a default boostrap command procedure (DEFBOO.CMD) to boot the system from the system disk. For these processors, you must select a default boot command procedure from those available on your console medium and copy it to DEFBOO.CMD. The method of booting the system depends on the type of VAX processor. See your VAX processor installation and operations guide for detailed booting instructions.

- **Build a site-specific startup command procedure**

  The DIGITAL-supplied command procedure SYS$SYSTEM:STARTUP.COM executes immediately after the operating system is booted. This procedure is called the site-independent startup procedure, and it contains commands that must execute at startup time in order for the system to run properly. You should *not* modify the commands in this file.

  Also included in your VMS distribution kit are several site-specific command procedures that are executed from within STARTUP.COM. These command files come as *template* files for you to modify or add commands to as the needs of your site dictate. One of the site-specific command procedures is called SYS$MANAGER:SYSTARTUP_V5.COM. You usually add commands to SYSTARTUP_V5.COM to perform routine system management operations that execute each time the system is started (for example, initializing and starting queues, and setting device characteristics). See Chapter 2 for more information on creating site-specific command procedures.

- **Make site-specific modifications for special configuration or workload needs**

  When you boot your system during installation, the AUTOGEN command procedure generates system parameters that are suitable for your hardware configuration. However, if you have an unusual hardware configuration or special workload requirements, you may want to modify some system parameters and then rerun AUTOGEN. See Chapter 6 for more information on AUTOGEN procedures.

- **Reboot the system to install modifications**

  Once you make the necessary site-specific modifications to the operating system, you must shut down the system and reboot it to ensure that the latest modifications are installed. Modifications to system parameters do not take effect until the system is shut down and rebooted. See Chapter 3 for information on VMS startup and shutdown procedures.

- **Back up the system**

  After installing and customizing your system, you perform various backup operations, such as backing up the console volume and backing up the system disk. You should back up the system disk whenever you make changes to it to ensure that you always have a copy of the latest contents. See your VAX processor installation and operations guide for detailed instructions on backing up your system.

- **Set up user accounts and control 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). User account records are maintained in the system UAF file, SYS$SYSTEM:SYSUAF.DAT. Each record consists of fields providing information about the account's identification, login characteristics, login restrictions, and resource control attributes.

  The VMS operating system uses the UAF to validate login requests and to set up processes for users who successfully log in to the system. You create, examine, and modify UAF records with the Authorize Utility. The following system resource control attributes are assigned in the UAF record:

  - Priority—A user's priority is the base software priority used in scheduling computer time for the process associated with the user's account.
  - 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).
  - Privileges— Privileges determine what functions users are authorized to perform on the system.
  - Identifiers—Access Control List (ACL) identifiers can be used to define a user's or group's access to system objects. Users are associated with ACL identifiers in the *rights database,* which contains all identifiers defined for the system.

2 Customizing the Operating System

After you have installed your VMS operating system, you can customize it for site-specific operations by performing the following operations:

- Creating site-specific command procedures that execute at startup time
- Adjusting the system for special configuration or workload requirements
- Building and copying the system disk

This chapter focuses on creating site-specific command procedures and building and copying the system disk. Your specific workload or system configuration may require that you modify system parameters or system file sizes (for example, to accommodate the installation of certain optional software products). Chapter 6 of this manual provides detailed information on modifying system parameters using AUTOGEN.

Your VMS distribution kit includes the following system startup command procedures. These command procedures are executed automatically at system startup:

- **SYS$SYSTEM:STARTUP.COM**—a file containing a series of procedures that must execute at system startup time in order for the system to run properly. STARTUP.COM is the site-independent startup command procedure supplied by DIGITAL. Do not modify this command procedure. The STARTUP.COM procedure invokes the site-specific procedures that are described in this section.
- **SYS$MANAGER:SYCONFIG.COM**—a template file supplied by DIGITAL to which you can add site-specific device configuration commands.
- **SYS$MANAGER:SYLOGICALS.COM**—a template file supplied by DIGITAL for defining logical names. This file contains a command procedure for adding system logical names for a MicroVAX that is not in a cluster. If your processor is not a standalone MicroVAX, you can ignore this procedure and add any systemwide logical name assignments to the end of this file.
- **SYS$MANAGER:SYLOGIN.COM**—a template file supplied by DIGITAL to which you can add commands that are executed whenever a user logs in.
- **SYS$MANAGER:SYSTARTUP_V5.COM**—a template file supplied by DIGITAL to which you can add various commands for setting up site-specific operations that are executed at startup time. The template contains MicroVAX-specific commands that you can modify to meet the needs of your processing environment.
- **SYS$MANAGER:SYPAGSWPFILES.COM**—a file supplied by DIGITAL to which you can add commands to install page and swap files on any disk.

Two versions of the template files are included in your VMS distribution kit: an executable version with the file extension COM, and a nonexecutable version with the file extension TEMPLATE. For example, your distribution kit contains two SYCONFIG files: SYS$MANAGER:SYCONFIG.COM and SYS$MANAGER:SYCONFIG.TEMPLATE.

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

The version with the file extension COM is executed by the system; you can edit this version to meet your site-specific needs. More information on STARTUP.COM and the site-specific command procedures is provided in the sections that follow. Before you make site-specific modifications, however, you must first log in to the system manager's account.

2.1 Logging in to the System Manager's Account

When you boot the system, a message is displayed on the terminal from which the system is booted. (A standalone workstation is an exception to this, because OPCOM is not automatically enabled at startup and you therefore do not receive the OPCOM message.) The message is similar to the following:

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

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

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

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.

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

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

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

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

If you have a MicroVAX processor in a standalone environment, by default the system displays a system manager menu when you log into the SYSTEM account. It then prompts you to select the system management operation you want to perform. See your MicroVAX processor installation and operations guide for a detailed description of menu options.

The command procedure SYS$MANAGER:MGRMENU.COM generates the system manager menu. For system managers who do not have a standalone MicroVAX, this command procedure can serve as a sample for designing site-specific system manager menus.

2.2 Site-Independent Startup Command Procedure (STARTUP.COM)

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

- Defines systemwide logical names required for the symbolic debugger, language processors, linker, image activator, and help processor.
- Starts processes that control error logging, SMISERVER (the system management server), the job controller, and the operator's log. (On a standalone workstation, the operator's log is not automatically started.)
- Connects and configures devices that are physically attached to the system and load their I/O drivers by invoking the SYCONFIG.COM procedure.
- Installs known images to reduce I/O overhead in activating the most commonly run images or to identify images that must have special privileges.

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

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

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

1. VMS$PHASES.DAT—This file determines the order of the phases of the startup procedure. It is a sequential list of the phases that will be started by STARTUP.COM. It includes a series of four basic phases (INITIAL, CONFIGURE, SYSFILES, and BASEENVIRON) needed to bring the VMS operating system up to a basic working environment, followed by a series of phases for layered products. This file must not be modified.
2. VMS$VMS.DAT—This is a component data file for starting the base VMS operating system environment. You should not modify this file.

3 VMS$LAYERED.DAT—This is a component file for layered products that are installed using the callback procedure of VMSINSTAL. It is an indexed-sequential file, containing the following fields for each file:

1 Name of the component file (either .EXE or .COM) to be run.

2 Phase in which the component file is to be run. The valid phases are LPBEGIN, LPMAIN (default), LPBETA, and END.

3 Method (or mode) by which the component file is to run. The valid choices are DIRECT (the default, where the command procedure or image is executed immediately), BATCH (valid only for command procedures), or SPAWN.

4 Node restrictions for the component. This is either the node or nodes on which the component file should *only* be run, or the node or nodes on which the component file should *not* be run.

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

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

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

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

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

If a layered product can use the callback procedure included in VMSINSTAL, then you can install it at system startup using the method described above, and you do not have to include the layered product in the site-specific startup file (SYSTARTUP_V5.COM). In these cases, the component files must be in the SYS$STARTUP directory. Layered products that do not use the callback procedure should be installed at system startup using SYSTARTUP_V5.COM.

You can also use the System Management Utility (SYSMAN) to manage the new startup process. With the STARTUP command of SYSMAN, you can add, modify, display, or remove elements of existing component files, create a new startup file, and perform other startup functions. See the *VMS SYSMAN Utility Manual* for more information about using SYSMAN.

Several site-specific command procedures are executed from within STARTUP.COM. You can add commands to these files or modify the template files supplied in your VMS distribution kit. Remember, however, to modify only the executable version of the file (with the file extension COM) and not the template version (with the file extension TEMPLATE).

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

1 SYS$MANAGER:SYPAGSWPFILES.COM

2 SYS$MANAGER:SYCONFIG.COM

3 SYS$MANAGER:SYLOGICALS.COM

4 SYS$MANAGER:SYSTARTUP_V5.COM

2.3 Site-Specific Procedure for Installing Page and Swap Files (SYPAGSWPFILES.COM)

DIGITAL supplies an empty file to which you can add site-specific commands to install page and swap files. This file, named SYS$MANAGER:SYPAGSWPFILES.COM, lets you install page and swap files on disks other than the system disk.

At boot time, the system activates the latest versions of SYS$SYSTEM:PAGEFILE.SYS, SWAPFILE.SYS, and SYSDUMP.DMP. If the page and swap files exist in SYS$SYSTEM, they are installed. If they do not exist in SYS$SYSTEM, the system displays an informational message such as the following, to indicate that the files are not present and that the boot operation continues:

```
%SYSINIT-I-PAGEFILE.SYS not found - system initialization continuing...
```

The STARTUP.COM procedure searches for and invokes SYPAGSWPFILES.COM before SYCONFIG.COM is invoked and before any of the system overhead processes are created (for example, OPCOM, JOBCTL, etc.). You place commands in SYPAGSWPFILES.COM to install page and swap files. This file may also include other commands such as INITIALIZE, SYSGEN CREATE, and MOUNT as necessary to define the paging device and files. You create secondary system files with the SYSGEN command CREATE. You install a secondary file by entering the SYSGEN command INSTALL to SYS$MANAGER:SYPAGSWPFILES.COM, for example:

```
$ RUN SYS$SYSTEM:SYSGEN
INSTALL DISK_SYS2:[SYSTEM]PAGEFILE1.SYS /PAGEFILE
INSTALL DISK_SYS2:[SYSTEM]SWAPFILE1.SYS /SWAPFILE
```

Note that disks other than the system disk are not yet mounted at the time SYPAGSWPFILES.COM is invoked. Therefore, you may need to add MOUNT commands to this file to mount the disks that contain the secondary page and swap files.

When control returns to STARTUP.COM after it has invoked SYPAGSWPFILES.COM, at least one page file must have been successfully installed; otherwise, STARTUP displays the following error message:

```
%STARTUP-E-NOPAGFIL, no page files have been successfully installed.
```

Caution: To use the primary page file for writing crash dumps, it must be located on the system disk. Also, disks that are mounted by SYPAGSWPFILES.COM must not be mounted by other processors while performing VMS upgrades with the SYSGEN parameter VAXCLUSTER set to zero.

2.4 Site-Specific Device Configuration Command Procedure (SYCONFIG.COM)

One step in the site-independent startup command procedure SYS$SYSTEM:STARTUP.COM is to connect the devices that are physically attached to the system and load their I/O drivers. As part of this step, STARTUP.COM first invokes the command procedure SYS$MANAGER:SYCONFIG.COM. (This command procedure comes with the software distribution kit as a template file.)

If you want to connect a nonstandard device (that is, a device not supplied by DIGITAL), you can do so by adding your own device configuration commands to SYCONFIG.COM.

In the following example, a command is added to SYCONFIG.COM to connect a nonstandard device called the QQ device:

```
$ SYSGEN := $SYSGEN
$ SYSGEN CONNECT QQA0/CSR=%Oxxxxxx/VECTOR=%Oxxx
```

You can also use SYCONFIG.COM to add site-specific MOUNT commands that you want executed at system startup time.

After SYCONFIG.COM executes, control is returned to STARTUP.COM. The following commands are then executed; they automatically connect all devices and load their drivers:

```
$ SYSGEN := $SYSGEN
$ SYSGEN AUTOCONFIGURE ALL
```

During autoconfiguration, a section of STARTUP.COM called CONFIGURE runs a program that creates a detached process to detect any devices connected to an Hierarchical Storage Controller (HSC), load their drivers, and make the devices known to the system. To suppress autoconfiguration, include the following command as the last line in SYCONFIG.COM:

```
$ STARTUP$AUTOCONFIGURE_ALL == 0
```

Caution: If you set STARTUP$AUTOCONFIGURE_ALL to zero in SYCONFIG.COM, the CONFIGURE section of STARTUP.COM will not execute.

To ensure proper configuration of devices connected to an HSC, DIGITAL recommends adding the following lines to the end of SYCONFIG.COM:

```
$ STARTUP$AUTOCONFIGURE_ALL == 0
$ @SYS$SYSTEM:STARTUP CONFIGURE
$ EXIT
```

This procedure suppresses autoconfiguration and then executes the commands in STARTUP.COM that start the CONFIGURE process.

2.5 Site-Specific Logical Names Command Procedure (SYLOGICALS.COM)

This file is supplied in SYS$MANAGER as a template containing a procedure for assigning systemwide logical names on a MicroVAX system that is not in a cluster. If your processor is not a standalone MicroVAX, this procedure has no effect.

Regardless of the type of processor you have, you can add your own systemwide logical name assignments to the SYLOGICALS.COM template file before the EXIT command. In addition to the logical names assigned automatically in STARTUP.COM, the system assigns your logical names when STARTUP.COM invokes SYLOGICALS.COM.

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

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

See the *VMS DCL Concepts Manual* for detailed information on logical name assignments and the privilege modes (executive, kernel, supervisor, and user).

2.6 Site-Specific Startup Command Procedure (SYSTARTUP_V5.COM)

The command procedure SYS$MANAGER:SYSTARTUP_V5.COM is invoked by STARTUP.COM to execute commands that perform site-specific operations. DIGITAL recommends that you edit the SYSTARTUP_V5.COM template to modify or add commands that perform tasks such as the following:

- Mounting public disks
- Setting the characteristics of terminals and other devices
- Initializing and starting queues
- Installing known images
- Starting the License Management Facility (LMF)
- Starting up the DECnet network
- Running the System Dump Analyzer
- Purging the operator's log file
- Submitting batch jobs that are run at system startup time
- Starting up layered products
- Creating systemwide announcements
- Starting up the LAT Network
- Defining the number of interactive users

The following sections describe how to create a site-specific startup command procedure to perform these tasks. Any commands shown are provided as examples only; you may need to alter them to meet the specific needs of your computing environment.

To minimize the processing overhead when executing SYSTARTUP_V5.COM, you can include the DCL command SET NOON at the beginning of the file. This command disables error checking after the execution of each command in SYSTARTUP_V5.COM.

2.6.1 Mounting Public Disks

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

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

The expression *ddcu* indicates that you supply the physical device name (including a colon immediately after the device name) in this parameter. When you mount a disk, the MOUNT command produces a special logical name called a logical volume name that you can use to reference the volume. Consider the advantages of using logical volume names to conceal the physical device names. If you and the users consistently use the logical volume name, it is not necessary to know on which physical drive the volume is mounted. Thus, you can avoid including physical device names in programs and command procedures.

The following command produces the logical volume name USER and equates it to DRA1, the device name. However, USER only translates to a physical device while the data disk is actually mounted. When you dismount the volume, the logical name is deleted from the system logical name table.

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

If you mount a disk and do not give an explicit logical volume name, MOUNT assigns a default logical name in the format DISK$volume_label. In the preceding example, if no logical volume name were specified, the default logical volume name would have been DISK$USERFILES. (The logical volume name is printed on the flag page of listings and displayed on the terminal by the DCL commands SHOW DEVICE/FILES and SHOW MEMORY/FILES; therefore, you may occasionally see such labels.)

Note that when SYSTARTUP_V5.COM is executed (and only then), the MOUNT command default includes the /NOASSIST qualifier. This qualifier means that operator-assisted mounts are disabled. To enable this feature during SYSTARTUP_V5, specify /ASSIST with each MOUNT command. Also, it is necessary to insert a WAIT statement in your SYSTARTUP_V5.COM prior to the first MOUNT statement for a DSA disk. The wait time is controller dependent. If this wait is omitted, the MOUNT request may fail with a "no such device" status. See the *VMS I/O User's Reference Manual: Part I* for more information.

2.6.2 Setting Device Characteristics

To establish the characteristics of the terminals and other devices on the system, use a series of SET commands in a command procedure. You may want to include comments that give the user names for terminal owners. For example:

```
$ SET TERMINAL TTC2: /SPEED=300/DEVICE_TYPE=LA36/PERMANENT  !JONES
$ SET TERMINAL TTD1: /SPEED=9600/PERMANENT                  !WRENS
$ SET TERMINAL TTD4: /SPEED=1200/PERMANENT                  !JRSMITH
$ SET TERMINAL TTG4: /SPEED=1200/MODEM/PERMANENT            !DIALUP1
```

The /SPEED qualifier sets both transmission and reception speeds to the same value. The /MODEM qualifier defines a terminal for use on a dial-in line. Printer characteristics (SET PRINTER and SET DEVICE) must be set prior to establishing queues for the printers. You may want to include SET TERMINAL commands in a separate file (for example, TERMSET.COM) and include a command in SYSTARTUP_V5.COM to invoke the TERMSET.COM command procedure. When the TERMSET.COM command procedure finishes executing, control returns to SYSTARTUP_V5.COM.

2.6.3 Initializing and Starting Queues

In SYSTARTUP_V5.COM, you can include commands to start the system job queue manager, establish *spooled* devices, and set up batch and output queues. Spooling devices is an efficient method of balancing the workload demand on line printers if you are running applications on a time-shared system. It directs the application output to an intermediate storage disk until the application program finishes, and then submits the file for printing. See the *Guide to Maintaining a VMS System* for guidelines for setting up spooled devices.

Initialize and start each queue with a separate INITIALIZE/QUEUE/START command line. The following are examples of commands that start the system job queue manager and initialize and start queues. For more examples, see the SYS$MANAGER:SYSTARTUP_V5.COM template.

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

Note: DIGITAL recommends using the /RESTART qualifier with the START/QUEUE/MANAGER command. This qualifier causes the queue manager to restart automatically in the event of a job controller abort.

On systems with a large number of queues, you may want to include queue commands in a separate file named, for example, STARTQ.COM, and include a command in SYSTARTUP_V5.COM to invoke the queue command procedure. When the queue command procedure finishes executing, control returns to SYSTARTUP_V5.COM. See the chapter on batch and print operations in the *Guide to Maintaining a VMS System* for more information.

2.6.4 Installing Known Images

It is important to install commonly used programs as known images to reduce the I/O overhead in activating those images and to assign attributes or privileges to the images.

All known images must be reinstalled each time the system is rebooted, because known file lists are not saved if the system is shut down or fails. For this reason, STARTUP.COM includes a series of INSTALL commands that install certain system programs as known images.

You should include additional INSTALL commands in SYSTARTUP_V5.COM to install those images that meet the following conditions:

- They are frequently run.
- They are usually run concurrently by several processes.
- They require special privileges.

By specifying appropriate qualifiers to INSTALL commands, you can assign any of the following attributes to known images:

- **Permanently open**—Directory information on the image file remains permanently resident, eliminating the usual directory search required to locate a file. The cost of keeping an image file permanently open is approximately one page of nonpaged dynamic memory per file.
- **Header resident**—The header of the image file (native images only) remains permanently resident, saving one disk I/O operation per file access. For images with single-block file headers, the cost is less than one page of paged dynamic memory per file; for images with multiblock headers, the cost varies according to the header block count. The images must also be declared permanently open.
- **Privileged**—Amplified privileges are temporarily assigned to any process running the image (executable images only), permitting the process to exceed its user authorization file (UAF) privilege restrictions during execution of the image. In this way, users with normal privileges can run programs that require higher than normal privileges.
- **Protected**—A shareable image contains protected code, that is, code that runs in Kernel or Executive mode but that can be called by a user-level image. Protected images must be declared shared.

- **Shared**—More than one user can access the read-only and noncopy-on-reference read/write sections of the image concurrently, so that only one copy of those sections ever need be in physical memory. (Copy-on-reference sections always require a separate copy for each process.) The image is implicitly declared permanently open.
- **Writable**—When a shared noncopy-on-reference writable section is removed from physical memory (for paging reasons or because no processes are referencing it), it is written back to the image file. Any updates made by processes mapped to the section, therefore, are preserved (while the initial values are lost). The image must also be declared shared.

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

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

For more information on installing images as known images, see the *VMS Install Utility Manual*.

2.6.5 Starting Up the License Management Facility

Most software that you can purchase, including VMS, layered products for VMS (for example, programming languages or application development tools), and products sold by third-party vendors, is sold under an agreement called a software license. For the purposes of this section, the term license refers only to the agreement that authorizes you to use a software product.

The software license describes the terms and conditions of the sales or rental agreement (for example, a simple purchase, a purchase with options, a lease arrangement, or other, more complicated, terms) between you and the software vendor. The purpose of the license is to authorize your use of the software product while protecting the proprietary rights of the software vendor.

As you acquire additional software for your system, you also have more licenses. Keeping track of these software licenses and their terms and conditions can be a time-comsuming task. The VMS License Management Facility (LMF), which you can start up in SYSTARTUP_V5.COM, can simplify the management of software licenses for your system. The LMF, which includes the License Management Utility (LICENSE), is a management tool provided with the VMS operating system with which you can keep track of your software licenses.

Specifically, you can accomplish the following license management tasks on line using the LMF:

- Register software license data in a database
- Display current and previous license data
- Control access to installed software products
- Include and exclude product access on different nodes in a VAXcluster
- Move a software license to another processor

- Maintain records related to software licenses
- Use software in compliance with the terms and conditions of a license agreement.

For information about how to use the LMF, see the *VMS License Management Utility Manual.*

2.6.6 Starting Up the DECnet Network

If you install DECnet software on your system, you must remove the comment delimiters from one of the commands in the SYSTARTUP_V5.COM template to start up the DECnet network. You must run NETCONFIG.COM before starting up the network, or the system returns error messages stating that the database is not initialized. (See the DECnet documentation for detailed information on setting up a network.)

To start up the DECnet software at system startup time, select one of the commands described in the SYSTARTUP_V5.COM template, as follows:

```
$!
$! If you have batch queues set up on your system, insert a comment delimiter
$! (!) after the dollar sign in the first command line below. Remove the
$! comment delimiter from the next second and third command lines below and
$! remove the dollar sign from the third command line below. This allows the
$! system to startup more quickly, and decreases the amount of time you must
$! wait to log in.
$!
$ IF F$SEARCH("SYS$SYSTEM:NETACP.EXE") .NES. "" THEN @SYS$MANAGER:STARTNET
$!
$! IF F$SEARCH("SYS$SYSTEM:NETACP.EXE") .NES. "" - !This is faster, if you
$! THEN SUBMIT SYS$MANAGER:STARTNET.COM            !have batch queues set up.
```

2.6.7 Running the System Dump Analyzer

Each time the system is booted, run the System Dump Analyzer (SDA) in case the system failed the last time it was running. You can do this by adding command lines to SYSTARTUP_V5.COM that are similar to the following:

```
$ ANALYZE/CRASH_DUMP SYS$SYSTEM:SYSDUMP.DMP
COPY SYS$ERRORLOG:SYSDUMP.DMP
SET OUTPUT LPA0:SYSDUMP.LIS
SHOW CRASH
SHOW STACK/ALL
SHOW SUMMARY
SHOW PROCESS/PCB/PHD/REGISTERS
EXIT
```

For further information, invoke the System Dump Analyzer for an interactive session upon completion of startup. (See the *VMS System Dump Analyzer Utility Manual.*)

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

2.6.8 Purging the Operator's Log File

Each time the system is rebooted, a new version of OPERATOR.LOG is created. You should devise a plan for regular maintenance of these files. Adding the following command to your site-specific startup command procedure purges all but the last two versions of the operator's log file:

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

See the *Guide to Maintaining a VMS System* for additional suggestions for maintaining the operator's log file.

2.6.9 Submitting Batch Jobs That Are Run at Startup Time

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

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

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

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

See the chapter on batch and print operations in the *Guide to Maintaining a VMS System* for more information on submitting batch jobs.

2.6.10 Starting Up the LAT Network

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

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

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

To configure a node as a service node that connects only to interactive terminals on a terminal server, one line is required in SYSTARTUP_V5.COM. You can include arguments to the @SYS$MANAGER:LTLOAD command that define the characteristics of the VMS service node.

However, to use remote printers on a terminal server or to create dedicated application services on the VMS service node requires modification of LTLOAD.COM. See Section 2.7 for details. For more information on the LAT protocol, see Chapter 7.

Supporting User Terminals on a Terminal Server

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

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

The arguments P1 through P4 have the following meaning:

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

The argument P1 assigns a service name to the node, using the LATCP command CREATE SERVICE. Arguments P2 through P4 can be any valid qualifier to the SET NODE command. For a full description of LATCP commands and qualifiers, see the *VMS LAT Control Program (LATCP) Manual.*

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

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

2.6.11 Creating Systemwide Announcements

Usually, the last command in SYSTARTUP_V5.COM announces to all terminals that the system is up and running:

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

Before SYSTARTUP_V5.COM exits, you can provide site-specific definitions for one or both of the following logical names: SYS$ANNOUNCE and SYS$WELCOME. Whenever a user logs in, the messages associated with SYS$ANNOUNCE and SYS$WELCOME are displayed on the user's terminal screen.

Defining SYS$ANNOUNCE

You can define SYS$ANNOUNCE to print an announcement at the beginning of the login procedure for each user. The text is printed immediately after a successful dial-in, CTRL/Y, or RETURN is received. The text can contain up to 63 characters. For longer messages, precede the name of a text-containing file with an at sign (@) so that the login command procedure prints the entire file as an announcement.

For example, you could include the following command in your SYSTARTUP_V5.COM file:

```
$ DEFINE/SYSTEM SYS$ANNOUNCE "SIRIUS CLUSTER AT ANDROMEDA, INC."
```

Or you might prefer to print a file by including the following command:

```
$ DEFINE/SYSTEM SYS$ANNOUNCE "@SYS$MANAGER:ANNOUNCE.TXT"
```

If you do not define SYS$ANNOUNCE, no announcement is displayed.

Defining SYS$WELCOME

You can define SYS$WELCOME to display a welcome message whenever a user logs in. The text is printed immediately after the correct password is entered. The text may contain up to 63 characters. For longer messages, precede the name of a text-containing file with an at sign (@) so that the login command procedure displays the entire file as a welcoming announcement.

For example, you could include a command such as the following in your SYSTARTUP_V5.COM file:

```
$ DEFINE/SYSTEM SYS$WELCOME "WELCOME TO THE VMS OPERATING SYSTEM"
```

If you prefer to display the contents of a file containing a message, you could use the following line in SYSTARTUP_V5.COM:

```
$ DEFINE/SYSTEM SYS$WELCOME "@SYS$MANAGER:WELCOME.TXT"
```

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

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

You can add the DECnet–VAX node name to this message by including a translation of the logical name SYS$NODE. DECnet creates the logical name assignment for SYS$NODE when it is started up.

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

2.6.12 Defining the Number of Interactive Users

To change the default value for the number of interactive users that you permit to log in to your system at one time, include the following command in SYSTARTUP_V5.COM:

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

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

When SYSTARTUP_V5.COM terminates, control is returned to STARTUP.COM, which checks whether the number of logins was set by the above command. If the command was used to specify a value, that value is used. Otherwise, STARTUP.COM sets the number to 64 by default and then exits.

Note: You cannot set the number of interactive users to a value above that which is authorized by your VAX processor license.

The maximum number of interactive users established influences the service rating that the LAT protocol assigns to a VMS service node. The LAT protocol uses a ratio of current users to maximum users in calculating a rating. An artificially high user limit results in a high service rating, indicating – erroneously—that the VMS node is most able to provide services.

2.7 Site-Specific LAT Command Procedure (LTLOAD.COM)

LTLOAD.COM, located in the directory SYS$MANAGER, is the command procedure that starts up the LAT protocol on a VMS service node. You execute LTLOAD.COM from SYSTARTUP_V5.COM.

You must modify LTLOAD.COM under the following three conditions:

- Using remote printers connected to terminal servers.
- Creating special application services.
- Tailoring VMS node characteristics; for example, assigning special service announcements or Ethernet links.

Note: To create a LAT configuration where the VMS service node supports only interactive terminals does not require modification of LTLOAD.COM. You can assign a service name and other characteristics in the SYSTARTUP_V5.COM command line.

To customize your VMS service node, modify the following commands in LTLOAD.COM. The commands create the VMS service node and assign characteristics to it; they also create ports to support remote printers or special applications. For a full description of the commands shown in LTLOAD.COM, see the *VMS LAT Control Program (LATCP) Manual*.

```
$ ! This command procedure starts up the LAT protocol and configures
$ ! logical devices to support remote printers or special applications.
$ !
$ ! Invoke SYSGEN to load the LAT port driver:
$ !
$ RUN SYS$SYSTEM:SYSGEN
CONNECT LTA0: /NOADAPTER
$
$ ! Invoke the LATCP Utility
$ !
$ RUN SYS$SYSTEM:LATCP
!
! Set up the LAT service node, using the default name SYS$NODE and the
! default identification SYS$ANNOUNCE or your own names and identification.
!
SET NODE /IDENT  /NOLOG
CREATE SERVICE /IDENT /NOLOG
!
! Set up logical ports to support remote printer access.
!
! First create the ports on the VMS service node, inserting your own names.
!
CREATE PORT LTA1: /NOLOG
!
CREATE PORT LTA2: /NOLOG
!
! Next map the logical ports to specific ports on the terminal server.
!
SET PORT LTA1: /APPLICATION /NODE=terminal-server-name /PORT=port-name
SET PORT LTA2: /APPLICATION /NODE=terminal-server-name /PORT=port-name
!
! Start the node.
!
START NODE
EXIT
```

Creating a VMS Service

You must create at least one service in LTLOAD.COM. It can be a primary service, one through which users can access the general computing environment. Or it can be a special application service, such as a data entry program or an online news service. When you are creating an application service, DIGITAL recommends that you assign the name of the application program, for example:

```
CREATE SERVICE NEWS /IDENT /NOLOG
```

Setting Up Ports

The second set of commands in LTLOAD.COM creates logical ports on the VMS service node and associates them with physical ports on the terminal server node. If you are establishing a special application service, include the /DEDICATED qualifier when defining a LAT port. The application program, which the service connects with, must define the same dedicated port. For example, the following commands set up ports for an application service called NEWS:

```
CREATE PORT LTA1: /DEDICATED
SET PORT LTA1: /DEDICATED /SERVICE=NEWS
```

Starting Queues or Application Programs

Once LTLOAD.COM is modified, an additional step is needed to complete the configuration of remote printers or application services on a VMS service node. To use printers connected to a terminal server, you must write a DCL command procedure to create spooled output queues for the LAT printers. You can integrate the LAT queue commands into a systemwide command procedure, which starts all queues. Usually, you execute this command procedure from SYSTARTUP_V5.COM after the queue manager is started. See *Guide to Maintaining a VMS System* for a description of configuring remote printers on a terminal server.

Before application services can be available to user terminals on the LAT network, you must start the application program. You usually do this from SYSLOGIN.COM.

2.8 Site-Specific System Login Command Procedure (SYLOGIN.COM)

As system manager, you usually create and maintain a standard login command procedure, SYS$MANAGER:SYLOGIN.COM, which is executed each time a user logs in. This file is supplied on your VMS distribution kit as a template, which includes commands that you can modify and add to as the needs of your site dictate.

The SYS$MANAGER:SYSTARTUP_V5.COM template includes the following command line that assigns the logical name SYS$SYLOGIN to SYLOGIN.COM:

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

This logical name assignment is needed in order for this procedure to execute whenever a user logs in. See Example 4–2 of Chapter 4 for a sample SYLOGIN command file.

2.9 Symmetric Multiprocessing (SMP)

A multiprocessing system consists of two or more CPUs that address a common pool of memory and are capable of executing instructions simultaneously.

2.9.1 Overview of VMS Multiprocessing

VMS Version 5.0 supports a tightly coupled, symmetrical multiprocessing (SMP) system. In a tightly coupled SMP system, all processors execute a single copy of VMS and have equal access to all operating system code and system resources. As a result, in a tightly coupled SMP system, the scheduler can assign a job to any processor on the basis of which processor is free to execute the job, regardless of the requirements of the job or the system resources it must access. This capability is known as *dynamic load leveling*.

A VMS multiprocessing system can function as an isolated entity, a node in a network, or a member of a VAXcluster. VMS multiprocessing and uniprocessing systems run the same VMS operating system, although multiprocessing can be enabled only on selected processors in the VAX family. All processors in a VMS multiprocessing environment must be at the same hardware and firmware level to guarantee that a given processor is

capable of resuming the execution thread of a process that had been executing previously on another processor in the system.

2.9.1.1 Primary and Secondary Processors

In a VMS multiprocessing system, one processor has the responsibility of starting other processors in the system. The *primary* processor is that processor in the system that is either logically or physically attached to the console device. As such, it is the processor that is the target of the console commands that bootstrap the VMS multiprocessing system. In this role, only the primary processor performs the initialization activities that define the VMS environment and prepare memory for the entire system. In addition, the primary processor serves as the system timekeeper, maintaining the system time and monitoring the timer queue for the expiration of its elements. In this sense, all processors in a multiprocessing system that do *not* have these responsibilities are known as *secondary* processors.

2.9.1.2 Available and Active Sets

VMS uses the term *available set* to identify those processors that have passed the system's power-on hardware diagnostics and may or may not be actively involved in the system. Together, the primary and the secondary processors comprise the multiprocessing system's active set. The *active set* is the subset of the VAX system's processors that have passed its power-on diagnostics and are actively participating in system operations. VMS identifies each processor in these sets by its *CPU ID*, a value prevalent in the syntax and displays of certain DCL and utility commands. In a VAX 8300 system, for instance, the CPU ID is the VAXBI node number of the processor; in a VAX 8800, the CPU ID of the left processor is *1* and that of the right processor is *0*.

2.9.1.3 Processor Capabilities

The processors in a VMS multiprocessing system offer certain *capabilities* to the processes executing in the system. VMS Version 5.0 defines two capabilities: primary and timekeeper. Sometimes, a process must be scheduled on a processor that provides a specific, needed capability. (In VMS Version 5.0, the primary CPU and no other can have the primary and timekeeper capabilities.) Similarly, processes can exist that must be executed on a specific processor. An example of the latter is the VMS process that executes in response to a request to stop a given processor.

2.9.2 Creating the Multiprocessing Environment

The system manager can control the membership and character of a VMS multiprocessing system at boot time by using the System Generation Utility (SYSGEN) and specifying SYSGEN parameters designed for these purposes. Among the new SYSGEN parameters in VMS Version 5.0 that manage a VMS multiprocessing system are the following:

Parameter	Function
MULTIPROCESSING	Determines which synchronization image is loaded into the operating system at boot time
SMP_CPUS	Determines which processors are brought into the multiprocessing environment at boot time

For more information about these and other SYSGEN parameters, see the *VMS System Generation Utility Manual*.

The system manager can add an available processor to the active set at boot time, or can add it later using the DCL command START/CPU. The DCL command STOP/CPU removes a processor from the active set.

2.9.3 Monitoring the Multiprocessing Environment

Several features of VMS Version 5.0 provide special information about the character, capabilities, and status of a VMS multiprocessor system. They include the DCL command SHOW CPU, the Monitor Utility, and the system log files.

2.9.3.1 Obtaining Information About a Multiprocessor Configuration

The SHOW CPU command displays three levels of information describing the configuration and status of a VMS multiprocessing system:

Level	Command Example	Display Contents
Summary	SHOW CPU	Indicates which processor is the primary, which are configured, and which are active; displays the minimum revision levels for processors in the system and the setting of the MULTIPROCESSING SYSGEN parameter; and indicates whether multiprocessing is enabled and whether the presence of a uniprocessing device driver in the system prohibits its functioning as a multiprocessor.
Brief	SHOW CPU/BRIEF	Produces information from the summary display; lists the current CPU state and the current process (if any) for each configured processor.
Full	SHOW CPU/FULL	Produces information from the summary display; lists the current CPU state, current process (if any), revision levels, and capabilities for each configured processor; indicates which processes can be executed only on certain processors; lists the names of uniprocessing device drivers present in the system (if any) that prohibit its function as a multiprocessor.

For more information about the DCL commands relating to SMP, see the *VMS DCL Dictionary;* for information about the Monitor Utility, see the *VMS Monitor Utility Manual.*

2.10 Backing Up the System

Once you have installed and customized your system, DIGITAL recommends that you perform the following sequential operations:

1 Back up the console volume

2 Build a standalone backup kit

3 Back up the system disk

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

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

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

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

2.11 Building and Copying a VMS System Disk

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

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

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

You may want to move your operating system files to another disk. For example, assume that your operating system is initially stored on an RK07 disk together with some of your user files. Suppose that you have purchased an RP06 disk, and that you want to move only the operating system files from the RK07 disk to the RP06 disk. You can build the operating system on the RP06 disk (which is called the target, or destination, disk) without

affecting the user files on the RK07 disk (the source disk) by using the BUILD option of the VMSKITBLD command procedure.

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

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

2.11.1 Building the Operating System on Another Disk

If you want to build your operating system on another disk and you are not concerned about losing the current contents of the target disk, use the BUILD option (or COMMON option for a cluster-common system disk) as described in the following procedure:

Note: VMSKITBLD automatically dismounts the target disk when the kit is completed.

1. Boot the operating system from the current system disk (source disk), typically an RK07 disk.
2. Log in to the system manager's account.
3. Set the default to the SYS$UPDATE directory:

   ```
   $ SET DEFAULT SYS$UPDATE
   ```

4. Place the target disk (assuming you are using a removable disk) in an appropriate drive and put it on line.
5. Enter the following command to invoke VMSKITBLD:

   ```
   $ @VMSKITBLD
   ```

6. In response to the VMSKITBLD prompts, enter the required information about the source and target disk drives.
7. After you enter the required information, VMSKITBLD prompts you to choose one of the following options:

   ```
   Operation [BUILD,ADD,COPY,COMMON]?
   ```

   Enter BUILD to build your target disk as a single-node system disk, or enter COMMON to build your target disk as a cluster-common system disk. When the build resumes, VMSKITBLD displays messages that either prompt you for information needed to complete the operation or inform you of the procedure's status.

A typical message sequence for a single-node system is as follows: (Note that if you are building a cluster-common system disk, the source disk specified in response to the first prompt would be SYS$COMMON.)

```
Enter mounted source disk name (ddcu:): SYS$SYSDEVICE:
Enter SOURCE disk top level system directory [default = SYS0]: RET
Enter target disk name (ddcu:): DRA0: RET
Enter the target disk's label [default = VAXVMSRL5]: RET
Enter TARGET disk top level system directory [default = SYS0]: RET
  It will be necessary to initialize the target disk.
Is the target disk, DRA0:, ready to be initialized? (Y/N): Y
  _DRA0: allocated
%MOUNT-I-MOUNTED, VAXVMSRL5   mounted on _DRA0:
Create directory entries on the target disk.
Copy the system executive files.
Copy the system library files.
Copy the system message files.
Copy the system manager files.
Copy the system update command files.
Copy the system EXE files.
Copy the system help files.
Write a bootblock.
Copy BLISS require files and STARLET DCL library.
Copy coding examples.
Copy the UETP files.
```

When the system disk is built, VMSKITBLD automatically dismounts the disk and then sends the following message to your terminal:

```
System disk complete.
```

8 At this point, the target disk contains all the required VMS files for a complete system. You should then configure the system with appropriate system parameters by booting from the target disk with the default parameters and then using the AUTOGEN procedure to configure the target system. You can do this using the *conversational boot procedure*, which allows you to change parameters in the course of the boot. Follow the conversational boot procedure, found in your VAX processor installation and operations guide, until the system displays the following prompt:

```
SYSBOOT>
```

9 When the SYSBOOT> prompt appears, enter the following commands:

```
SYSBOOT> USE DEFAULT
SYSBOOT> CONTINUE
```

10 After the system boots, log into the system manager's account and enter the following command to run the AUTOGEN procedure:

```
$  @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

See Chapter 6 for detailed information on AUTOGEN.

2.11.2 Copying the Operating System Files to Another Disk

You can also use VMSKITBLD to copy the operating system files to a target disk without deleting the files already on it. In order to do this, the VMS operating system must be running, and the source disk that you intend to copy from must be mounted. Remember, the user-modified files are *not* copied from the source disk; VMSKITBLD uses the unaltered TEMPLATE versions of the site-specific command files.

To copy the operating system files from the source disk to a target disk, use the following procedure:

1. Log in to the system manager's account.
2. Place the target disk on an appropriate drive.
3. Create the new system directories on the target disk using the DCL command CREATE/DIRECTORY. To find out the names of the system directory files, enter the following DCL command:

   ```
   $ DIRECTORY DUA0:[SYS0]
   ```

4. Check the number of free blocks on the target disk to be sure there is enough space to copy the operating system (using the DCL command SHOW DEVICE).
5. Note the device name of the target disk.
6. Set the default to the SYS$UPDATE directory:

   ```
   $ SET DEFAULT SYS$UPDATE
   ```

7. Enter the following command to invoke VMSKITBLD:

   ```
   $ @VMSKITBLD
   ```

8. In response to the VMSKITBLD prompts, enter the requested information about the source and target disk drives.
9. After you supply this information, VMSKITBLD prompts you to choose one of the following options:

   ```
   Operation [BUILD,ADD,COPY,COMMON]?
   ```

   Enter the word COPY.

10. After you reply, VMSKITBLD displays the following messages that either prompt you for information needed to complete the copy operation or inform you of the procedure's status.
11. When the copy operation is finished, VMSKITBLD automatically dismounts the target disk and displays the following message:

    ```
    System disk complete.
    ```

2.11.3 Adding an Operating System to an Alternate System Root Directory

You use the ADD option to add another copy of the operating system to an unused alternate root directory on the same system disk. Some cases where you may want to have two operating systems available on the same system disk are as follows:

- To create a test environment

  If you want to test software on the operating system without interfering with the current version of the system, you could use the ADD option to copy the operating system to an alternate system root directory and create a boot command procedure to select that version for testing sessions.

- To conserve disk drives

  You could add a copy of the current operating system to an alternate root directory on the same system disk.

Use the following procedure to add a copy of the operating system to an alternate root directory:

1. Log in to the system manager's account.
2. Check the number of free blocks on the system disk to be sure you have room to add another operating system.
3. Enter the following command to set the default to the SYS$UPDATE directory:

   ```
   $ SET DEFAULT SYS$UPDATE
   ```

4. Place the target system disk (assuming you are using a removable disk) in an appropriate drive and put it on line.
5. Enter the following command to invoke VMSKITBLD:

   ```
   $ @VMSKITBLD
   ```

6. In response to VMSKITBLD prompts, supply the requested information.
7. After you supply the information, VMSKITBLD prompts you to choose one of the following options:

   ```
   Operation [BUILD,ADD,COPY,COMMON]?
   ```

   Enter the word ADD.

8. You will receive messages that either prompt you for information needed to complete the operation or inform you of the procedure's status.

   When you are prompted for the mounted source disk name, enter SYS$SYSROOT: if the source is a common root; otherwise, enter SYS$SYSDEVICE: as the source disk logical name.

   When you are prompted for the source disk's top-level system directory, enter the directory from which you are copying the system files.

   When you are prompted for the target disk's top-level system directory, be sure to enter a directory root not already in use. (Note that in addition to roots used by existing systems on the disk, roots SYSE and SYSF are reserved for other system functions.)

A typical message sequence might look like this:

```
Enter mounted source disk name (ddcu:): SYS$SYSROOT:
Enter SOURCE disk top level system directory [default = SYS0]:
SYS5 [RET]
Enter target disk name (ddcu:): SHEMP$DUA1: [RET]
Enter the target disk's label [default = VAXVMSRL5]: [RET]
Enter TARGET disk top level system directory [default = SYS0]: SYSA [RET]
  Allocate and mount target disk.
  _SHEMP$DUA1: allocated
%MOUNT-I-MOUNTED, VAXVMSRL5   mounted on _SHEMP$DUA1:
Create directory entries on the target disk.
Copy the system executive files.
Copy the system library files.
Copy the system message files.
Copy the system manager files.
Copy the system update command files.
Copy the system EXE files.
Copy the system help files.
Write a bootblock.
Copy BLISS require files and STARLET DCL library.
Copy coding examples.
Copy the UETP files.
```

VMSKITBLD informs you when the operation completes by displaying the following message on your terminal:

```
System disk complete.
```

At this point, the target system directory contains all the required VMS files for a complete system. (Note that if any optional products from the source system are needed on the new system, you must reinstall them.)

9 Boot the new copy of the operating system. See your VAX processor installation and operations guide for detailed instructions on booting the system from an alternate root directory.

10 Shut down the system and halt your VAX processor. Then boot the system using the command procedure you just created.

If you are running a VAXcluster, it is important to verify and, if necessary, reset values (using SHOW and SET commands) for the following SYSGEN parameters before rebooting:

- ALLOCLASS
- DISK_QUORUM
- EXPECTED_VOTES
- SCSSYSTEMID
- SCSSYSTEMIDH
- SCSNODE
- VAXCLUSTER
- VOTES

For more information on these parameters, see the *VMS System Generation Utility Manual* and the *VMS VAXcluster Manual.*

11 After the system boots, log into the system manager's account and run the AUTOGEN procedure. (See Chapter 6 for information on AUTOGEN functions.)

3 Starting Up and Shutting Down the System

During routine system operation, you often shut down and restart your system to perform various maintenance functions. For example, you may need to modify system parameters or make hardware modifications. Also, if the system fails and does not automatically restart, you must manually boot the system to restart it. This chapter gives a general overview of the procedures that DIGITAL provides for starting up and shutting down a VMS operating system.

Some startup and shutdown procedures vary for different VAX processors. See your VAX processor installation and operations guide for the specific procedures that apply to your VAX processor.

3.1 Boot Procedures

Booting is the process of loading system software from the system disk into processor memory. You must have installed the VMS operating system before you boot the system for the first time.

The booting procedure is processor specific. Some VAX processors with console storage devices use a default boot command procedure to boot the system; others have an internal memory device that provides the name of the system disk during boot operations. See your VAX processor installation and operations guide for detailed booting instructions for your VAX processor.

For processors that use a boot command procedure, select either a nonstop or a conversational boot command procedure to boot the system. You perform a nonstop boot if you do not want to stop to change system parameters during the boot operation. You perform a conversational boot if you want to change system parameters during the boot operation.

The most common booting operation is a nonstop boot from the system disk. If the system disk drive is broken and you have a copy of the VMS operating system on an alternate system disk, perform a nonstop boot from the alternate system disk.

A conversational boot is common in programming research and development environments where you must alter operating conditions for experimentation, testing, and debugging. Perform a conversational boot to obtain the SYSBOOT> prompt. You can then enter SYSGEN commands at the SYSBOOT> prompt to change the value of SYSGEN parameters or to specify alternate system startup procedures.

During a conversational boot operation, the experienced user can perform the following tasks:

- Specify a minimum startup
- Select an alternate file as the source of system parameter values
- Set and show individual parameter values
- Specify an alternate site-independent startup procedure

Starting Up and Shutting Down the System

3.1 Boot Procedures

Table 3–1 lists the SYSGEN commands that you can enter at the SYSBOOT> prompt. For more information on SYSGEN commands, see the *VMS System Generation Utility Manual.*

Table 3–1 SYSGEN Commands Used in SYSBOOT

Command	Description
CONTINUE	Resumes the boot procedure
DISABLE CHECKS	Inhibits checking of parameter values specified with the SET command
ENABLE CHECKS	Permits checking of parameter values specified with the SET command
HELP	Displays a summary of the SYSBOOT commands at your terminal
SET parameter-name value	Establishes the value of a system generation parameter
SET/STARTUP	Sets the name of the system startup command procedure. The default startup procedure is SYS$SYSTEM:STARTUP.
SHOW [parameter]	Displays active, current, default, maximum, and minimum values for specific system parameters; displays, with appropriate qualifiers, characteristics of parameters grouped by categories
USE [file-spec]	Optionally specifies a parameter file to be used as a source of values (you must enter the entire file specification, including device and directory)

3.2 Emergency Startup Procedure

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

3.2.1 Bypassing the User Authorization File

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

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

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

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

1. Obtain the SYSBOOT> prompt by following the instructions in your processor installation and operations guide for performing a conversational boot procedure. (Follow the procedure only to the point where you receive the SYSBOOT> prompt.)
2. Set the UAFALTERNATE system parameter by entering the following command in response to the SYSBOOT> prompt:

   ```
   SYSBOOT> SET UAFALTERNATE 1
   ```

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

   The system assigns the following values to your user account:

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

   The process name is usually the name of the device on which you logged in (for example, _OPA0).
5. Fix the problem that caused you to be locked out of the system. That is, make the necessary repairs to the UAF or to the startup or login procedures. If you cannot solve the problem, restore the procedure to its previous state.

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

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

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

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

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

7 Shut down and reboot the system.

3.2.2 Emergency Startup After Modifying System Parameters

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

1 Perform a conversational boot by following the instructions in your VAX processor installation and operations guide.

2 When the SYSBOOT> prompt appears, enter the following commands:

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

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

3.2.3 Bypassing Startup and Login

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

1 Perform a conversational boot operation by following the instructions in your VAX processor installation and operations guide.

2 Define the console to be the startup procedure by entering the following command at the SYSBOOT> prompt:

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

Type CONTINUE and press the RETURN key in response to the next SYSBOOT> prompt. Wait for the DCL prompt to return.

3 Correct the error condition that caused the login failure. That is, make the necessary repairs to the startup or login procedures, or to the UAF. You may want to enter the following DCL commands because bypassing the startup procedures leaves the system in a partially initialized state:

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

Invoke a text editor to correct the file. Note that some system consoles may not supply a screen-mode editor. You can also copy a corrected file and delete the incorrect version by using the RENAME and DELETE commands.

4 Reset the startup procedure by invoking SYSGEN and entering the following commands:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET/STARTUP SYS$SYSTEM:STARTUP.COM
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

5 Perform a normal startup by entering the following command:

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

3.2.4 Startup Problems

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

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

1 Consult the hardware manual for your VAX processor.

2 Contact your DIGITAL Field Service representative.

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

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

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

3.3 Shutdown Procedures

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

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

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

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

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

3.3.1 Orderly Shutdown with SHUTDOWN.COM

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

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

3.3.1.1 SHUTDOWN.COM Sequence of Prompts and Messages

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Entering the SAVE_FEEDBACK option records feedback data collected from the system since it was last booted. This option creates a new version of the AUTOGEN feedback data file, which can be used when you next run AUTOGEN. (See Chapter 6 for detailed information on using the AUTOGEN feedback mechanism.)

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

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

(For more information on cluster shutdown options, see the *VMS VAXcluster Manual.*) Otherwise, only the REBOOT_CHECK and SAVE_FEEDBACK options are listed. Enter REBOOT_CHECK to verify the presence of a subset of files necessary to reboot the system after shutdown completes. (If you are certain that the files exist, press RETURN.)

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

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

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

1. A message requesting users to log out is broadcast at decreasing time intervals to all users on the system.
2. The system logical name SHUTDOWN$TIME is defined as the absolute time of shutdown. For example, if the value 10 is specified at 12:00 in response to the first question, the logical name is assigned the absolute time value 12:10 along with the date. By requesting the logical name definition for SHUTDOWN$TIME (with the SHOW LOGICAL command), you can see if a shutdown is in progress or determine the actual time of shutdown. This feature is useful if a user missed the shutdown broadcast message.

3 At six minutes or less until system shutdown, the terminal from which shutdown was invoked is made an operator's console and all future nonoperator logins are disabled. If the DECnet network is running, it is shut down.

4 When there is one minute left until system shutdown, batch and device queues and the system job queue manager are stopped.

5 At zero minutes before system shutdown, the site-specific command procedure SYS$MANAGER:SYSHUTDWN.COM is invoked.

6 All user processes are stopped; however, system processes continue. Ancillary Control Processes (ACPs) may delete themselves when their mounted volumes are finally dismounted.

7 For dual-processor systems, the secondary processor is stopped.

8 All installed images are removed.

9 All mounted volumes are dismounted and, if you request it, the disks are spun down. Note, however, that the system disk cannot be spun down. Also, the quorum disk (if present on your system) is not dismounted or spun down.

10 The operator's log file is closed.

11 The program SYS$SYSTEM:OPCCRASH is invoked to shut down the system.

12 If you did not request an automatic reboot, the following message appears on the system console:

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

If you requested an automatic reboot, the system reboots, provided the necessary controls are set.

13 If you are not automatically rebooting, halt the system after the previous message is printed at the console terminal.

Example 3–1 demonstrates an orderly system shutdown on standalone node AVALON.

Example 3–1 Orderly System Shutdown with SHUTDOWN.COM

```
$ @SYS$SYSTEM:SHUTDOWN

        SHUTDOWN -- Perform an Orderly System Shutdown

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

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

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

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

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

17 terminals have been notified on AVALON.

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

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

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

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

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

Example 3–1 Cont'd. on next page

Example 3–1 (Cont.) Orderly System Shutdown with SHUTDOWN.COM

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

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


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

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

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

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

3.3.1.2 Defining SHUTDOWN$INFORM_NODES

You can define the logical name SHUTDOWN$INFORM_NODES to equate to a list of cluster member nodes that will be notified when the system is shutdown. To do so, you must define SHUTDOWN$INFORM_NODES before executing SYS$SYSTEM:SHUTDOWN.COM, as shown in the following example:

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

If you define SHUTDOWN$INFORM_NODES and then execute SYS$SYSTEM:SHUTDOWN.COM, all cluster member nodes included in the list are notified of the shutdown. Users on the node that is being shutdown are always notified regardless of whether you define SHUTDOWN$INFORM_NODES. If you omit the name of the node that is being shutdown from the list specified in the DEFINE command, the name is automatically added to the list by the SHUTDOWN.COM procedure.

The following table indicates what nodes are notified at different phases of the shutdown sequence, depending on whether SHUTDOWN$INFORM_NODES is defined.

Shutdown Phase	Notification if SHUTDOWN$INFORM_NODES is not defined	Notification if SHUTDOWN$INFORM_NODES is defined
First shutdown notification	Notify all terminals on all nodes	Notify all terminals on all listed nodes
First shutdown notification to two minutes before final shutdown	Notify all logged in users on the node that is shutting down	Notify all logged in users on all listed nodes
Two minutes before final shutdown notification to final shutdown	Notify all logged in users on all nodes	Notify all logged in users on all listed nodes
Shutdown canceled	Notify all terminals on all nodes	Notify all terminals on all listed nodes

3.3.2 Emergency Shutdown with OPCCRASH

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

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

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

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

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

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

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

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

4 Halt the system.

Example 3–2 illustrates an emergency shutdown using the OPCCRASH procedure.

Example 3–2 Emergency Shutdown Using OPCCRASH

```
$ RUN SYS$SYSTEM:OPCCRASH
     SYSTEM SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM

CTRL/P
>>>HALT
     HALTED AT 8000708A
```

4 Setting Up and Managing User Accounts

Managing the access and resource usage of the users on your system is a primary function of system management. The proper management of user accounts is essential to system security and the effective use of system resources.

To manage users effectively, you must determine the following:

- Who needs access to the system?
- What tasks must they perform?
- What system resources do they need?
- What information do they need from you?

Once you understand user needs, you can establish controls that customize the system appropriately. The VMS operating system provides several tools to authorize and control the use of system resources by individual users. This chapter describes these tools and when to use them.

4.1 The User Authorization File (UAF)

You manage users on a VMS operating system by creating and maintaining user accounts. The information stored with these accounts is used to control who can log in to the system and how system resources can be used. You use the Authorize Utility (AUTHORIZE) to do the following:

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

See the *VMS Authorize Utility Manual* for a detailed description of the Authorize Utility.

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

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

Setting Up and Managing User Accounts

4.1 The User Authorization File (UAF)

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

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

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

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

The procedures for adding a user account are discussed in detail later in this chapter. Because the UAF is the prime repository for storing information about user accounts, it is important to understand its components before you add accounts.

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

Example 4–1 Sample UAF Record Display

```
Username: WELCH                          Owner:  ROB WELCH    ❶
Account:  INVOICE                        UIC:    [21,51] ([INV,WELCH])
CLI:      DCL                            Tables: DCLTABLES    ❷
Default:  USER3:[WELCH]
LGICMD:
Login Flags:  Diswelcome Disnewmail                           ❸
Primary days:    Mon Tue Wed Thu Fri
Secondary days:                    Sat Sun
Primary   00000000001111111112222  Secondary 00000000001111111112222
Day Hours 01234567890123456789023  Day Hours 01234567890123456789023
Network:  ------ No access -------            ----- Full access ------
Batch:    XXXXXXXXX--------XXXXXXX             ---------XXXXXXXXX------
Local:    XXXXXXXXX--------XXXXXXX             ---------XXXXXXXXX------
Dialup:   ----- Full access ------             ------ No access -------
Remote:   ----- Full access ------             ------ No access -------
Expiration:            (none)    Pwdminimum:  6   Login Fails:     0
Pwdlifetime:           (none)    Pwdchange:   15-APR-1988 13:58
Last Login:            (none) (interactive),          (none) (non-interactive)
Maxjobs:         0  Fillm:        20  Bytlm:         8192      ❹
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:           0
Maxdetach:       0  BIOlm:        10  JTquota:       1024
Prclm:           2  DIOlm:        10  WSdef:          150
Prio:            4  ASTlm:        10  WSquo:          256
Queprio:         4  TQElm:        10  WSextent:       512
CPU:        (none)  Enqlm:       100  Pgflquo:      10240
Authorized Privileges:                                        ❺
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX
Identifier                    Value              Attributes ❻
```

Example 4–1 Cont'd. on next page

Example 4–1 (Cont.) Sample UAF Record Display

```
PROJECT_X                       %X8001001E      RESOURCE NODYNAMIC
DOCU_PROC                       %X80010044      NORESOURCE NODYNAMIC
```

❶ **User identification fields** contain information used by the system for accounting purposes and user identification.

❷ **Default fields** contain the default specifications for the following:

- Command language interpreter (CLI) is DCL by default.
- Name of the command procedure to be executed automatically at login time. If the field is blank, SYS$LOGIN:LOGIN.COM is executed by default.
- Command language interpreter tables (if blank, same as CLI field).
- Device and directory names for default file access.

❸ **Login characteristics fields** impose specific login restrictions that do the following:

- Inhibit certain login functions
- Control the days of the week when various types of logins are permitted
- Control the times of day when various types of logins are permitted

❹ **Resource control fields** control system resources by

- Limiting the use of reusable system resources
- Specifying the base priority used in scheduling the process that the system creates for the user

❺ **Privileges fields** specify the privileges that allow use of restricted and sensitive system functions.

❻ **Identifier fields** lists the ACL identifiers that the user holds and that are recorded in the rights database file.

To gain access to a specific user record, set the default directory to SYS$SYSTEM, enter the command RUN AUTHORIZE to invoke the Authorize Utility, and enter the command SHOW username at the UAF> prompt. You can then enter AUTHORIZE commands such as ADD and MODIFY to create or change the information in the fields of the UAF record. See Chapter 5 for a list of privileges, limits, and quotas that you can specify in the resource control and privileges fields of the UAF record.

4.1.1 System-Supplied UAF Records

The Authorize Utility provides a set of commands and qualifiers to assign values to any field in a UAF record. The discussion of the Authorize Utility in the *VMS Authorize Utility Manual* explains each field in the UAF record and describes the commands and qualifiers used to assign attributes to these fields.

The software distribution kit provided with a new VMS operating system contains a UAF of four records:

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

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

  Section 4.3 gives an example of how to use AUTHORIZE to add a user account. Section 4.8.1 explains how to create and use additional default templates.

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

- **FIELD**—Permits DIGITAL Field Service personnel to check out a new system. The FIELD record should be disabled once the system is installed.
- **SYSTEM**—Provides a means for you to log in with full privileges. The SYSTEM record can be modified but cannot be renamed or deleted from the UAF.

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

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

4.1.2 General Maintenance of the UAF

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

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

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

- **SYSTEM, FIELD, and SYSTEST accounts**—Change the passwords immediately. Use obscure passwords of six characters or more and continue to change them on a regular basis. You should not permit general users access to these accounts.

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

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

  The login flag DISUSER disables the account and prevents anyone from logging into the account. To enable the account when it is needed, run AUTHORIZE and enter the following:

  MODIFY username /FLAGS=NODISUSER.

Caution: **Be careful not to disable all of your privileged system accounts. If you inadvertently do so, you can recover by setting the UAFALTERNATE SYSGEN parameter during a conversational boot operation. See Chapter 3 for information on emergency startup procedures.**

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

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

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

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

If you want to receive mail sent to the SYSTEM account, define a systemwide logical name in the site-specific command procedure (SYS$MANAGER:SYLOGICALS.COM) to equate SYSTEM to the user name of the system manager's account. Alternatively, you could log in to the SYSTEM account and then use the SET FORWARD command in MAIL to forward SYSTEM mail to another account.

4.2 Preparing to Add User Accounts

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

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

In conjunction with adding a user account, do the following:

1. Select a user name and password.
2. Select a user identification code (UIC).
3. Decide where the account's files will reside (which device and directory)
4. Use the System Management Utility (SYSMAN) to add a disk quota entry for this UIC, if disk quotas are in effect. This can be done only after you have created the user's account with the Authorize Utility.
5. Create a default directory on the appropriate volume, using the following DCL command:

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

6. Determine the security needs of the account (that is, the level of file protection, privileges, and access control).
7. Establish any login/logout command procedures.

These tasks are described in detail in the sections that follow. When you have completed the tasks for preparing to add a user account, you are ready to add the account by following one of the methods described in Sections 4.3 and 4.4.

4.2.1 User Name and Password

To determine a user name and password, use naming conventions that take into consideration the nature of the account. For example, some installations use the name of the person who will use the account. Captive accounts, on the other hand, often use a name that describes the function of the account. Thus, an interactive account for Robert Jones might have a user name of JONES, while a captive account for an inventory system might be called INVENTORY. On systems with a large number of users, remember to assign unique user names.

For interactive accounts, it is best to let the person using the account control the password. Initially, provide a simple password and tell the user to change the password with the DCL command SET PASSWORD. Only the person using the account need know the password. Encourage all users to set obscure passwords of at least six characters and to change them frequently.

To ensure that the user changes the password, use the /PWDEXPIRED qualifier with the AUTHORIZE command ADD or MODIFY to set a password expiration date in the UAF. You can also specify the FORCE_PWD_EXP_CHANGE keyword with the /FLAG qualifier for the ADD or MODIFY commands. This keyword forces a user to change the password immediately upon logging in.

To provide additional protection to a user account, a secondary password can be added. The initial setting of the secondary password is controlled by the system manager using the Authorize Utility. See the *VMS Authorize Utility Manual* for information on adding and modifying secondary passwords.

Also, you can use the /PWDMINIMUM and /PWD_LIFE qualifiers when setting up the UAF record for an account to enforce timely password modifications. The /PWDMINIMUM qualifier specifies the minimum password length in characters (default is 6). The /PWD_LIFE qualifier is used to specify a delta-time value. After that date, the system issues a warning message to the user and the password expires if it has not been changed.

For captive accounts, the degree of sensitivity of the data used by the account should determine the type of password. For example, the password for a payroll application should be obscure, while the password for a suggestions account might not even be required; it could be null (in which case users would not be prompted for the password).

You should prohibit users from changing the passwords of captive accounts. To do this, specify /FLAGS=LOCKPWD when you invoke the Authorize Utility. Change the password whenever you feel it might be compromised (for example, if a person using the account moves to another job). To change a user's password, enter the command MODIFY user-name/PASSWORD at the UAF> prompt.

4.2.2 User Identification Code (UIC)

In general, you assign each account a unique user identification code (UIC). A UIC has two formats: numeric and alphanumeric. The numeric UIC consists of a group identifier and a member identifier separated by a comma and enclosed within square brackets (for example, [11,200]). These identifiers may also appear as alphanumeric characters, consisting of a member name and, optionally, a group name (for example [DOCO,PRICE].

You should assign accounts the same group number if they perform similar work, access the same files frequently, or use many of the same logical names. See the *VMS DCL Concepts Manual* for a detailed discussion of the user identification code.

4.2.3 Disk Quota Entry

If disk quotas are in effect for a disk volume, run the System Management Utility (SYSMAN) and use the DISKQUOTA command to add an entry for the new UIC. Disk quotas limit the amount of disk space available to individual users on a particular volume. To add a disk quota entry, follow the steps in this example:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/NODE=LARRY
SYSMAN> DISKQUOTA ADD [014,JONES] /DEVICE=DISK$USER -
_SYSMAN> /PERMQUOTA=2000 /OVERDRAFT=500
SYSMAN> EXIT
```

The commands in this example do the following:

- Invoke SYSMAN.
- Define the management environment to be node LARRY.
- Add a disk quota entry on the volume DISK$USER for UIC [014,JONES]. The entry has a permanent quota of 2000 blocks and an overdraft of 500 blocks.
- Exit from the utility.

The sum of the quota and overdraft values is the absolute maximum number of blocks allotted to the user, which in this example is 2500 blocks. For more information on the Sysman Utility and establishing disk quotas, see the *VMS SYSMAN Utility Manual*.

4.2.4 User Default Device and Directory

For each interactive account, you should create a top-level (default) directory (using the DCL command CREATE/DIRECTORY). In the directory place a login file, login file template, and/or logout file, as appropriate. The interactive user creates and maintains files and subdirectories in this directory. Make the owner of the directory the UIC for the new account. Usually, you also use the name of the account for the default directory. For example, if you have decided on an account name of JONES and a UIC of [014,1], you would enter the following DCL command to create a default directory for the account on the volume DISK$USER:

```
$ CREATE/DIRECTORY DISK$USER:[JONES]/OWNER_UIC=[014,1]
```

The volume on which the directory is established depends on which devices you reserve for interactive accounts and how much space is available on each.

The default file specification you provide the new account (when you run AUTHORIZE) should be the name of the device and the name of the top-level directory you used in the DCL command CREATE/DIRECTORY.

For a captive account, whether you create a top-level directory depends on the nature of the user system. Where the user system uses files in a particular directory, you should make that directory the default directory specification. For example, if the inventory system uses the files DISK$DATA:[INV]STOCK1.DAT and DISK$DATA:[INV]STOCK2.DAT, the default device specification should be DISK$DATA:, and the default directory specification should be [INV].

4.2.5 Account Security Considerations

The level of security that you establish for an account depends on the purpose of the account and whether it is shared with more than one user or group. For an interactive user account, the default UIC-based protection is usually adequate. No WORLD access is the default protection for top-level directories. Users can further protect their files and subdirectories on an individual basis with the DCL command SET PROTECTION.

However, in some cases (such as project accounts) you may want to set up an additional level of protection by using access control lists (ACLs). 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 where necessary, because they can consume additional amounts of paged system dynamic memory when files are open. They may also require additional processing time.

ACLs offer a way to grant or deny users access to specific objects such as files, directory files, global sections, devices, system logical name tables, and queues. The rights database (RIGHTSLIST.DAT) is a file that associates users of the system with access controlling identifiers. When a user logs in, the system checks the rights list for the identifiers that the user holds. You use the AUTHORIZE Utility to maintain the rights database by adding, granting, or deleting ACL identifiers as needs dictate.

By allowing a group of users to hold common ACL identifiers, the system manager can create a group protection scheme that is more intricate than the user's UIC-based protection.

Section 4.6 describes how to set up a project account with ACL-based protection. For more information on how to set up and edit ACLs, see the *VMS Access Control List Editor Manual*.

4.2.6 Login Command Procedures for Interactive Accounts

For interactive accounts, login command procedures contain commands commonly executed at the beginning of every user session. These commands do such tasks as

- Define symbols
- Assign logical names
- Display messages and the time of day
- Set terminal characteristics
- Define keys to perform certain functions

Login command procedures are useful for saving keystrokes and standardizing operations. In establishing login command procedures for interactive accounts, you have the following choices:

- **System**—As system manager, you normally create and maintain a standard login command procedure in the system directory (the file is usually named SYS$MANAGER:SYLOGIN.COM). You then assign the logical name SYS$SYLOGIN to the name of the file so that whenever a user logs in, the procedure is executed.

- **Individual**—For any or all accounts, you may specify an additional login command procedure with the /LGICMD qualifier of AUTHORIZE command ADD, MODIFY, or COPY. You can give the login command procedure any valid file specification. Whenever the user logs in, the additional procedure is executed after SYS$SYLOGIN.
- **User-specified command file**—If individual or system login command procedures are not implemented, the system looks for a command file called LOGIN in the user's login directory (as defined by the UAF record device and directory fields). If the file is found, the system executes it. This command file is developed and maintained by the user and should follow these conventions:
  - Device and directory names must take the default file specification for the account.
  - The file name and extension must be LOGIN.COM.

  As an aid to new users, you might copy a login command procedure template into newly created top-level directories. However, to ensure proper ownership of the file, change the owner UIC of the file to that of the user. You make this change with the DCL command SET FILE/OWNER_UIC.

Examples 4–2 and 4–3 illustrate typical system and user-specified login command procedures.

Example 4–2 Sample SYS$MANAGER:SYLOGIN.COM Login Command Procedure

```
$ V = F$VERIFY(0)
$START:
$ !
$ SET NOCONTROL=Y         ! Do not allow CTRL/Y to exit procedure
$ SET NOON
$ !
$ !     Set default file protection back to the old default
$ !
$ SET PROTECTION=(SY:RWED,OW:RWED,GR:RWED,WO:RE)/DEFAULT
$ !
$ !     Allow network jobs to start faster
$ !
$ IF F$MODE() .EQS. "NETWORK" THEN GOTO EXIT
$ !
$ !     Enable CTRL/T handling by DCL
$ !
$ SET CONTROL=T
$ !
$ !     Define Foreign Commands For Installed Utilities
$ !
$ SDA               ==    "ANALYZE/CRASH_DUMP"
$ USERS             ==    "SHOW USERS"
$ DISPLAY           ==    "MONITOR PROCESSES/TOPCPU"
$ NCP               ==    "$NCP"
$ INFO              ==    "SHOW PROCESS/CONTINUOUS"
$ SUSPEND           ==    "SET PROCESS/SUSPEND"
$ RESUME            ==    "SET PROCESS/RESUME"
$ SETNAME           ==    "SET PROCESS/NAME"
$ !
```

Example 4–2 Cont'd. on next page

Example 4–2 (Cont.) Sample SYS$MANAGER:SYLOGIN.COM Login Command Procedure

```
$ !     Define a symbol indicating whether the terminal
$ !     is on a dialup port
$ !
$ TT == F$GETDVI("TT","DEVNAM")-"_"
$ DIALUP == ((TT .GES. "TTGO:" .AND. TT .LES. "TTG4:") -
        .OR. (TT .GES. "TTH1:" .AND. TT .LES. "TTH4:") -
        .OR. (TT .EQS. "TTI5:"))
$ IF DIALUP THEN SET TERMINAL/INQUIRE
$ !
$EXIT:
$ IF V THEN SET VERIFY
   .
   .
   .
$ SET CONTROL=Y
$ EXIT
```

As Example 4–2 shows, you can disable the CTRL/Y function (which suspends execution of the current image and invokes the command interpreter) to force execution of the complete login command procedure whenever the user logs in. You do this with the DCL command SET NOCONTROL=Y. Before the login command procedure exits, you should add the command that resets the CTRL/Y function, the DCL command SET CONTROL=Y.

Example 4–3 Sample Login Command Procedure (LOGIN.COM)

```
$ SET NOON
$ SET PROTECTION=(S=RD,O=RWED,G=R,W=R)/DEFAULT
$ !
$ ! Define abbreviations for often used commands
$ !
$ DIR*ECTORY    ==     DIRECTORY/DATE/SIZE
$ PH*ONE        ==     PHONE/SCROLL
$ !
$ !
$ ! Other useful abbreviations
$ !
$ SHP           ==     "SHOW PROCESS/PRIVILEGES"
$ PRI*NT        ==     "PRINT/NOTIFY"
$ SHD           ==     "SHOW DEFAULT"
$ UP            ==     "SET DEFAULT [-]"
$ SP            ==     "SET PROCESS/PRIVILEGES="
$ SQ            ==     "SHOW QUEUE/BATCH/ALL/DEVICE"
$ H*OME         ==     "SET DEFAULT SYS$LOGIN"
$ SUB*MIT       ==     "SUBMIT/NOTIFY"
$ SPC           ==     "SHOW PROCESS/CONTINUOUS"
$ SYS           ==     "SHOW SYSTEM"
$ DAY           ==     "SHOW TIME"
$ !
```

Example 4–3 Cont'd. on next page

Example 4–3 (Cont.) Sample Login Command Procedure (LOGIN.COM)

```
$ ! Set /LOG for all commands
$ !
$ BACK*UP       ==      "BACKUP/LOG"
$ DEL*ETE       ==      "DELETE/LOG"
$ LIB*RARY      ==      "LIBRARY/LOG"
$ PUR*GE        ==      "PURGE/LOG"
$ REN*AME       ==      "RENAME/LOG"
$ !
$ ! End of LOGIN.COM processing
$ !
$ GOTO 'F$MODE()
$NETWORK:
$ EXIT
$INTERACTIVE:
$ VN            ==      "SET TERMINAL/WIDTH=80"
$ VW            ==      "SET TERMINAL/WIDTH=132"
$ EXPERT        ==      "SET MESSAGE/NOFACIL/NOSEVER/NOIDENT"
$ NOVICE        ==      "SET MESSAGE/FACILITY/SEVERITY/IDENTIF"
$ NOVICE
$ !
$ ! Symbols for network users
$ !
$ SYSA          ==      "SET HOST SYSA"
$ SYSB          ==      "SET HOST SYSB"
$ SYSC          ==      "SET HOST SYSC"
$ EXIT                          ! End of interactive login
$BATCH:
$ SET VERIFY                    ! End of batch login
$ EXIT
```

4.2.7 Login Command Procedures for Captive Accounts

For captive accounts, the login command procedure specified by the /LGICMD qualifier of AUTHORIZE directs the account user into an application program and logs the user out upon termination of the task. A simple login command procedure for an inventory system, for example, might consist of the following commands:

```
$ DEFINE SYS$DISK DISK$INVENT
$ RUN INVENTORY
$ LOGOUT
```

The application program INVENTORY assumes control when the user logs into the account. You should assign the CAPTIVE flag to the login flags field of the captive account UAF record by specifying the AUTHORIZE qualifier /FLAGS=CAPTIVE. The CAPTIVE flag locks the user of the captive account into the application software. Section 4.3 shows how to use AUTHORIZE to create a UAF record for a captive account.

There are a number of additional measures you can take to ensure that a captive account is secure from potential break-ins. Here are a few guidelines for setting up a command procedure for a captive account:

- To produce a highly controlled and restricted environment for the user, you must design the command procedure so that it runs in a loop until some exit condition occurs. Once the exit condition occurs, the command procedure logs the account out.

- The command procedure must handle all possible error conditions; otherwise, it might exit prematurely under some circumstances. Check the error condition handling carefully so that no error could cause it to loop indefinitely.
- Do not allow the DCL command INQUIRE to appear in any of the command procedures. Instead use the DCL command READ/PROMPT. For example, to request the user to input the date, you might enter the following command:

  ```
  READ/PROMPT="Enter date:" SYS$COMMAND DATE
  ```

- For similar reasons, avoid any use of the construction 'x, where x contains a string typed in by the user, because this requires an evaluation of the symbol. Never permit a captive command procedure to attempt an evaluation of a symbol that the user enters. Clever applications of lexical functions could break the command procedure.
- If the function of the command procedure requires text preparation, you may want to run a text editor. If you do this, design the environment of the account with extra caution. Remember that most editors are capable of reading and writing arbitrary files (within the access rights of the account); certain editors allow for the execution of arbitrary DCL commands.

Example 4–4 shows a command procedure for a highly secure captive account environment, which restricts the user to a very limited set of commands. Note that the security manager would use the AUTHORIZE qualifier /NOINTERACTIVE when establishing this account.

4.2.8 Logout Command Procedures

The system does not provide for automatic execution of a command procedure at logout time. However, you can supply one as follows:

1. Create a systemwide logout command procedure that executes whenever a user logs out (the file is usually named SYS$MANAGER:SYLOGOUT.COM).
2. To ensure that this command procedure executes, include a command in SYS$MANAGER:SYLOGIN.COM that equates the most commonly used abbreviation of the LOGOUT command (often LO) to the execution of the logout command procedure. For example:

   ```
   $ LO*GOUT:==@SYS$MANAGER:SYLOGOUT
   ```

   The last line of the logout command procedure then uses an alternate form of the LOGOUT command, such as a LOGOUTNOW command. (You can create any command name you like, beginning with LO.) You cannot use the same abbreviation as used for the symbol (in this case LO) because it will start the procedure again. As an alternative, you could make the next to the last line of the procedure:

   ```
   $ DELETE/SYMBOL/GLOBAL LOGOUT
   ```

Possible to have more than two lines on a line 2

If one allows mail to be used then there is a tricky way of editing it! Therefore do not allow any owner access to it.

Example 4–4 Example of a Captive Login Command Procedure

```
$ deassign sys$input
$ prev_sysinput == f$logical("SYS$INPUT")
$ on control_y then $goto next_cmd
$ set control=(y,t)
$next_cmd:
$ on error then $goto next_cmd
$ if prev_sysinput .nes. f$logical("SYS$INPUT") then deassign sys$input
$ read /end=next_cmd /prompt="$ " sys$command cmd
$ cmd :== 'cmd
$!
$ delete = "delete"
$ delete /symbol /local /all
$ if f$locate ("@", cmd) .ne. f$length(cmd) then goto illegal_cmd
$ if f$locate ("=", cmd) .ne. f$length(cmd) then goto illegal_cmd
$ t1 = f$locate (" ",cmd)
$ cmd1 = f$extract (0, t1, cmd)
$ if f$locate (cmd1, "LOGOUT") .eq. 0 then goto logout
$ if f$locate (cmd1, "HELP") .eq. 0 then goto help
$!
$! Place other validation checks here
$!
$ write sys$output "%CAPTIVE-W-IVVERB, unrecognized command"
$ write sys$output "   \",cmd1,"\"
$ goto next_cmd
$!
$illegal_cmd:
$ write sys$output "%CAPTIVE-W-ILLEGAL, bad characters in command"
$ goto next_cmd
$!
$cmd_ok:
$ define sys$input sys$command:
$logout:
$ logout
$ goto next_cmd
$help:
$ help
$ goto next_cmd
$!
$! Place other prevalidated commands here
$!
```

4.3 Adding a User Account with AUTHORIZE

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

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

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

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

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

- **Limits and Quotas** (/ASTLM, /BIOLM, /CPUTIME, /DIOLM, /ENQLM, /FILLM, /JTQUOTA, /MAXACCTJOBS, /MAXDETACH, /MAXJOBS, /PGFLQUOTA, /PRCLM, /SHRFILLM, /TQELM, /WSDEFAULT, /WSEXTENT, /WSQUOTA)—These qualifiers impose limits on the use of reusable system resources; the default values are adequate in most cases.

  See Chapter 5 for a discussion of limits and quotas.

- **Priority** (/PRIORITY, /QUEPRIORITY)—The default values are usually adequate for accounts not running real-time processes. See Chapter 5 for a discussion of priorities.

- **Privileges** (/DEFPRIVILEGES, /PRIVILEGES)—The default privileges (TMPMBX, NETMBX) are usually adequate, depending on the purpose of the account. See Section 5.3 for a discussion of privileges.

- **Primary and Secondary Login Times; Login Functions** (/ACCESS, /DIALUP, /FLAGS, /INTERACTIVE, /LOCAL, /PRIMEDAYS, /REMOTE)—By default, users are allowed to log in at any hour of any day. To override the setting of a particular day, use the DCL command SET DAY. Use this command if a holiday occurs on a day that would normally be treated as a primary day and you want it treated as a secondary day. See Section 4.8.4 for a discussion of using these fields to restrict login times and functions. Specify MCR as the CLI if your system is running in RSX compatibility mode.

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

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

In this example, the /FLAGS and /NOACCESS qualifiers restrict users from logging in to the captive account. The /NOACCESS qualifier limits logins to specific hours. (See Section 4.8.4 for more information on restricting logins access to certain hours.)

The /FLAGS=CAPTIVE qualifier adds the login flag CAPTIVE to the captive account record. The CAPTIVE flag locks the person using the account into the application software by doing the following:

- Disabling the CTRL/Y function to prevent users from interrupting the execution of the command procedure and gaining access to the command interpreter

- Preventing the user from specifying an alternate command interpreter with the /CLI qualifier at login time

- Preventing the user from specifying an alternate default disk device with the /DISK qualifier at login time

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

Setting Up an Individual User Account with AUTHORIZE

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

Setting Up a Captive Account with AUTHORIZE

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

4.4 Adding a User Account with a Command Procedure

As an alternative to using the Authorize Utility, you can use a command procedure to create user accounts. The ADDUSER.COM procedure, which is located in the SYS$EXAMPLES directory, is an example of such a procedure; it supplies prompts and several default values for creating the new account. To run ADDUSER.COM, log in to the SYSTEM account and enter the following command:

```
$ @SYS$EXAMPLES:ADDUSER.COM
```

ADDUSER.COM prompts you to enter values in a number of UAF record fields. If you press RETURN without specifying a value for a field, ADDUSER supplies the following default values:

UAF Field	Default Value
User name	no default;must supply
Full name	no default;must supply
Password	username specified

UAF Field	Default Value
UIC group number	200
UIC member number	must supply
Account name	optional
Privileges	TMPMBX,NETMBX
Login directory	username specified
Login device	$DISK1
Disk quota	1000
Overdraft quota	100

The member number of the UIC must be unique for the system. You can list the UICs currently assigned to users by typing a question mark (?) after the UIC member number prompt. The account is not finally created until you have answered all of the questions in the procedure. The final prompt in the procedure is the following:

```
Is everything satisfactory with the account [YES]?
```

If you press the RETURN key, the account is created and remains in SYSUAF.DAT as specified. If you type NO, the account is removed.

Note: If you type CTRL/Y before, during, or directly after the account's characteristics are displayed (that is, before you respond to the "satisfactory?" prompt), the account (or portions of it) will still be added.

The ADDUSER.COM command procedure is an example of a command procedure that you can use to add user accounts. You should modify ADDUSER.COM as appropriate for the needs of your system.

4.5 Setting Up an Automatic Login Account with ALFMAINT

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

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

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

Individual Account with Automatic Login

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

$
$ SET DEFAULT SYS$MANAGER
$
$ @ALFMAINT

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

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

Captive Account with Automatic Login

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

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

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

4.6 Setting Up a Project Account with ACL Identifiers

This section describes how to set up a project account that uses access control lists (ACLs) to control access to files shared by members of a project group. See the *Guide to VMS System Security* for complete details on setting up accounts with ACLs.

You must first add an ACL identifier to the rights database for the project account. You use the AUTHORIZE command ADD/IDENTIFIER to add identifiers to the rights database. All users who hold the project's identifier can use the resources of the project account. You associate users as holders of existing ACL identifiers with the AUTHORIZE command GRANT /IDENTIFIER.

You can set up the project account so that disk space is charged to the project, rather than to individual users, by assigning the RESOURCE attribute to the project identifier. These concepts are made clearer in the following procedure, which summarizes the steps for setting up a project account:

1. Set up individual user accounts for each member sharing the project account (as described in the previous sections on adding a user account).
2. Create the project identifier with the RESOURCE attribute and grant it to those users who will have access to the project account. In the example that follows, the project identifier KITE_FLYING is given the RESOURCE attribute. This identifier is then granted to users GEORGE and LINDORF.

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> ADD/IDENTIFIER KITE_FLYING /ATTRIBUTES=RESOURCE
{message}
UAF> GRANT/IDENTIFIER KITE_FLYING GEORGE /ATTRIBUTES=RESOURCE
{message}
UAF> GRANT/IDENTIFIER KITE_FLYING LINDORF/ATTRIBUTES=RESOURCE
{message}
```

3. Create the disk quota authorization for the project identifier. For example, the following command invokes the SYSMAN Utility and assigns the identifier KITE_FLYING 2000 blocks of disk quota with 200 blocks of overdraft:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> DISKQUOTA ADD KITE_FLYING /PERMQUOTA=2000 /OVERDRAFT=200
```

4. Create the project directory. For example, the following DCL command creates the project directory [KITE_FLYING] and establishes the identifier KITE_FLYING as the owner:

```
$ CREATE/DIRECTORY [KITE_FLYING] /OWNER=[KITE_FLYING]
```

5. Set up the necessary ACL and default ACL on the project directory. For example, the following DCL command places an ACL on the directory [KITE_FLYING] that permits any holder of the identifier KITE_FLYING to gain READ, WRITE, or EXECUTE access to the directory. It also ensures that files created in the directory receive the same *ACE* (or ACL identifier record) as a default:

```
$ SET FILE [000000]KITE_FLYING.DIR;1 /OBJECT_TYPE=FILE -
_$ /ACL=((IDENTIFIER=KITE_FLYING, ACCESS=READ+WRITE), -)
_$ IDENTIFIER=KITE_FLYING,OPTIONS=DEFAULT, -
_$ ACCESS=READ+WRITE+EXECUTE
```

Access must be granted through ACL entries, because the owner identifier of the directory and the files (KITE_FLYING) does not match the UIC of any of the project members; thus, only WORLD access is available through the UIC-based protection mask. The first ACE of the specified ACL gives all project members READ and WRITE access to the directory; the second ACE gives them READ, WRITE, and EXECUTE access for all files created in the directory.

Note that project members are not allowed to delete (or control) files created by others. However, the members each have complete access to files that they have created in the directory, because the file system supplies an additional default ACL entry that grants to the creator CONTROL access plus the access specified in the OWNER field of the UIC-based protection mask. This ACE only appears when the owner of the created file does not match the UIC of the creator.

Thus, when LINDORF creates the file [KITE_FLYING]THURSDAY.TXT, it receives the following access control list by default:

```
(IDENTIFIER=LINDORF,OPTIONS=NOPROPAGATE,
ACCESS=READ+WRITE+EXECUTE+CONTROL)
(IDENTIFIER=KITE_FLYING,ACCESS=READ+WRITE+EXECUTE)
```

4.7 Setting Up a Network Proxy Account

A *network proxy* account allows users on a remote node in a network to access data by way of a local account on your system. Proxy accounts are useful when you want to grant one or more users on a remote node access to specific files but you do not want to give them a private account on your system. You establish and control proxy accounts with the Authorize Utility.

With proxy accounts, you can authorize one or more users on a remote node to enter DCL commands that access data from a particular account on your system. Proxy accounts allow remote users to access specific local data (for example, type and print files) without having to log in to your system or use an access control string. Remote users assume the same file access rights as the local account and also receive the default privileges of the local account. The following sections explain the procedures for setting up proxy accounts.

4.7.1 Creating a Network Proxy Authorization File

A proxy account permits an authorized user from a remote node to log in to a local node, as if the user owned an account on the local node. Proxy accounts are created and maintained using the Authorize Utility. You use the AUTHORIZE command CREATE/PROXY to create and initialize the network proxy authorization file (SYS$SYSTEM:NETPROXY.DAT).

Proxy accounts must be associated with user accounts in the SYSUAF.DAT file on your local system. You will probably want to create a "standard access" account in the UAF for proxy accounts. For example, you could create an account named REMOTE_MARKET with limited privileges, which allows access to certain data files on your local system.

Suppose you have a group of users on your local system who prepare marketing reports and who rely on input from users on other systems. You would assign the REMOTE_MARKET account in SYSUAF.DAT the same group number and default privileges you assign to the local marketing group. In this way, the remote contributors could access any data files that are

"owned" by users in your marketing group and that are not protected from GROUP access.

4.7.2 Adding Proxy Accounts

You create a proxy account by adding entries to the network proxy authorization file that equate one or more users on a remote node to users on your home node. The command syntax for adding a proxy account is as follows:

```
ADD/PROXY  node::remote-user  local-user/DEFAULT  [,...]
```

You can allow remote users access to up to 16 local accounts, one default proxy account and 15 alternate proxy accounts. Use the /DEFAULT qualifier to specify the default proxy account.

For example, the following command adds a user proxy account:

```
UAF> ADD/PROXY HAL::WALTER REMOTE_MARKET/DEFAULT,PROXY2,PROXY3
```

Assume that you have created three accounts named REMOTE_MARKET, PROXY2, and PROXY3 on your home node. The entry in this example permits the user WALTER on remote node HAL to access data by way of the REMOTE_MARKET account on your home node. WALTER can access any data from his node that REMOTE_MARKET can access locally. To access data through either PROXY2 or PROXY3, WALTER must specify the desired proxy account in the access control string of the DCL command used to perform network file operations.

Note: Because the remote user receives the same privileges as the local user, you should not set up proxy accounts associated with local accounts that have special privilege. Granting remote users such access powers poses a threat to the security of your system.

Remote users can be specified by user name or, for remote systems that do not recognize the user name syntax, by UIC. The following example allows the user associated with the UIC [360,54] on remote node RSTS32 proxy access to the GENERIC account on the local node.

```
UAF> ADD/PROXY RSTS32::[360,54] GENERIC/DEFAULT
```

A number of your users may have accounts on a remote node and require ready access to their local files. You can create a network proxy authorization file record that grants access to each of them, provided that the user name on your system is the same as the user name on the remote node. The following form of the ADD/PROXY command adds such a record:

```
UAF> ADD/PROXY HAL::* */DEFAULT
```

This command authorizes any user on the remote node HAL to access any account with the same user name on your system.

Similarly, you might want to permit this sort of access for just one user.

```
UAF> ADD/PROXY HAL::BARBARA */DEFAULT
```

4.7.3 Controlling Proxy Logins

Whenever a proxy login request occurs, the system verifies that proxy access on the participating nodes is enabled. (By default, both incoming and outgoing proxy access is enabled for each node.) If access is enabled, the system checks account information in your NETPROXY.DAT. You can, however, change proxy access or disable it totally by using the Network Control Program (NCP). Use of NCP to control proxy access is described in the *VMS Networking Manual.*

4.8 Maintaining the User Environment

As the work requirements of your system change, you may need to do the following:

- Create additional default records to serve as templates for new categories of users
- Delete or disable the accounts of users who leave your site
- Impose login restrictions to limit system use by certain accounts

With the Authorize Utility, you can perform these maintenance operations by modifying or deleting records in the UAF.

4.8.1 Creating Additional Default Record Templates

On systems where all users perform the same type of work, you usually use the system-supplied default record, DEFAULT, as the template for adding new user records. You may find, however, that your system supports several different user categories, each category performing a specific type of work and requiring unique record attributes. Instead of always using the system-supplied default record as a template and making numerous changes each time you add a user record, you can create additional default UAF records to serve as templates for each user category.

Before you create additional default records, you must first decide

- What the individual user categories are
- What attributes are common to each category
- What to name the default records

Once you define a user category and establish which record attributes are needed, you can create the default record. For example, the following command creates a default record for a category of user that requires a special captive account:

```
UAF> ADD DEFAULT2/LGICMD=ALT_COM_PROC/FLAGS=CAPTIVE -
_UAF> /DEVICE=USER3:/DIRECTORY=[PRODUCT]
```

The command in this example uses the system-supplied default record DEFAULT to create the record DEFAULT2 and changes the LGICMD, login flags, default device, and default directory fields. The AUTHORIZE command COPY can then be used to create additional records having the same attributes as DEFAULT2.

The COPY command creates a new UAF record that duplicates the specified default record except where you explicitly override field values. For example, you could use the following command to create a record for a new user that duplicates the default record DEFAULT2:

```
UAF> COPY DEFAULT2 PALOOKA/PASSWORD=W7YA84MI/UIC=[360,114]
```

This example uses DEFAULT2 as a template to create a duplicate record for the user PALOOKA. Notice that the only values that are changed are those for password and UIC.

4.8.2 Deleting a User Account

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

1. Copy (or have the outgoing user of the account copy) any files of value to the ownership of another account. Be sure to change the owner UIC of the files to match the owner UIC of the new owner. You can also use the Backup Utility (BACKUP) to copy the files to a backup tape or disk.
2. Change the password and log in as a user of that account. This is to avoid inadvertently deleting files that may point to other files of different ownership. If you are working from a nonprivileged account, it eliminates this potential consequence.
3. Delete the account's files and directories from the deepest level up to the top level, using the following procedure:
   a. Locate and examine all subdirectories using the DCL command DIRECTORY [default . . .], where default is the name of the account's default directory.
   b. Delete the files in each subdirectory and then delete the subdirectory. Note that directory files are protected against owner deletion; therefore, you must change the protection before deleting directory files.
   c. Delete the account's top-level directory. Example 4–5 illustrates a command procedure that deletes an account's files from the bottom level up.

Note: The command procedure in Example 4–5 should not be executed from a privileged account.

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

Example 4–5 Command Procedure Template for Deleting an Account's Files

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

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

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

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

When you are cleaning up old files and directories, it is possible to delete a directory that still points to files. When you delete a directory file (a file with the file type DIR) without first deleting its subordinate files, the files referred to by that directory become *lost files*—files that remain on the disk but are not referenced by any directory. Lost files are a nonproductive use of disk space and act as debits against a user's disk quota.

To recover lost files, enter the Analyze/Disk_Structure Utility command:

ANALYZE/DISK_STRUCTURE/REPAIR/CONFIRM device-name:

See the *VMS Analyze/Disk_Structure Utility Manual* for a complete description of how to recover lost files.

4.8.3 Disabling a User Account

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

4.8.4 Restricting the Use of Accounts

Workload schedules often dictate the days and times your system is used to perform specific operations. Depending on the nature of the work performed at your site, you may want to control when certain users are allowed to log in. Using the Authorize Utility, you can place controls in the login characteristics fields of the UAF record to restrict the days and times a user can log in and to inhibit certain login functions. The next sections discuss some of the most common restrictions.

For a detailed description of the qualifiers used to restrict the use of accounts, see the discussion of the Authorize Utility in the *VMS Authorize Utility Manual*.

4.8.4.1 Setting Day Types

You can restrict the use of certain accounts by defining the days of the week as either PRIMARY or SECONDARY, and then assigning login restrictions to these defined day types. For example, if you define the days Saturday and Sunday as SECONDARY days, then any restrictions you assign to the SECONDARY day type apply to both.

There are two types of login restrictions you can assign to either day type:

- Time restrictions—Limit logins to specific hours of the day
- Function restrictions—Limit types of login

By default, in every user record, the five weekdays (Monday through Friday) are defined as PRIMARY days, and the two weekend days (Saturday and Sunday) are defined as SECONDARY days.

The way you define days and assign restrictions depends on your site. For example, suppose that on weekdays your system supports a large number of interactive users, but on weekends it is used for certain operations that require dedicated system resources. By assigning restrictions to the SECONDARY day type, you can restrict users from accessing the system during the days defined as SECONDARY. You can change these day type definitions for any account using the following AUTHORIZE qualifier:

```
/PRIMEDAYS=([NO]day[,...])
```

The /PRIMEDAYS qualifier uses a list of day names to define the PRIMARY and SECONDARY days of the week. To define a day as a SECONDARY day, use the prefix NO before the day name. Any days you omit from the list take their default value.

4.8.4.2 Restricting Logins to Specific Times

By default, there are no restrictions on login hours. You can specify login time restrictions using the following AUTHORIZE qualifiers:

Qualifier	Meaning
/[NO]ACCESS	Specifies access hours for all modes of logins
/[NO]DIALUP	Specifies access hours for interactive logins via dialup terminals
/[NO]INTERACTIVE	Specifies access hours for interactive logins via any terminal
/[NO]LOCAL	Specifies access hours for interactive logins via local terminals
/[NO]REMOTE	Specifies access hours for interactive logins via network remote terminals (SET HOST)

Users still logged in when the access time has expired receive the following warning message and have two minutes to log out before their process is terminated by the job controller:

```
JBC-W-RESTRICT, UAF restricts access at this time, please log out immediately
```

4.8.4.3 Restricting Login Functions

In addition to specifying hourly login restrictions, you can also assign function restrictions to an account by using appropriate keywords with the /FLAGS qualifier in the Authorize Utility. By default, there are no restrictions. Options are shown in the following table:

Keyword	Meaning
[NO]AUDIT	[Do not] audit all security-relevant actions
AUTOLOGIN	Prevent access except by automatic login when automatic logins are enabled
[NO]CAPTIVE	[Do not] prevent user from changing any defaults at login
[NO]DEFCLI	[Do not] prevent user from changing default CLI or CLI tables
[NO]DISCTLY	[Do not] disable CTRL/Y interrupts
[NO]DISNEWMAIL	[Do not] suppress "New Mail . . . " announcements
[NO]DISREPORT	[Do not] report login information (last login date, login failures, and so on)
[NO]DISUSER	[Do not] disable the account completely
[NO]DISWELCOME	[Do not] suppress "Welcome to . . . " login message
[NO]GENPWD	[Do not] require user to use generated passwords
[NO]LOCKPWD	[Do not] prevent user from changing password
[NO]DISMAIL	[Do not] prevent mail delivery to the user
[NO]PWD_EXPIRED	[Do not] mark password as expired
[NO]PWD2_EXPIRED	[Do not] mark second password as expired

4.9 UAF Login Checks

To help you understand the effect of login restrictions, this section describes how the system checks the login fields of the UAF when a user attempts to log in.

When a user activates a terminal (by turning it on and pressing RETURN if directly connected, or by dialing in to a system and observing the remote connect protocol), and that terminal is not allocated by a user process, the system prompts for a name and password. The person using the terminal must type a name and password combination that exists in a UAF record, or the system denies him further access. If the name and password are accepted, the system then performs the following operations:

1. Examines the login flags, beginning with DISUSER. If DISUSER is set, the login attempt fails. Note that setting this flag for powerful, infrequently used accounts (such as SYSTEM, SYSTEST, and FIELD) virtually eliminates the risk of guessed passwords for those accounts.
2. If the DISUSER flag is not set, verifies primary or secondary day restrictions. After checking the current day type, the system determines whether hourly login restrictions are in effect (as defined by the /ACCESS, /DIALUP, /INTERACTIVE, /LOCAL, and /REMOTE qualifiers). If the current hour is restricted, the login fails immediately. Otherwise it succeeds.
3. If the login is successful, passes control to the command interpreter (for example, DCL) named in the user's UAF record.
4. Checks whether SYS$SYLOGIN is defined. If so, the logical name is translated (in most cases to SYS$MANAGER:SYLOGIN.COM) and that procedure is executed. When the procedure completes, the system searches for the name of a login command procedure in that user's UAF record. If a command procedure is specified in the LGICMD field and that procedure exists, it is executed. Otherwise, if the LGICMD field is blank, the user's command file named LOGIN is executed automatically (if it exists).

After a successful login, the command interpreter prompts for user input (DCL usually displays a dollar sign), and the user responds with commands acceptable to the command interpreter. (DCL accepts those commands documented in the *VMS DCL Dictionary*.) However, the system prohibits activities that violate the user's privilege allowance or exceed resource quotas.

5 Controlling System Resources

This chapter contains detailed descriptions of the resource control attributes you can assign to a user process when creating a record in the UAF:

- Limits on reusable system resources
- Base priority for scheduling user processes
- Privileges allowing use of restricted and sensitive system functions

5.1 Setting Limits on Reusable System Resources

Each user of the system is limited in the consumption of such resources as system memory, volatile (pagefile) disk space, number of processes, I/O requests, and so forth. You set limits when you define the user to the system through the creation of an account in the UAF.

Limits control the way in which a process shares its allotment of a resource with the subprocesses it creates. In addition to restricting the number of processes that a single user or account may have at any given time, the system uses the following four types of limits for sharing resources:

- **Pooled**—If the limit on the use of a resource is pooled, a process and created subprocesses share the total limit on a first-come, first-served basis.
- **Deductible**—If the limit on the use of a resource is deductible, a subprocess is allotted a portion of the total limit; the portion given to the subprocess is deducted from the total limit.
- **Nondeductible**—If the limit is nondeductible, the subprocess is allotted the total limit of the creating process; there is no deduction from the allotment of the creating process.
- **Systemwide**—If the limit is systemwide, a process and all created subprocesses with the same user name or account share the total limit on a first-come, first-served basis.

In creating a UAF record, you assign values to the limits shown in Table 5–1. These limits are described in the following sections. Usually, you simply assign the default values for these limits. However, see the *Guide to VMS Performance Management* for a discussion of how to evaluate and adjust the limits in the context of performance optimization strategies.

Table 5–1 summarizes each of these limits, the value supplied in the UAF record for the SYSTEM account, and the type of limit.

Table 5–1 Process Resource Limits, Suggested Values for SYSTEM, Types, and Descriptions

Limit	Value	Type[1]	Description
ASTLM	24	N	AST queue limit
BIOLM	18	N	Buffered I/O count limit
BYTLM	32768	P	I/O byte count limit
CPU	0	D	CPU time limit (0 = no limit)
DIOLM	18	N	Direct I/O count limit
ENQLM	200	P	Enqueue quota
FILLM	40	P	Open file limit
JTQUOTA	1024	P	Initial byte quota for jobwide logical name table
MAXACCTJOBS	0	S	Maximum active processes for a single account (0 = no limit)
MAXDETACH	0	S	Maximum detached processes for a single user name (0 = no limit)
MAXJOBS	0	S	Maximum active processes for a single user name (0 = no limit)
PBYTLM	0	S	Paged pool byte count limit (0 = no limit)
PGFLQUO	20480	P	Paging file limit
PRCLM	10	P	Subprocess creation limit
SHRFILLM	0	P	Maximum number open shared files (0 = no limit)
TQELM	20	P	Timer queue entry limit
WSDEF	256	N	Default working set size
WSEXTENT	2048	N	Working set extent
WSQUO	512	N	Working set quota

[1]D=deductible, N=nondeductible, P=pooled, S=systemwide

5.1.1 AST Queue Limit (ASTLM)

The AST queue limit (ASTLM) limits the sum of the following:

- The number of asynchronous system trap (AST) requests that a user's process can have outstanding at one time
- The number of scheduled wakeup requests that a user's process can have outstanding at one time

This limit affects not only all system services that accept an AST address as an argument, but also the Schedule Wakeup ($SCHDWK) system service.

If the deferred write option (DFW) is enabled, the number of ASTs used per file is equal to 1, plus the number of record streams, plus the multibuffer count. Otherwise, the number is 1 plus the number of record streams.

ASTLM is a nondeductible limit with a suggested typical value of 24. The default values are as follows: SYSTEM account = 24, DEFAULT account = 10.

5.1.2 Buffered I/O Count Limit (BIOLM)

The buffered I/O count limit (BIOLM) limits the number of outstanding buffered I/O operations permitted a user's process.

In a buffered I/O operation, the data transfer takes place from an intermediate buffer in the system pool, not from a process-specified buffer. Buffered operations for a user process include terminal I/O, file system and network I/O, card reader input, and unspooled printer output. During a buffered I/O operation, the pages containing the process-specified buffer need not be locked in memory.

BIOLM is a nondeductible limit with a suggested typical value of 18. The default values are as follows: SYSTEM account = 18, DEFAULT account = 10.

5.1.3 Buffered I/O Byte Count Limit (BYTLM)

The buffered I/O byte count limit (BYTLM) limits the amount of buffer space that a user's process can use.

This buffer space is used for buffered I/O operations and for the creation of temporary mailboxes. It also limits the number of mapping windows the user can create as segmented (or cathedral) windows. Cathedral windows are primarily useful for reducing the overhead required to read large files.

BYTLM is a pooled limit with a suggested typical value of 32768. The default values are as follows: SYSTEM account = 32768, DEFAULT account = 8192.

5.1.4 CPU Time Limit (CPU)

The CPU time limit (CPU) limits the amount of CPU time that a user's process can use per interactive session or batch job.

The time must be specified in abbreviated delta format—hh:mm:ss:cc.

CPU is a deductible limit with a suggested typical value of 0 (no limit), but the value only applies to this instance or other instances of the user's processes. CPU is not cumulative across separate sessions or batch jobs. The default values are as follows: SYSTEM account = 0, DEFAULT account = 0.

5.1.5 Direct I/O Count Limit (DIOLM)

The direct I/O count limit (DIOLM) limits the number of outstanding direct I/O operations permitted a user's process.

In a direct I/O operation, the data transfer takes place directly from a process-specified buffer. Direct I/O operations for a user process typically include disk and tape I/O. The pages containing this buffer are locked in memory by the operating system during the direct I/O operation.

DIOLM is a nondeductible limit with a suggested typical value of 18. The default values are as follows: SYSTEM account = 18, DEFAULT account = 10.

5.1.6 Enqueue Quota (ENQLM)

The enqueue quota (ENQLM) limits the number of locks a process (and its subprocesses) can own. VAX Record Management Services (RMS) uses the Lock Management Facility to synchronize shared file access, global buffers, and record locks. Because VAX RMS takes out one lock for every shared file, local buffer, global buffer section, and outstanding record lock, users who expect to perform large amounts of VAX RMS file sharing should have ENQLM set to a large value.

If your process performs extensive VAX RMS file sharing without sufficient enqueue quota, you could receive the SS$_EXENQLM error message. Furthermore, if your system performs extensive VAX RMS file sharing and the value of the LOCKIDTBL system parameter is too low, you could receive the SS$_NOLOCKID error message. Note that whenever you increase the value of LOCKIDTBL, you may have to increase the value of the RESHASHTBL system parameter (see the discussion of the System Generation Utility (SYSGEN) in the *VMS System Generation Utility Manual*).

For shared files, the value of ENQLM should represent the number of files open as shared multiplied by the number of locks per process per file. If you use the default multibuffer counts, estimate the number of locks as 4 for indexed sequential files and 3 for relative files. If you use other than the default value for the multibuffer counts, estimate the number of locks per process per file as one per file plus the multibuffer count for that file plus the number of records locked (which is usually one). Use the DCL command SHOW RMS_DEFAULT to display the default multibuffer counts.

ENQLM is a pooled limit with a suggested typical value of 200. The default values are as follows: SYSTEM account = 200, DEFAULT account = 100.

5.1.7 Open File Limit (FILLM)

The open file limit (FILLM) limits the number of files that a user's process can have open at one time. This limit includes the number of network logical links that can be active at the same time.

FILLM is a pooled limit with a suggested typical value of 40. Note that each open file also requires at least 96 bytes of BYTLM. The default values are as follows: SYSTEM account = 40, DEFAULT account = 20.

5.1.8 Job Table Quota (JTQUOTA)

The job table quota (JTQUOTA) specifies the initial byte quota with which the jobwide logical name table is to be created.

JTQUOTA is a pooled quota with a suggested typical value of 1024. The default values are as follows: SYSTEM account = 1024, DEFAULT account = 1024.

5.1.9 Maximum Account Jobs Limit (MAXACCTJOBS)

The maximum account jobs limit (MAXACCTJOBS) specifies the maximum number of batch, interactive, and detached processes that may be active at one time for all users of a single account.

MAXACCTJOBS is a systemwide limit with a suggested typical value of 0. The default values are as follows: SYSTEM account = 0, DEFAULT account = 0.

5.1.10 Maximum Detached Processes Limit (MAXDETACH)

The maximum detached processes limit (MAXDETACH) specifies the maximum number of detached processes that may be active at one time for a single user name. Processes that exceed this limit are terminated.

MAXDETACH is a systemwide limit with a suggested typical value of 0. The default values are as follows: SYSTEM account = 0, DEFAULT account = 0.

5.1.11 Maximum Process Jobs Limit (MAXJOBS)

The maximum process jobs limit (MAXJOBS) specifies the maximum number of interactive, batch, and detached processes that can be active at one time for a user name. Processes that exceed this limit are terminated.

MAXJOBS is a systemwide limit with a suggested typical value of 0. The default values are as follows: SYSTEM account = 0, DEFAULT account = 0.

5.1.12 Paged Pool Byte Count Limit (PBYTLM)

The paged pool byte count limit (PBYTLM) specifies the size of the paged pool in bytes. The suggested typical value of PBYTLM is 0. The default values are as follows: SYSTEM account = 0, DEFAULT account = 0.

5.1.13 Paging File Limit (PGFLQUO)

The paging file limit (PGFLQUO) limits the number of pages that the user's process can use in the system paging file. The paging file provides temporary disk storage for pages forced out of memory by a memory management operation. PGFLQUO limits the total virtual address space that can be created using the Create Virtual Address Space ($CRETVA) or Expand Program/Control Region ($EXPREG) system services.

PGFLQUO is a pooled limit with a suggested typical value of 20480. The default values are as follows: SYSTEM account = 20480, DEFAULT account = 10240.

5.1.14 Subprocess Creation Limit (PRCLM)

The subprocess creation limit (PRCLM) limits the number of subprocesses a user's process can create.

The process created when a user logs in to the system can in turn create subprocesses. These subprocesses are all accountable to the user and share the resources allotted to the initial process.

PRCLM is a pooled limit with a suggested typical value of 10. The default values are as follows: SYSTEM account = 10, DEFAULT account = 2.

5.1.15 Shared Files Limit (SHRFILLM)

The shared files limit (SHRFILLM) specifies the maximum number of shared files that an account may have open at one time. The shared files limit is a pooled limit with a suggested typical value of 0. The default values are as follows: SYSTEM account = 0, DEFAULT account = 0.

5.1.16 Timer Queue Entry Limit (TQELM)

The timer queue entry limit (TQELM) limits the sum of the following:

- The number of entries that a user's process can have in the timer queue
- The number of temporary common event flag clusters that a user's process can have

This limit does not govern the creation of permanent event flag clusters.

Timer queue entries are used in time-dependent scheduling; common event flags are used in synchronizing activities among groups of cooperating processes.

TQELM is a pooled limit with a suggested typical value of 20. The default values are as follows: SYSTEM account = 20, DEFAULT account = 10.

5.1.17 Default Working Set Size (WSDEF)

The default working set size (WSDEF) sets the initial working set size limit for a user's process.

WSDEFAULT is a nondeductible limit with a suggested typical value of 256 pages. If the value specified exceeds the value of WSQUOTA, the lesser value is used. The default values are as follows: SYSTEM account = 256, DEFAULT account =150.

5.1.18 Working Set Extent (WSEXTENT)

The working set extent (WSEXTENT) specifies the maximum size to which a user's physical memory usage can grow, independent of the system load. This enlargement of the physical memory for a user is accomplished by the Adjust Working Set Limit ($ADJWSL) system service, and is normally done for the user by the VMS operating system in response to heavy page faulting by the user.

WSEXTENT is a nondeductible quota with a suggested typical value of 2048. This value should always be greater than or equal to WSQUOTA. The value is controlled by the system parameter WSMAX. The default values are as follows: SYSTEM account =2048, DEFAULT account = 512.

5.1.19 Working Set Quota (WSQUOTA)

The user's physical memory usage can grow on a typically loaded system. That is, this parameter guarantees the user that the number of physical pages specified will be available. For example, WSQUOTA limits the number of pages a user can lock in memory.

WSQUOTA is a nondeductible quota with a suggested typical value of 512. This value should be greater than or equal to WSDEFAULT. The value is controlled by the system parameter WSMAX. The default values are as follows: SYSTEM account = 512, DEFAULT account = 256.

5.2 Setting Priorities for User Processes

A user's priority is the base priority used in scheduling the process that the system creates for the user. There are 32 levels of software priority in the VMS operating system, 0 through 31. The highest priority is 31; the lowest is 0. The range of priorities for timesharing processes is 1 through 15; the range for real-time processes is 16 through 31.

Processes with real-time priorities are scheduled strictly according to base priority; in other words, the executable real-time process with the highest base priority is executed first. Processes with timesharing priorities are scheduled according to a slightly different principle, to promote overlapping of computation and I/O activities.

In the user's account record of the UAF, the default value of a user's priority is 4; for practical purposes, the minimum value is 1. You should ensure that the priority for timesharing users remains at the default. Note that if you give some users an advantage over other users by raising their priorities, ragged performance will result, because the system reacts sharply to even small base priority differences.

Never specify a value over 31 (system operation will be unpredictable).

5.3 Assigning Privileges

Privileges restrict the performance of certain system activities to certain users. These restrictions protect the integrity of the operating system's data and thus the integrity of user service. You should grant privileges to each user on the basis of two factors:

- Whether the user has the skill and experience to use the privilege without disrupting the system
- Whether the user has a legitimate need for the privilege

Privileges fall into the following seven categories according to the damage that the user possessing them could cause the system:

- None—No privileges
- Normal—Minimum privileges to use the system effectively
- Group—Potential to interfere with members of the same group
- Devour—Potential to consume noncritical systemwide resources
- System—Potential to interfere with normal system operation
- Files—Potential to compromise file security
- All—Potential to control the system

A user cannot execute an image that requires a privilege he or she does not possess, unless the image is installed as a known image with the privilege in question. (See the *VMS Install Utility Manual* for information on installing known images.) Execution of a known image with temporary privileges (for the duration of the image's execution) grants those privileges to the user process executing the image. Thus, you should install user images with amplified privileges only after ensuring that the user needs the access and is unlikely to misuse it.

A user's privileges are recorded in the user's UAF record in a 64-bit privilege mask. When a user logs in to the system, the user's privilege vector is stored in the header of the user's process. In this way, the user's privileges are passed on to the process created for the user. Users can use the DCL command SET PROCESS/PRIVILEGES to enable and disable privileges for which they are authorized, to further control the privileges available to the images they run. Moreover, any user with the SETPRV privilege can enable any privilege.

Table 5–2 lists the privileges by category and gives brief, general definitions of them. The following sections describe each privilege in detail in alphabetical order and indicate privilege categories.

Table 5–2 VMS Privileges by Category, with Definitions

Category	Privilege	Activity Permitted
None	None	None requiring privileges
Normal	MOUNT	Execute mount volume QIO
	NETMBX	Create network connections
	TMPMBX	Create temporary mailbox
Group	GROUP	Control processes in the same group
	GRPPRV	Group access via SYSTEM protection field
Devour	ACNT	Disable accounting
	ALLSPOOL	Allocate spooled devices
	BUGCHK	Make bugcheck error log entries
	EXQUOTA	Exceed disk quota
	GRPNAM	Insert group logical names in the name table
	PRMCEB	Create/delete permanent common event flag clusters
	PRMGBL	Create permanent global sections
	PRMMBX	Create permanent mailboxes
	SHMEM	Create/delete structures in shared memory
System	ALTPRI	Set base priority higher than allotment
	OPER	Perform operator functions
	PSWAPM	Change process swap mode
	SHARE	Access devices allocated to other users
	SYSLCK	Lock systemwide resources
	WORLD	Control any process
Files	DIAGNOSE	Diagnose devices
	SYSGBL	Create systemwide global sections
	VOLPRO	Override volume protection
All	BYPASS	Disregard protection
	CMEXEC	Change to executive mode
	CMKRNL	Change to kernel mode
	DETACH	Create detached processes of arbitrary UIC
	LOG_IO	Issue logical I/O requests
	PFNMAP	Map to specific physical pages
	PHY_IO	Issue physical I/O requests
	READALL	Possess read access to everything
	SECURITY	Perform security-related functions
	SETPRV	Enable any privilege
	SYSNAM	Insert/delete system logical names in the name table
	SYSPRV	Access objects via SYSTEM protection field

5.3.1 ACNT Privilege (Devour)

The ACNT privilege allows a user to create subprocesses or detached processes in which accounting is disabled. Thus, only such a privileged user can issue the DCL command RUN with the /NOACCOUNTING qualifier or inhibit accounting in the Create Process ($CREPRC) system service.

5.3.2 ALLSPOOL Privilege (Devour)

The ALLSPOOL privilege allows the user's process to allocate a spooled device by executing the Allocate Device ($ALLOC) system service. This service lets a process allocate, or reserve, a device for its exclusive use. A shareable mounted device cannot be allocated. The user may also allocate a spooled device by using the DCL command ALLOCATE.

You should grant this privilege only to users who need to perform logical or physical I/O operations to a spooled device. Ordinarily, the privilege of allocating a spooled device is granted only to special printer processes known as symbionts. (See the chapter on batch and print operations in the *Guide to Maintaining a VMS System* for information on setting up spooled devices.)

5.3.3 ALTPRI Privilege (System)

The ALTPRI privilege allows the user's process to

- Increase its own base priority
- Set the base priority of another process to a value higher than that of the target process

The base priority is increased by executing the Set Priority ($SETPRI) system service or the DCL command SET PROCESS/PRIORITY. As a rule, this system service lets a process set its own base priority or the base priority of another process. However, one process can set the priority of a second process if

- The process calling the $SETPRI system service has the same UIC as the target process
- The calling process has process control privilege (GROUP or WORLD) over the target process

With ALTPRI, a process can create a process with a priority higher than its own. It creates such a process by using an optional argument to the Create Process ($CREPRC) system service or to the DCL command RUN.

You should not grant this privilege widely; if unqualified users have the unrestricted ability to set base priorities, the fair and orderly scheduling of processes for execution can easily be disrupted.

5.3.4 BUGCHK Privilege (Devour)

Use of the BUGCHK privilege should be restricted to system software supplied by DIGITAL that uses the VMS Bugcheck Facility. This privilege allows the user process to make bugcheck error log entries.

5.3.5 BYPASS Privilege (All)

The BYPASS privilege allows the user's process read, write, execute, and delete access to all files, bypassing UIC-based and ACL-based protection.

You should grant this privilege with extreme caution, as it overrides all file protection. It should be reserved for experienced users, well-tested, reliable programs and command procedures, or the system backup operation. SYSPRV is adequate for interactive use, as it ultimately grants access to all files, while still providing access checks.

5.3.6 CMEXEC Privilege (All)

The CMEXEC privilege allows the user's process to execute the Change Mode to Executive ($CMEXEC) system service.

This system service lets a process change its access mode to executive, execute a specified routine, and then return to the access mode that was in effect before the system service was called. While in executive mode, the process is allowed to execute the Change Mode to Kernel ($CMKRNL) system service.

You should grant this privilege only to users who need to gain access to protected and sensitive data structures and internal functions of the operating system. If unqualified users have unrestricted access to sensitive data structures and functions, the operating system and service to other users can easily be disrupted. Such disruptions can include failure of the system, destruction of the database, and exposure of confidential information to unauthorized persons.

5.3.7 CMKRNL Privilege (All)

The CMKRNL privilege allows the user's process to execute the Change Mode to Kernel ($CMKRNL) system service.

This system service lets a process change its access mode to kernel, execute a specified routine, and then return to the access mode that was in effect before the system service was called.

In granting this privilege, follow the guidelines for the CMEXEC privilege.

5.3.8 DETACH Privilege (All)

The DETACH privilege allows the user's process to create detached processes by executing the Create Process ($CREPRC) system service. Detached processes remain in existence even after the user who created them has logged off the system.

An example of a detached process is the process created by the system for a user when the user logs in to the system.

There is no restriction on the UIC that can be specified for a detached process. Thus, there are no restrictions on the files and directories to which a detached process can gain access.

5.3.9 DIAGNOSE Privilege (Files)

The DIAGNOSE privilege allows the user to run online diagnostic programs and to intercept and copy all messages that are written to the error log file.

5.3.10 EXQUOTA Privilege (Devour)

The EXQUOTA privilege allows the space taken by the user's files on given disk volumes to exceed any usage quotas set for the user (as determined by UIC) on those volumes.

5.3.11 GROUP Privilege (Group)

The GROUP privilege allows the user's process to affect other processes in its own group by executing the following process control system services (and the equivalent DCL commands where applicable):

- Cancel Wakeup ($CANWAK)
- Delete Process ($DELPRC)
- Force Exit ($FORCEX)
- Resume Process ($RESUME)
- Schedule Wakeup ($SCHDWK)
- Set Priority ($SETPRI)
- Suspend Process ($SUSPND)
- Wake ($WAKE)

The user's process is also allowed to examine other processes in its own group by executing the Get Job/Process Information ($GETJPI) system service. A user with the GROUP privilege can issue the SET PROCESS command for other processes in its group.

The GROUP privilege is not needed for a process to exercise control over, or to examine, subprocesses that it created. You should grant this privilege to users who need to exercise control over each other's processes and operations.

5.3.12 GRPNAM Privilege (Devour)

The GRPNAM privilege allows the user's process to insert names in the logical name table of the group to which the process belongs and to delete names from that table, using the following logical name system services: Create Logical Name ($CRELNM) and Delete Logical Name ($DELLNM).

In addition, the privileged user can use the DCL commands ASSIGN and DEFINE to add names to the group logical name table. The DEASSIGN command deletes names from the table, and the /GROUP qualifier of the MOUNT command shares volumes among group members.

This privilege should not be granted to all users of the system because it allows a user to create an unlimited number of group logical names. When unqualified users have the unrestricted ability to create group logical names, excessive use of system dynamic memory can degrade system performance. In addition, a user with the GRPNAM privilege can interfere with the activities of other users in the same group by creating definitions of commonly used logical names, such as SYS$SYSTEM.

5.3.13 GRPPRV Privilege (Group)

The GRPPRV privilege allows a process access to files using the files' SYSTEM protection field when the process's group matches the group of the file owner. It also allows a process to change the protection of files whose owner group matches the process's group.

5.3.14 LOG_IO Privilege (All)

The LOG_IO privilege allows the user's process to execute the Queue I/O Request ($QIO) system service to perform logical-level I/O operations. This privilege is also required for certain device control functions, such as setting permanent terminal characteristics.

Usually, user I/O requests are handled indirectly by use of an I/O subsystem, such as VAX Record Management Services. However, to increase their control over I/O operations and to improve the efficiency of I/O operations, skilled users sometimes prefer to handle directly the interface between their process and a system I/O driver program. They can do this by executing the Queue I/O Request system service; in many instances, the operation called for is a logical-level I/O operation.

You should grant this privilege only to users who need it, because it allows a process to access data anywhere on the selected volume without the benefit of any file structuring. If this privilege is given to unqualified users who have no need for it, the operating system and service to other users can easily be disrupted. Such disruptions can include the destruction of information on the system device, the destruction of user data, and the exposure of confidential information to unauthorized persons.

5.3.15 MOUNT Privilege (Normal)

The MOUNT privilege allows the user's process to execute the mount volume QIO function. The use of this function should be restricted to system software supplied by DIGITAL.

5.3.16 NETMBX Privilege (Normal)

The NETMBX privilege allows the user to perform functions related to a DECnet computer network. This privilege is usually granted to general users.

5.3.17 OPER Privilege (System)

The OPER privilege allows use of the operator communication process (OPCOM), as follows:

- To reply to user requests
- To broadcast messages to all terminals logged in
- To designate terminals as operators' terminals and specify the types of messages to be displayed
- To initialize and control the operator log file

In addition, this privilege enables the user to set devices spooled, create and control both batch and output queues, and initialize and mount public volumes.

You should grant this privilege only to operators of the system. These are the users who respond to the requests of ordinary users, who tend to the needs of the system's peripheral devices (mounting reels of tape and changing printer forms), and who attend to all the other day-to-day chores of system operation. (A nonprivileged user can log in on the console terminal to respond to operator requests, for example, to mount a tape.)

5.3.18 PFNMAP Privilege (All)

The PFNMAP privilege allows the user's process to map to specific pages of physical memory or I/O device registers, no matter who is using the pages or registers.

You should exercise caution in granting this privilege. If unqualified users have unrestricted access to physical memory, the operating system and service to other users can easily be disrupted. Such disruptions can include failure of the system, destruction of the database, and exposure of confidential information to unauthorized persons.

5.3.19 PHY_IO Privilege (All)

The PHY_IO privilege allows the user's process to execute the Queue I/O Request ($QIO) system service to perform physical-level I/O operations.

Usually, users' I/O requests are handled indirectly by use of an I/O subsystem such as VAX Record Management Services. However, to increase their control over I/O operations and to improve the efficiency of their applications, skilled users sometimes prefer to handle directly the interface between their process and a system I/O driver program. They can do this by executing the Queue I/O Request system service; in many instances, the operation called for is a physical-level I/O operation.

You should grant the PHY_IO privilege only to users who need it; in fact, this privilege should be granted even more carefully than the LOG_IO privilege. If this privilege is given to unqualified users who have no need for it, the operating system and service to other users can easily be disrupted. Such disruptions can include the destruction of information on the system device, the destruction of user data, and the exposure of confidential information to unauthorized persons.

5.3.20 PRMCEB Privilege (Devour)

The PRMCEB privilege allows the user's process to create or delete a permanent common event flag cluster by executing the Associate Common Event Flag Cluster ($ASCEFC) or Delete Common Event Flag Cluster ($DLCEFC) system service. Common event flag clusters enable cooperating processes to communicate with each other, thus providing the means of synchronizing their execution.

This privilege should not be granted to all users of the system, because it allows the user to create an unlimited number of permanent common event flag clusters. A permanent cluster remains in the system even after the creating process has been terminated and continues to use up a portion of system dynamic memory. When many users have the unrestricted ability to create permanent common event flag clusters, the excessive use of system dynamic memory can degrade system performance.

5.3.21 PRMGBL Privilege (Devour)

The PRMGBL privilege allows the user's process to create global sections by executing the Create and Map Section ($CRMPSC) system service. In addition, the user with this privilege (plus the CMKRNL and SYSGBL privileges) can use the Install Utility.

Global sections are shared structures that can be mapped simultaneously in the virtual address space of many processes. All processes see the same code or data. Global sections are used for reentrant subroutines or data buffers.

You should grant this privilege with care. If permanent global sections are not explicitly deleted, they tie up space in the global section and global page tables, which are limited resources.

5.3.22 PRMMBX Privilege (Devour)

The PRMMBX privilege allows the user's process to create or delete a permanent mailbox by executing the Create Mailbox and Assign Channel ($CREMBX) system service or the Delete Mailbox ($DELMBX) system service.

Mailboxes are buffers in virtual memory that are treated as if they were record-oriented I/O devices. A mailbox is used for general interprocess communication.

The PRMMBX privilege should not be granted to all users of the system. Permanent mailboxes are not automatically deleted when the creating processes are deleted and, thus, continue to use up a portion of system dynamic memory.

5.3.23 PSWAPM Privilege (System)

The PSWAPM privilege allows the user's process to control whether it can be swapped out of the balance set by executing the Set Process Swap Mode ($SETSWM) system service. Not only must a process have this privilege to lock itself in the balance set (that is, to disable swapping), but also to unlock itself (that is, to enable swapping).

With this privilege, a process can create a process that is locked in the balance set (process swap mode disabled) by using an optional argument to the Create Process ($CREPRC) system service or, when the DCL command RUN is used to create a process, by using a qualifier of the RUN command.

You should grant this privilege only to users who need to lock a process in memory for performance reasons. Typically, this will be a real-time process. If unqualified users have the unrestricted ability to lock processes in the balance set, physical memory can be held unnecessarily, thereby degrading system performance.

5.3.24 READALL Privilege (All)

The READALL privilege allows the process to bypass existing restrictions that would otherwise prevent the process from reading a file. However, unlike the BYPASS privilege, which permits writing and deleting, READALL permits only reading of the file.

You should grant this privilege only to individuals with appropriate management responsibilities at your site.

5.3.25 SECURITY Privilege (All)

The SECURITY privilege allows a process to perform security-related functions such as enabling or disabling security audits or setting the system password.

You should grant this privilege only to individuals responsible for system security.

5.3.26 SETPRV Privilege (All)

The SETPRV privilege allows the user's process to create processes whose privileges are greater than its own, by executing the Create Process ($CREPRC) system service with an optional argument, or by issuing the DCL command RUN to create a process. A user with this privilege can also execute the DCL command SET PROCESS/PRIVILEGES to obtain any desired privilege.

You should exercise the same caution in granting the SETPRV privilege as in granting any other privilege in the ALL category, since SETPRV allows the user to enable any or all privileges.

5.3.27 SHARE Privilege (System)

The SHARE privilege allows the user's process to assign channels to devices allocated to other users.

Normally you should grant this privilege only to system software such as symbionts. This privilege would allow an irresponsible user to interfere with the operation of devices belonging to other users.

5.3.28 SHMEM Privilege (Devour)

The SHMEM privilege allows the user's process to create and delete global sections and mailboxes (permanent and temporary) in multiport memory (VAX 11/780 processors only). The process must also have appropriate PRMGBL, PRMMBX, SYSGBL, and TMPMBX privileges. Just as in local memory, the space required for a multiport memory temporary mailbox counts against the buffered I/O byte count limit (BYTLM) of the process.

5.3.29 SYSGBL Privilege (Files)

The SYSGBL privilege allows the user's process to create system global sections by executing the Create and Map Section ($CRMPSC) system service. In addition, the user with this privilege (plus the CMKRNL and PRMGBL privileges) can use the Install Utility.

You should exercise caution in granting this privilege. System global sections require space in the global section and global page tables, which are limited resources.

5.3.30 SYSLCK Privilege (System)

The SYSLCK privilege allows the user's process to lock systemwide resources with the Enqueue Lock Request ($ENQ) system service. You should grant this privilege to users who need to run programs that lock resources in the systemwide resource name space. Exercise caution in granting this privilege, since users who hold it can interfere with the synchronization of system and other users' software.

5.3.31 SYSNAM Privilege (All)

The SYSNAM privilege allows the user's process to insert names in the system logical name table and to delete names from that table by using the Create Logical Name ($CRELNM) and Delete Logical Name ($DELLNM) system services.

In addition, the user with this privilege can use the DCL commands ASSIGN and DEFINE to add names to the system logical name table, and can use the DEASSIGN command to delete names from the table.

You should grant this privilege only to the system operators or to system programmers who need to define system logical names (such as names for user devices, library directories, and the system directory). For example, to mount or dismount a system volume, which entails defining a system logical name, you must have the SYSNAM privilege. Note that a user with

the SYSNAM privilege could redefine such critical system logical names as SYS$SYSTEM and SYSUAF, thus gaining control of the system.

5.3.32 SYSPRV Privilege (All)

The SYSPRV privilege allows the user to assume the file access rights of a system user and to change the owner UIC and protection of a file. Even if a file is protected against SYSTEM access, the user with the SYSPRV privilege can simply change the file's protection to gain access to it.

You should exercise caution in granting this privilege. If unqualified users have SYSTEM access rights, the operating system and service to others can easily be disrupted. Such disruptions can include failure of the system, destruction of the database, and exposure of confidential information to unauthorized persons.

5.3.33 TMPMBX Privilege (Normal)

The TMPMBX privilege allows the user's process to create a temporary mailbox by executing the Create Mailbox and Assign Channel ($CREMBX) system service.

Mailboxes are buffers in virtual memory that are treated as if they were record-oriented I/O devices. A mailbox is used for general interprocess communication. Unlike a permanent mailbox, which must be explicitly deleted, a temporary mailbox is deleted automatically when it is no longer referenced by any process. Note that this privilege is required to use the DCL commands SUBMIT and PRINT.

You should usually grant this privilege to all users of the system to facilitate interprocess communication. System performance is not likely to be degraded by permitting the creation of temporary mailboxes, because their number is controlled by limits on the use of system dynamic memory (BYTLM quota).

5.3.34 VOLPRO Privilege (Files)

The VOLPRO privilege allows the user to perform the following tasks:

- Initialize a previously used volume with an owner UIC different from the user's own UIC
- Override the expiration date on a disk or disk volume he or she does not own
- Mount with the /FOREIGN qualifier a Files–11 volume he or she does not own
- Override the owner UIC protection of a volume

The VOLPRO privilege permits control only over volumes that the user can mount or initialize. Volumes mounted with the /SYSTEM qualifier are safe from the user with the VOLPRO privilege as long as the user does not also have the SYSNAM privilege.

You should exercise extreme caution in granting the VOLPRO privilege. If unqualified users can override volume protection, the operating system and service to others can be disrupted. Such disruptions can include destruction of the database and exposure of confidential information to unauthorized persons.

5.3.35 WORLD Privilege (System)

The WORLD privilege allows the user's process to affect other processes both inside and outside its group by executing the following process control system services (and the equivalent DCL commands where applicable):

- Cancel Wakeup ($CANWAK)
- Delete Process ($DELPRC)
- Force Exit ($FORCEX)
- Resume Process ($RESUME)
- Schedule Wakeup ($SCHDWK)
- Set Priority ($SETPRI)
- Suspend Process ($SUSPND)
- Wake ($WAKE)

The user's process is also allowed to examine processes outside its own group by executing the Get Job/Process Information ($GETJPI) system service. The user with WORLD privilege can issue the DCL commands SET QUEUE, DELETE/ENTRY, STOP/ENTRY, and SET PROCESS for all other processes.

To exercise control over subprocesses that it created or to examine these subprocesses, a process needs no special privilege. To affect or to examine other processes inside its own group, a process needs only the GROUP privilege. But to affect or examine processes outside its own group, a process needs WORLD privilege. You should normally grant this privilege to any user who needs to affect other processes on a systemwide basis.

6 Performing AUTOGEN and SYSGEN Operations

When your system is installed or upgraded, a DIGITAL-supplied command procedure (SYS$UPDATE:AUTOGEN.COM) is executed to set the values of system parameters, the sizes of page, swap, and dump files, and the contents of the default installed image list (VMSIMAGES.DAT in SYS$MANAGER). AUTOGEN makes these adjustments after evaluating your hardware configuration and estimating typical workloads.

In many cases, AUTOGEN can improve system performance by using dynamic feedback information gathered from the running system. You may want to monitor performance for a period of time after installation and then execute AUTOGEN again using the feedback mechanism to make further system adjustments. Feedback allows AUTOGEN to *size* the VMS operating system based on your actual workload. Sizing is the process of matching the amount of memory and disk space used by the VMS operating system (for example, for page and swap files) with the workload requirements of your site.

The feedback mechanism should minimize the need for explicitly modifying calculated parameter values or system file sizes. However, if you must make modifications, DIGITAL recommends that you use AUTOGEN rather than SYSGEN. In addition to invoking SYSGEN to install your explicit modifications, AUTOGEN also analyzes them and adjusts any related parameter values.

AUTOGEN operations are discussed in Sections 6.1 through 6.1.7.

There are, however, certain system modifications and other operations that require the use of SYSGEN. These operations are described in Sections 6.2 through 6.2.5.

6.1 AUTOGEN Functions

AUTOGEN performs some or all of the following operations (depending on which phases of AUTOGEN are executed):

- Collects the following types of data:
  - Feedback data (on the running system)
  - Hardware configuration characteristics
  - User-supplied SYSGEN parameter information
  - DIGITAL-supplied SYSGEN parameter information
- Calculates appropriate new values for significant SYSGEN parameters (listed in Table 6–2)
- Creates a new installed image list
- Calculates the sizes of system page, swap, and dump files

- Resets system parameter values and file sizes if necessary
- Optionally shuts down and reboots the system

Carefully examine the results of calculations that AUTOGEN makes during a new installation or upgrade procedure to determine whether AUTOGEN has drawn the correct conclusions about your hardware configuration and to be sure the system parameter values shown are appropriate for your workload requirements.

To modify system parameter values or page and swap file sizes to accommodate special requirements at your site, invoke AUTOGEN as described in the next section and follow the procedures in Sections 6.1.5 and 6.1.6.

6.1.1 When to Use AUTOGEN

You invoke AUTOGEN when you install the system software and when you want to reset system parameter values and system file sizes. The new values and file sizes take effect the next time the system is bootstrapped. DIGITAL recommends running AUTOGEN in the following circumstances:

- During a new installation or upgrade
- Whenever your workload changes significantly
- When you add an optional (layered) software product
- When you install images that are shared

Certain layered products may require executing AUTOGEN to adjust parameter values and page and swap file sizes. See the specific product documentation for installation requirements.

You should execute AUTOGEN if you install many images with the /SHARED attribute. The GBLSECTIONS and GBLPAGES parameters in MODPARAMS.DAT may need to be increased to accommodate the additional global pages and global sections consumed.

6.1.2 How to Invoke AUTOGEN

To invoke AUTOGEN, log in to the system manager's account and use the following command syntax at the DCL prompt:

```
@SYS$UPDATE:AUTOGEN [start-phase] [end-phase] [execution-mode]
```

You can enter up to three parameters to perform the desired AUTOGEN operation. All parameters are optional; however, if you do not supply an option in parameter 1, you must enter a null argument (that is, ""). For online help information about AUTOGEN, enter the word HELP in parameter 1.

AUTOGEN executes all of the phases in sequence, starting with the specified start-phase through to the specified end-phase. The start phase must either precede or be identical to the end phase, according to the phase sequence shown in Table 6–1.

If you do not specify a start phase in parameter 1, GENPARAMS is the default start phase (see the parameter list that follows). If you do not specify an end phase in parameter 2, the end phase is given the same value as the start phase, by default. The AUTOGEN parameter options and their functions are listed as follows:

PHASE PARAMETER OPTIONS (P1 and P2)

SAVPARAMS
Saves dynamic feedback data from the running system.

GETDATA
Collects all data to be used in AUTOGEN calculations.

GENPARAMS
Generates new system parameters; creates the installed image list.

TESTFILES
Displays the system page, swap, and dump file sizes calculated by AUTOGEN (cannot be used as a start phase).

GENFILES
Generates new system page, swap, and dump files if appropriate (cannot be used as a start phase).

SETPARAMS
Runs SYSGEN to set the new system parameters in VAXVMSSYS.PAR and generate a new parameter file, AUTOGEN.PAR. The original parameters are saved in the file, VAXVMSSYS.OLD.

SHUTDOWN
Prepares the system to await a manual reboot.

REBOOT
Automatically shuts down and reboots the system.

EXECUTION MODE PARAMETER OPTIONS (P3)

NOFEEDBACK
Specifies that feedback data should not be used in AUTOGEN's calculations. The SAVPARAMS phase is not executed. Use NOFEEDBACK mode for the initial system installation or upgrade. NOFEEDBACK is synonymous with INITIAL.

FEEDBACK
Specifies that AUTOGEN should use dynamic feedback data to make its calculations.

Note: The execution mode parameter is effective only if you specify SAVPARAMS or GETDATA as the start phase.

You use the NOFEEDBACK execution mode during a new installation. In FEEDBACK mode, AUTOGEN uses dynamic feedback data from the running system to make its calculations. Feedback data is only useful when the system has been running long enough to accurately reflect your normal workload. Section 6.1.3 describes AUTOGEN's feedback mechanism in detail.

If you do NOT specify an execution mode, by default, feedback information is used in the calculations. However, if AUTOGEN suspects that the feedback information may not be accurate, it makes the calculations, issues the feedback report, and then stops prior to modifying any parameters or system files. This is the AUTOGEN default behavior even if you specified SETPARAMS, SHUTDOWN, or REBOOT as the end phase parameter. See Section 6.1.4 for more information on the AUTOGEN phases.

6.1.3 AUTOGEN Feedback

The VMS Executive maintains information about how various resources are utilized by the user's workload. The AUTOGEN feedback mechanism sizes a VMS operating system by using this information in its calculations. AUTOGEN does not require a separate monitoring process, because the Executive maintains the data needed for feedback.

The feedback mechanism affects the following resources:

- Nonpaged pool
- Lookaside lists
- Paged pool
- Lock resources
- Number of processes
- Global pages
- Global sections
- File system caches
- Page files
- Swap files

This data is gathered during AUTOGEN's SAVPARAMS phase and is written to the file SYS$SYSTEM:AGEN$FEEDBACK.DAT. This file is then read during the GETDATA phase. (See Section 6.1.4 for more information on AUTOGEN phases.)

Whenever you modify the system (for example, a hardware upgrade, a change in the number of users, an optional product installation), you should operate in the new system environment for a period of time, and then execute AUTOGEN again starting from the SAVPARAMS phase. AUTOGEN performs some basic checks on the feedback data and issues a warning message for any of the following conditions:

- The system has been up for less than 24 hours.
- The feedback information is over 30 days old.
- The feedback information was collected on another system, and the file was copied to the current system.

You must decide if you want to use the resulting SYSGEN parameter values and system file sizes calculated by AUTOGEN. To help in your decision making, AUTOGEN generates a report that includes the following information:

- All parameters and system files directly affected by the feedback information
- Current values
- New values
- The feedback data used in each parameter calculation
- Any user- or DIGITAL-supplied modifications found in MODPARAMS.DAT or VMSPARAMS.DAT.

The report is written to SYS$SYSTEM:AGEN$FEEDBACK.REPORT. The following example shows the contents of an AUTOGEN feedback report:

Example 6–1 A Sample AUTOGEN Feedback Report

```
20-FEB-1988 16:50          AUTOGEN FEEDBACK REPORT
                       ON NODE LITTLE, SID = %X0484F00B
   *WARNING*    Feedback information was collected on node MONSTR
                with SID = %X047D9D0B
Feedback information was collected on 18-JAN-1988 12:53:32.
   *WARNING*    Feedback information is over 30 days old.
Feedback information is based on 20 hours of up time.
   *WARNING*    The system was up for less than 24 hours
                when the feedback information was recorded.

Parameter: MAXPROCESSCNT       Current value: 130      New value: 110
        Relevant feedback information:
           Maximum observed processes: 66

Parameter: BALSETCNT           Current value: 101      New value: 80
        Relevant feedback information:
           Maximum observed processes: 66

Parameter: NPAGEDYN            Current value: 550400   New value: 1073952
        Relevant feedback information:
           Maximum observed non-paged pool size: 1115136 bytes.
           Non-paged pool request rate: 18 requests per 10 sec.

        User or Digital-supplied parameter modifications / overrides:
           100000 has been added to NPAGEDYN.

Parameter: PAGEDYN             Current value: 553472   New value: 642083
        Relevant feedback information:
           Current paged pool usage: 550312 bytes.
           Paged pool request rate: 5 requests per 10 sec.

Parameter: SRPCOUNT            Current value: 1606     New value: 3965
        Relevant feedback information:
           Maximum observed SRP list size: 6325
           Number of failed SRP allocation attempts: 108

Parameter: IRPCOUNT            Current value: 920      New value: 1306
        Relevant feedback information:
           Maximum observed IRP list size: 1693
```

Example 6–1 Cont'd. on next page

Example 6–1 (Cont.) A Sample AUTOGEN Feedback Report

```
Parameter: LOCKIDTBL          Current value: 1103     New value: 1381
        Relevant feedback information:
           Current number of locks: 1720
        Relevant feedback information:
           Current number of resources: 1180

Parameter: GBLPAGES           Current value: 16384    New value: 12000
        Relevant feedback information:
           Current used GBLPAGES: 11050

        User or Digital-supplied parameter modifications / overrides:
           The calculation has been disabled by a hard-coded value of 12000.

20-FEB-1988 16:50          AUTOGEN FEEDBACK REPORT

Swap file calculations have been overridden by a user-supplied value of 0.

File name: DISK$VMS02APR:[SYS2.SYSEXE]PAGEFILE.SYS
        Current size: 8192            New size: 8192
        Maximum observed usage: 4109

File name: DISK$NPAGED$:[PAGESWAP]PAGEFILE_LITTLE.SYS
        Current size: 80000           New size: 102100
        Maximum observed usage: 51037

File name: DISK$VMS02APR:[SYS2.SYSEXE]SWAPFILE.SYS
        Current size: 4096            New size: 4096
        Maximum observed usage: 3936

File name: DISK$NPAGED$:[PAGESWAP]SWAPFILE_LITTLE.SYS
        Current size: 60000           New size: 60000
        Maximum observed usage: 57498
```

6.1.4 AUTOGEN Phases

This section describes the AUTOGEN phases in the order in which they are executed. Table 6–1 lists the AUTOGEN phases in sequence, along with the files needed as input and the files generated or changed for output. All files except VMSIMAGES.DAT (which contains the installed image list) reside in the SYS$SYSTEM directory. VMSIMAGES.DAT resides in the SYS$MANAGER directory.

Table 6–1 AUTOGEN Phases

AUTOGEN Phase	Input Files	Output Files
SAVPARAMS	None	AGEN$FEEDBACK.DAT
GETDATA	MODPARAMS.DAT VMSPARAMS.DAT AGEN$FEEDBACK.DAT	PARAMS.DAT
GENPARAMS	PARAMS.DAT	SETPARAMS.DAT VMSIMAGES.DAT AGEN$FEEDBACK.REPORT
TESTFILES	PARAMS.DAT	SYS$OUTPUT

Table 6–1 (Cont.) AUTOGEN Phases

AUTOGEN Phase	Input Files	Output Files
GENFILES	PARAMS.DAT	PAGEFILE.SYS SWAPFILE.SYS (and secondary page and swap files) SYSDUMP.DMP AGEN$FEEDBACK.REPORT
SETPARAMS	SETPARAMS.DAT	VAXVMSSYS.PAR AUTOGEN.PAR VAXVMSSYS.OLD
SHUTDOWN	None	None
REBOOT	None	None

The events that take place in each AUTOGEN phase are described as follows:

- **SAVPARAMS**

  Running the SAVPARAMS phase of AUTOGEN records feedback information in AGEN$FEEDBACK.DAT, which is used in subsequent AUTOGEN phases. If you specify NOFEEDBACK as the execution mode, this phase is not executed.

Note: You can specify the SAVE_FEEDBACK option during an interactive orderly shutdown with SYS$SYSTEM:SHUTDOWN.COM. Entering this option in response to the prompt "Shutdown options:" records feedback data collected since the system was last booted. The SAVE_FEEDBACK option causes a new version of SYS$SYSTEM:AGEN$FEEDBACK.DAT to be created. To use this new version of the feedback data, execute AUTOGEN from the GETDATA phase after the system reboots.

- **GETDATA**

  The GETDATA phase collects all of the information required for AUTOGEN calculations and places it in the file PARAMS.DAT. The following information is collected:

  — Hardware configuration data

  — DIGITAL-supplied data from VMSPARAMS.DAT

  — Feedback data from AGEN$FEEDBACK.DAT (if feedback is being used)

  — User-supplied data from MODPARAMS.DAT

- **GENPARAMS**

  In the GENPARAMS phase, AUTOGEN calculates the SYSGEN parameter values based on data stored in PARAMS.DAT and produces SETPARAMS.DAT as output. AUTOGEN checks to see if feedback information is included, and if so, uses it in the calculations. Also during this phase, AUTOGEN generates the known image file list (VMSIMAGES.DAT).

- **TESTFILES**

  The TESTFILES phase displays system page, swap, and dump file sizes generated by AUTOGEN. File sizes for all currently installed primary and secondary page and swap files are displayed. The information is directed to SYS$OUTPUT by default. You can specify TESTFILES as an end-phase parameter only. Specify TESTFILES to display the AUTOGEN file size calculations, or GENFILES to generate the new file sizes; however, you cannot specify both of these parameters on the same command line. It is recommended that you use TESTFILES to display the file size changes before actually generating the changes on your system.

- **GENFILES**

  The GENFILES phase uses the information stored in PARAMS.DAT to actually generate the new page, swap, and dump files on the system. Unlike TESTFILES, it does not give you an opportunity to display AUTOGEN's calculations and decide whether you want to use them. You can specify GENFILES as an end-phase parameter only.

  The GENFILES phase does not modify a file if the calculated size change is within ten percent of the existing file size. The following files are affected: PAGEFILE.SYS, SWAPFILE.SYS, SYSDUMP.DMP, and all currently installed secondary page and swap files. (See Section 6.1.6 for more information).

- **SETPARAMS**

  The SETPARAMS phase uses as its input the SETPARAMS.DAT file created during the GENPARAMS phase. SYSGEN is run to update the system parameter values in SYS$SYSTEM:VAXVMSSYS.PAR. AUTOGEN saves the current system parameters in SYS$SYSTEM:VAXVMSSYS.OLD before updating these parameters in SYS$SYSTEM:VAXVMSSYS.PAR. The new values are also saved in SYS$SYSTEM:AUTOGEN.PAR.

- **SHUTDOWN and REBOOT**

  To install the new system parameter values generated in the SETPARAMS phase, you specify either SHUTDOWN or REBOOT in the command line. REBOOT automatically shuts down and reboots the system, thus installing the new parameter values. SHUTDOWN shuts down the system and awaits a manual reboot. You can define the logical name AGEN$SHUTDOWN_TIME (using the DCL command DEFINE) to specify the number of minutes before shutdown occurs.

6.1.5 Using AUTOGEN to Modify System Parameters

If, after examining the AGEN$FEEDBACK.REPORT or SETPARAMS.DAT file, you decide to correct hardware configuration data or modify system parameter values, you should edit the MODPARAMS.DAT file by using the procedures described in this section. Do not edit PARAMS.DAT.

Note: **AUTOGEN enhancements have eliminated the need for OLDSITE*.DAT files. Before VMS Version 4.6, AUTOGEN used these files to propagate to PARAMS.DAT any parameter settings that AUTOGEN did not calculate. DIGITAL recommends that you review your most recent version of PARAMS.DAT. If you have OLDSITE* files that are propagating parameter settings that AUTOGEN does not calculate, add records for these parameters to MODPARAMS.DAT.**

The recommended method of modifying system parameters is to execute AUTOGEN in two passes, as follows:

1 **First pass** —Execute AUTOGEN using the following command:

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS GENPARAMS
```

This command instructs AUTOGEN to do the following:

- Save the current feedback information
- Gather all of the information required for the calculations
- Calculate the system parameter values
- Generate the feedback report
- Write the information to SETPARAMS.DAT

You may then review the input to the calculations (PARAMS.DAT), the output from the calculations (SETPARAMS.DAT), and the report generated (AGEN$FEEDBACK.REPORT).

To modify a system parameter value, edit MODPARAMS.DAT and reexecute AUTOGEN from the SAVPARAMS phase. If you are satisfied with the contents of SETPARAMS.DAT, go on to the second pass described in the next step.

2 **Second pass** —Execute AUTOGEN a second time using the following command:

```
$ @SYS$UPDATE:AUTOGEN SETPARAMS REBOOT
```

This AUTOGEN command runs SYSGEN to update the new system parameter values and installs them on the system when it is rebooted. Note that the system files are not modified using this method.

There are three ways to modify a parameter value in MODPARAMS.DAT depending on the type of modification you want to perform.

- **Explicitly adding a new parameter name**

  Use this method to specify a value for a parameter that AUTOGEN does not calculate, or to override an existing AUTOGEN calculation. (See Table 6–2 for a list of the system parameters modified in AUTOGEN calculations.) You add a DCL assignment statement to MODPARAMS.DAT in the following format:

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

  For example, the following command assigns the node name BIGVAX to the SCSNODE parameter:

  ```
  SCSNODE = "BIGVAX"    ! the node name
  ```

Note: DIGITAL strongly recommends that you use this method only for parameters that AUTOGEN does not calculate. For those that it does calculate, this method disables the calculations. The preferred method is to use the ADD_ prefix described in the following paragraph.

- **Using the ADD_ prefix**

  Use the ADD_ prefix to increment the value of any NUMERIC parameter. The new values are updated in subsequent AUTOGEN calculations during the GENPARAMS phase. The following example demonstrates the use of the ADD_ prefix:

  ```
  ADD_GBLPAGES=500
  ADD_NPAGEDYN=10000
  ```

  The ADD_ parameter value can be negative in order to lower AUTOGEN's calculated value by the specified amount (for example, -10). An ADD_ parameter record for a parameter that AUTOGEN calculates (see Table 6–2) will add the value to AUTOGEN's calculations. An ADD_ parameter record for a parameter that AUTOGEN does not calculate will add the value to the parameter's default (not current) value.

  Typically, you would not use the ADD_ prefix for modifying parameters that are calculated by the feedback mechanism, because the feedback results should accurately reflect your workload. You can still use the ADD_ prefix even with feedback; however, be aware that the ADD_ record value will accumulate from one AUTOGEN run to the next.

- **Using the MIN_ prefix**

  Use the MIN_ prefix if you do not want AUTOGEN to set a parameter below a specified value. MIN_ refers to the minimum value to which a parameter can be set by AUTOGEN.

  ```
  MIN_PAGEDYN = 400000
  ```

Note: In all three of the preceding methods of modifying system parameters, the parameter name portion of the symbol assignment added to MODPARAMS.DAT must exactly match the name of the parameter you are attempting to modify. Misspelled and abbreviated symbols are ignored with no warning.

The following example shows the contents of a sample MODPARAMS.DAT file:

```
!
! ***************** A Sample MODPARAMS.DAT for Node LITTLE ***************
!
! MODPARAMS.DAT for "LITTLE"
! REVISED: 09/13/86 -CHG- Upped GBLPAGES to account for ADA.
!
SCSNODE         = "LITTLE"      ! This is not calculated by AUTOGEN.
SCSSYSTEMID     = 19577         ! Ditto.
TTY_DEFCHAR2    = %XOD34        ! Ditto.
ADD_ACP_DIRCACHE= 150           ! Hit rate was only 65% on directory cache.
MIN_PAGEDYN     = 500000        ! PAGEDYN must be at least 1/2 Mbyte to
                                ! account for a large number of logical names.
!
PAGEFILE1_SIZE  = 0             ! Do not modify primary.
SWAPFILE        = 0             ! Skip swap file calculations altogether.
DUMPFILE        = 0             ! Ditto.
ADD_GBLPAGES    = 425+507+157   ! Account for CMS, BLISS32 and ADA.
ADD_GBLSECTIONS = 4 + 5 + 2     ! Ditto.
VIRTUALPAGECNT  = 144264        ! So that we can read MONSTR's 68Mb dumps.
!
! end of MODPARAMS.DAT for LITTLE
```

Table 6–2 System Parameters Modified in AUTOGEN Calculations

ACP_DINDXCACHE	ACP_DIRCACHE	ACP_HDRCACHE
ACP_MAPCACHE	ACP_MULTIPLE	ACP_QUOCACHE
ACP_SWAPFLGS	ACP_SYSACC	BALSETCNT
BORROWLIM	CTLPAGES	EXPECTED_VOTES
FREEGOAL	FREELIM	GBLPAGES
GBLPAGFIL	GBLSECTIONS	GROWLIM
IRPCOUNT	IRPCOUNTV	KFILSTCNT
LOCKIDTBL	LOCKIDTBL_MAX	LONGWAIT
LRPCOUNT	LRPMIN	LRPSIZE
LRPCOUNTV	MAXPROCESSCNT	MPW_HILIMIT
MPW_LOLIMIT	MPW_LOWAITLIMIT	MPW_WAITLIMIT
MSCP_LOAD	NPAGEDYN	NPAGEVIR
PAGEDYN	PFCDEFAULT	PFRATL
PHYSICALPAGES	PIXSCAN	PQL_DWDEFAULT
PQL_DWEXTENT	PQL_DWQUOTA	PQL_MWDEFAULT
PQL_MWEXTENT	PQL_MWQUOTA	RESHASHTBL
SCSCONNCNT	SPTREQ	SRPCOUNT
SRPCOUNTV	LAST	SRPSIZE
SYSMWCNT	VAXCLUSTER	VIRTUALPAGECNT
VOTES	WSMAX	WS_OPA0

6.1.6 Using AUTOGEN to Modify System File Sizes

You can use AUTOGEN to modify the sizes of page, swap, and system dump files. This section describes how AUTOGEN handles system files and gives the procedure for modifying file sizes.

DIGITAL recommends that you execute AUTOGEN in two passes to update and install any modifications to system file sizes:

1 **First pass** —Enter the following command to instruct AUTOGEN to display its calculations of system file sizes in SYS$OUTPUT:

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS TESTFILES
```

If you are satisfied with the information displayed by TESTFILES, go on to perform the second pass. If not, follow the instructions for modifying system file sizes described later in this section and start again from the first pass.

2 **Second Pass** —Execute AUTOGEN a second time using the following command, which installs the new system file sizes when the system is rebooted.

```
$ @SYS$UPDATE:AUTOGEN GENPARAMS REBOOT
```

Generally, if secondary page or swap files exist, AUTOGEN's file manipulation involves secondary files but excludes primary files; AUTOGEN assumes that primary files are on a cluster-common system disk. The following list describes how AUTOGEN handles different types of input, and how to edit MODPARAMS.DAT to modify system file sizes:

1. If you do not supply system file size information in MODPARAMS.DAT, AUTOGEN performs default page and swap file size calculations. If no secondary files exist, AUTOGEN applies any changes to the primary files. If secondary files exist, AUTOGEN applies changes evenly across all secondary page or swap files, but does not modify primary files.

2. You can supply AUTOGEN with general system file size information by adding records to MODPARAMS.DAT in the format:

   ```
   PAGEFILE = n
   SWAPFILE = n
   DUMPFILE = n
   ```

   If *n* is zero, the corresponding section is skipped. If *n* is not zero, and no secondary files exist, AUTOGEN applies the value to primary files. If *n* is not zero, and secondary files exist, AUTOGEN applies any change evenly across all secondary files, but does not modify primary files. The PAGEFILE = n symbol defines the total amount of paging space that you want in all page files.

   For example, assume that your current system page file sizes are as follows:

   ```
   Primary = 10000
   Secondary1 = 30000
   Secondary2 = 30000
   ```

   If you specified PAGEFILE = 100000, the resulting page file sizes would be as follows:

   ```
   Primary = 10000
   Secondary1 = 45000
   Secondary2 = 45000
   ```

   The DUMPFILE = n symbol is the only one that applies to dump files, because there is only one dump file to modify. Symbols described in the following paragraphs apply only to page and swap files.

3. You can specify the explicit sizes of individual page and swap files (including secondary files), as well as the location and size of new files that you want AUTOGEN to create. To specify explicit file sizes, define symbols in MODPARAMS.DAT using the following format:

   ```
   {PAGE/SWAP}FILEn_{NAME/SIZE}
   ```

   where *n* is an integer that specifies the page or swap file. Refer to the primary page and swap files by specifying a value of 1 for *n*; refer to subsequent files by specifying increasingly higher integer values for *n*. For example, to refer to a secondary page or swap file, you could specify a value of 2 for *n*. Braces ({}) indicate that you must choose between the options delimited by a backslash (/). For example, specify PAGE or SWAP, NAME or SIZE.

Note: You cannot specify both general and explicit information, because AUTOGEN issues a warning if conflicting symbol definitions exist in MODPARAMS.DAT.

For existing files, you typically define _SIZE symbols only; AUTOGEN already has the name and location. For example, to direct AUTOGEN to set the primary page file size to 10000 blocks, you would use the following symbol definition:

```
PAGEFILE1_SIZE = 10000
```

Use the following symbol definitions to direct AUTOGEN to create a new secondary swap file named PAGED$:[PAGESWAP]SWAPFILE.SYS that holds 30000 blocks.

```
SWAPFILE2_NAME = "PAGED$:[PAGESWAP]SWAPFILE.SYS"
SWAPFILE2_SIZE = 30000
```

The file sizes and parameter values specified in MODPARAMS.DAT are copied into PARAMS.DAT during the next GETDATA phase, and AUTOGEN makes appropriate adjustments in its calculations.

If the creation or extension of a file would cause the target disk to become more that 95% full, AUTOGEN issues a warning and does not perform the operation. To install a new secondary file, you must edit the file SYS$MANAGER:SYPAGSWPFILES.COM and include the appropriate SYSGEN commands. See Chapter 2 for more information on adding commands to the SYPAGSWPFILES.COM file.

You can use AUTOGEN to create a page, swap, or dump file that is smaller than the current version of the file. After you have booted and begun using the new file, remember to use the DCL command PURGE to reclaim the disk space from the old version of the file. To determine the current sizes of page and swap files, enter the DCL command SHOW MEMORY/FILE.

Note: AUTOGEN will not change file sizes if you specify a value of 0, or a value that is within ten percent of the current size.

6.1.7 Specifying an Alternate Startup Command Procedure in MODPARAMS.DAT

DIGITAL recommends that you do not modify the STARTUP.COM file supplied in the software distribution kit. Usually, site-specific modifications are added to SYS$MANAGER:SYSTARTUP_V5.COM. However, if you must specify an alternate site-independent startup command procedure, this section describes the procedure for doing so. (See Chapter 2 for more information on startup command procedures.)

If you require a startup command procedure other than SYS$SYSTEM:STARTUP.COM, you can assign the name of your alternate procedure to the symbol STARTUP in MODPARAMS.DAT. After invoking AUTOGEN, your procedure becomes the default startup command procedure. (See Section 6.2.5 for more information on setting up an alternate startup procedure.)

For example, to specify MY_STARTUP.COM as the new default startup command procedure, make the following entry in MODPARAMS.DAT:

```
STARTUP = "SYS$SYSTEM:MY_STARTUP.COM"
```

6.2 SYSGEN Functions

You use SYSGEN to manipulate the following parts of the VMS operating system:

- System parameters—Create or modify a standard system parameter file for use in subsequent bootstrap operations; dynamically modify the parameter values of the active system (applies only to the dynamic system parameters)
- Devices and device drivers—Connect devices and load their device drivers (most of this work is automatic)
- System files—Create additional page and swap files
- Startup command procedure—Designate an alternate startup command procedure (see Section 6.2.5)

See the *VMS System Generation Utility Manual* for descriptions of SYSGEN commands.

6.2.1 Using SYSGEN to Modify System Parameters

The bootstrap process initializes the active system parameter values in memory from the current system parameter file on disk (that is, the starting parameter values are those in SYS$SYSTEM:VAXVMSSYS.PAR). In a conversational bootstrap operation, you can modify these values by reinitializing the active parameter values from a parameter file or the default list, and by setting new parameter values on an individual basis. At the end of the bootstrap operation, the system parameter file is modified to conform to the active parameter values.

Caution: Many of the system generation parameters can affect other parameters or the performance of the system. The recommended method of modifying system parameters is to edit the file SYS$SYSTEM:MODPARAMS.DAT and invoke AUTOGEN.

The SYSGEN procedure for creating and modifying a system parameter file is summarized as follows:

1. Run SYSGEN—to initialize a *work area* using the active parameter values.
2. Optionally enter a USE command—to reinitialize the work area to use the values of a new parameter file, the current system parameter file, or the default values, if the active values do not provide a suitable base for subsequent operations.
3. Enter SET commands—to modify parameters on an individual basis. These modifications have no effect outside the SYSGEN work area.
4. Enter a WRITE command—to create a parameter file, modify the current system parameter file on disk, or modify the active system in memory (dynamic parameters only).

During these operations, use the SHOW command to examine the parameter values in the SYSGEN work area.

6.2.1.1 Creating a New Parameter File

The creation of a new parameter file does not affect the system. During a subsequent conversational bootstrap operation, however, you can initialize the active system with the values of the new file. The following example creates a new version of the AUTOGEN.PAR system parameter file with a new value for the REALTIME_SPTS parameter:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN SYSGEN
SYSGEN>  USE AUTOGEN
SYSGEN>  SET REALTIME_SPTS 10
SYSGEN>  WRITE AUTOGEN
SYSGEN>  EXIT
```

The next example creates a user file named SYS$SYSTEM:OURSITE.PAR, using the AUTOGEN.PAR file as a base:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN SYSGEN
SYSGEN>  USE AUTOGEN
SYSGEN>  SET REALTIME_SPTS 8
SYSGEN>  WRITE OURSITE
SYSGEN>  EXIT
```

6.2.1.2 Modifying the System Parameter File

Modification of the current system parameter file also does not immediately affect the system. During subsequent bootstrap operations, however, the active system is initialized with the new values. A conversational bootstrap operation permits you to modify these values further, while a nonstop bootstrap operation makes the new values the values of the active system. The following example modifies the REALTIME_SPTS parameter value in the system parameter file:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  USE CURRENT
SYSGEN>  SET REALTIME_SPTS 10
SYSGEN>  WRITE CURRENT
%OPCOM, 15-APR-1988 16:04:06.30, message from user SYSTEM
%SYSGEN-I-WRITECUR, CURRENT system parameters modified by process
ID 00160030 into file VAXVMSSYS.PAR
SYSGEN>  EXIT
```

6.2.1.3 Modifying the Active System

Modification of the active system immediately affects that subset of the system parameters called the dynamic parameters by changing their values in memory. The discussion of the System Generation Utility in the *VMS System Generation Utility Manual* identifies the dynamic parameters (as does the SYSGEN command SHOW/DYNAMIC). The other parameters regulate structures that cannot be changed while the system is running. The following example illustrates how to modify the active value of the PFCDEFAULT parameter:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  SET PFCDEFAULT 127
SYSGEN>  WRITE ACTIVE
%OPCOM, 15-APR-1988 16:04:06.30, message from user SYSTEM
%SYSGEN-I-WRITEACT, ACTIVE system parameters modified by process
ID 00160030
SYSGEN>  EXIT
```

Modification of the active system does not affect the current system parameter file on disk. If, for example, you set new active parameter values (WRITE ACTIVE) and later want to use them for subsequent bootstrap operations, the values must be explicitly written to the current system parameter file on disk:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  WRITE CURRENT
%OPCOM, 15-APR-1988 16:04:06.30, message from user SYSTEM
%SYSGEN-I-WRITECUR, CURRENT system parameters modified by process
ID 00160030 into file VAXVMSSYS.PAR
SYSGEN>  EXIT
```

6.2.2 Using SYSGEN to Modify System File Sizes

When your system is installed or upgraded, AUTOGEN defines sizes for the existing page and swap files appropriate for your hardware configuration, and for the system dump file. The full file specification of each file is SYS$SYSTEM:filename.type. The file names and types are PAGEFILE.SYS for the page file, SWAPFILE.SYS for the swap file, and SYSDUMP.DMP for the dump file. Sizes are expressed in blocks.

AUTOGEN creates system files suitable for most systems. For special workloads or variant configurations, however, you must specify the file sizes with the SYSGEN command CREATE. (Or, to create primary files only, you can invoke a DIGITAL command procedure called SYS$UPDATE:SWAPFILES.COM and enter file sizes when the procedure prompts you for them.) The following example illustrates the use of the CREATE command (assume that smaller versions of each file already exist):

```
SYSGEN>  CREATE PAGEFILE.SYS /SIZE=16384
%SYSGEN-I-EXTENDED, SYS$SYSROOT:[SYSEXE]PAGEFILE.SYS;1 extended
SYSGEN>  CREATE SWAPFILE.SYS /SIZE=7168
%SYSGEN-I-EXTENDED, SYS$SYSROOT:[SYSEXE]SWAPFILE.SYS;1 extended
SYSGEN>  CREATE SYSDUMP.DMP /SIZE=2052
%SYSGEN-I-EXTENDED, SYS$SYSROOT:[SYSEXE]SYSDUMP.DMP;1 extended
SYSGEN>  EXIT
```

The next example uses the following command procedure:

```
$ @SYS$UPDATE:SWAPFILES

To leave a file size at its current value type a
carriage return in response to its size prompt.
Current file sizes are:

Directory SYS$SYSROOT:[SYSEXE]

PAGEFILE.SYS;1    16384
SYSDUMP.DMP;1      4128
SWAPFILE.SYS;1     3072

Total of 3 files, 23584 blocks.

There are 128741 available blocks on SYS$SYSDEVICE.

Enter new size for page file: [RET]
Enter new size for system dump file: 6052
Enter new size for swap file: [RET]

%SYSGEN-I-EXTENDED, SYS$SYSROOT:[SYSEXE]SYSDUMP.DMP;1 extended
```

```
*****************************************************************
*  Please reboot in order for the new files to be used by the system. *
*  After rebooting, purge obsolete copies of the files.              *
*  DO NOT delete the old files until after the reboot.               *
*****************************************************************
```

Both the command procedure and the CREATE command automatically extend the size of a page or dump file if you specify a size that is larger than that of an existing file. If you specify a smaller size for a system page, swap,or dump file, a new version of the file is created. Remember to purge the file after rebooting.

Frequent file creation and deletion can cause the free space on a disk to become severely fragmented. Eventually, this can reach a state such that extensions to a file require more than one header block to map all the extents of the file. The bootstrap process requires that SWAPFILE.SYS (the primary swap file, if present) and PAGEFILE.SYS (the primary page file) be mapped by a single file header.

Note: SYSGEN issues a warning message if it determines that the creation or extension of a system file would cause that file to become fragmented enough to render the system unbootable. If this occurs, DIGITAL recommends that you back up and restore your system volume, in order to consolidate the free space on the volume into one contiguous area, and then retry the SYSGEN operation. In cases where SYSGEN issues a warning message, the file is somewhat larger, but not as large as the value specified in the CREATE command.

If you create a new version of a system file, you must delete the old version explicitly (but not until after the next bootstrap operation). In the case of a primary file (PAGEFILE.SYS, SWAPFILE.SYS, or SYSDUMP.DMP), the new or extended size file does not become effective until the system is shut down and restarted. In the case of a secondary file, the new file becomes effective when it is installed.

Use SYSGEN to calculate appropriate sizes for your system files as follows:

- Page file—Size of the average program at your site (in pages) times the maximum number of processes (MAXPROCESSCNT system parameter). The system installation sets an initial size for your primary page file. Use the DCL command SHOW MEMORY/FILE to display statistics on your page file usage. Examine the data pertaining to the page files. Sufficient space in the page file is critical to system performance. Aim to keep page file usage less than half the size of the page files. If a page file starts to fill to the point where system performance is being affected, a message will be printed on the console terminal. Should this happen, you should increase the size of your page file.

  You limit the amount of page file space consumed by user programs by using the /PGFLQUOTA qualifier of the AUTHORIZE commands ADD and MODIFY. (See the *VMS Authorize Utility Manual* for more information.) You should not reduce the value of /PGFLQUOTA below 1024. Size requirements of the page file vary widely depending on user applications.

- Dump file size—Size of physical memory in pages (to save the contents of memory if the system fails) plus the number of error log buffers plus one. The SYSGEN parameter ERRORLOGBUFFERS sets the number of one-page error log buffers to permanently allocate in memory. The ERRORLOGBUFFERS parameter ranges from 2 to 64, with the default set at 4.

- Swap file—Maximum number of processes (MAXPROCESSCNT system parameter) times the average of the working set quotas of processes running on the system. The system installation sets an initial size for SWAPFILE.SYS based on your hardware configuration. As an alternative to trying to calculate a more accurate swap file size, monitor the swap file usage with the DCL command SHOW MEMORY/FILE and watch its usage under load. Try to keep at least one-third of the swap file space unused; otherwise, system performance may be severely affected.

At bootstrap time, the system activates the latest versions of SYS$SYSTEM:PAGEFILE.SYS, SWAPFILE.SYS, and SYSDUMP.DMP. After bootstrapping, you can augment the primary page or swap file by adding secondary files. Using secondary files is advantageous, because they do not have to be on the system disk and they can span volumes in a volume set. You create secondary system files with the SYSGEN command CREATE. You install a secondary file by entering the SYSGEN command INSTALL to the site-specific command file SYS$MANAGER:SYPAGSWPFILES.COM, which is executed during system startup.

6.2.3 Connecting Devices and Loading Device Drivers

Usually, you enter the AUTOCONFIGURE command to automatically connect all devices physically attached to the system and to load their device drivers, saving effort and reducing the possibility of error. Devices not attached to the system and devices with nonstandard names can be connected and their device drivers loaded with explicit CONNECT (or CONNECT and LOAD) commands. You must exercise great care in issuing CONNECT and LOAD commands. (See the discussion of the System Generation Utility in the *VMS System Generation Utility Manual* and *VMS Device Support Manual.*)

Devices not connected automatically by AUTOCONFIGURE include the network communications logical device and the console block storage device. To connect the network communications logical device, use the following explicit CONNECT command:

```
SYSGEN> CONNECT NET/NOADAPTER/DRIVER=NETDRIVER
```

To connect the console block storage device, use the following explicit CONNECT command:

```
SYSGEN> CONNECT CONSOLE
```

The commands in the following example autoconfigure the devices physically attached to the system and explicitly connect the network software device and the console block storage device:

```
SYSGEN>  AUTOCONFIGURE ALL
SYSGEN>  CONNECT NET/NOADAPTER/DRIVER=NETDRIVER
SYSGEN>  CONNECT CONSOLE
SYSGEN>  EXIT
```

The AUTOCONFIGURE ALL command is included in STARTUP.COM, the site-independent startup command file. STARTUP.COM executes any additional commands found in SYSTARTUP_V5.COM, SYCONFIG.COM and STARTNET.COM.

A DIGITAL-supplied driver named SYS$SYSTEM:CONINTERR.EXE permits real-time processes to connect to interrupt vectors for quick response to and special handling of real-time events. The driver is not associated with any one device type. (See the *VMS I/O User's Reference Manual: Part I* for further information.)

6.2.4 Setting Up Virtual Terminals

You can also use the SYSGEN command CONNECT to set up virtual terminals. Virtual terminals allow you to disconnect from a physical terminal without terminating a process; the process remains active on a virtual terminal. Virtual terminals are used for the following purposes:

- To reconnect to a process when a modem line connection is lost
- To maintain sessions on more than one disconnected terminal.
- To use dynamic asynchronous DECnet communication

You set up virtual terminals by entering the following SYSGEN command:

```
SYSGEN> CONNECT VTA0/NOADAPTER/DRIVER=TTDRIVER
```

Virtual terminals are identified by the device name VTAn. After the above SYSGEN command is entered, any terminal with the TT2$M_DISCONNECT characteristic set prior to login is treated as a virtual terminal.

Note: LAT terminals (LTAn) can be disconnected if the TT2$M_DISCONNECT characteristic is set, but remote terminals (RTAn) cannot be disconnected.

There are two ways to set the TT2$M_DISCONNECT characteristic. You can enable the feature on a systemwide basis by setting the appropriate bit in the SYSGEN parameter TTY_DEFCHAR2, or you can enable the feature on a per-terminal basis by using the DCL command SET TERMINAL/DISCONNECT.

6.2.4.1 Reconnecting to a Disconnected Terminal Process

If virtual terminals are enabled, and a modem line connection is lost, the process remains active on the system as a disconnected virtual terminal process. You must reconnect to the process within the time period specified by the system parameter TTY_TIMEOUT (the default value is 900 seconds or 15 minutes). If you fail to reconnect to the process before this time expires, the process is deleted.

Note: You can connect only to a virtual terminal process associated with your user identification code (UIC).

A terminal can be disconnected in the following circumstances:

- You lose the modem signal between the host and the terminal.
- You press the BREAK key on a terminal with the TT2$M_SECURE characteristic set.
- You enter the DCL command DISCONNECT.
- You enter the DCL command CONNECT/CONTINUE.

To reconnect to a disconnected terminal, use one of the following methods:

- Allow the system login process (SYS$LOGIN) to make the connection using a command procedure.
- Enter the DCL command CONNECT VTAn.

6.2.4.2 Managing Disconnected Processes

Virtual terminals allow you to maintain more than one disconnected process at a time. You must keep in mind, however, that while you are logged in to a virtual terminal the physical terminal is disconnected. Any I/O requests directed to a device other than the physical terminal associated with your current virtual terminal process will enter a waiting state. The pending process will terminate when the timeout period expires. If, however, you reconnect to the physical terminal that is to receive the I/O request, the process continues from the point at which it entered the waiting state. Naming each process with a name that relates to its context makes it easier to reconnect to the desired process.

A system manager may want to restrict the use of virtual terminals. For example, if your system is close to exhausting nonpaged pool, you may not want to enable this feature on a systemwide basis. Each virtual terminal requires an additional data structure, a logical unit control block (UCB), in addition to the data structure for each physical terminal. You can control the number of virtual terminal sessions by the value specified in the system parameter MAXDETACH, or restrict the use of virtual terminals by enabling them on a per-terminal basis.

6.2.4.3 Using Dynamic Asynchronous DECnet Lines

Virtual terminals are required for dynamic asynchronous DECnet communication. A dynamic asynchronous line differs from a static asynchronous line or other DECnet–VAX line in that it is normally switched on for network use only for the duration of a dial-up connection between two nodes. Dynamic switching of terminal lines to asynchronous DDCMP lines can occur if the following requirements are met:

- Both nodes have DECnet–VAX licenses installed.
- The system manager at each node has loaded the asynchronous DDCMP driver NODRIVER.
- The system manager at each node has installed the privileged shareable image DYNSWITCH.
- The system manager at the remote node has virtual terminals enabled.

See the *VMS Networking Manual* for a detailed description of the procedure for setting up dynamic asynchronous DECnet lines.

6.2.4.4 Determining the Physical Terminal Type

You may want to determine the physical terminal associated with a virtual terminal. For instance, both direct connect and LAT lines may be virtual, but you may not know the terminal characteristics of a LAT terminal at system startup time. You can set the characteristics of direct connect lines at system startup; however, you must enter a SET TERMINAL/INQUIRE command to determine the characteristics of a LAT line. (See Chapter 7 for more information on LAT.)

Note: Using the command SET TERMINAL/INQUIRE clears the type-ahead buffer.

The following command procedure determines the physical terminal characteristics of both direct and LAT lines at system startup. Insert the following lines in your systemwide login procedure (SYLOGIN.COM). (This procedure assumes that your startup procedure has set all switched and LAT lines to "unknown".)

```
$ DEVCLASS = 'F$GETDVI ("SYS$COMMAND","DEVCLASS")'
$ IF DEVCLASS .ne. 66 then goto alldone   !Not a terminal
$ DEVTYPE = 'F$GETDVI ("SYS$COMMAND","DEVTYPE")'
$ IF DEVTYPE .ne. 0 then goto got_devtype
$ SET TERMINAL/INQUIRE   !Try to determine the device type
$ DEVTYPE = 'F$GETDVI ("SYS$COMMAND","DEVTYPE")'
$ got_devtype:
$! Can now dispatch on 'devtype' to do different things depending
$! on the type of terminal.
$ alldone:
```

You can uniquely identify a LAT terminal by using the F$GETDVI lexical function, specifying the item TT_ACCPORNAN. The function returns the terminal server node name and port name.

6.2.5 Specifying an Alternate Startup Command Procedure

Following a bootstrap operation, the system executes the current site-independent startup command procedure SYS$SYSTEM:STARTUP.COM. DIGITAL recommends that you do not modify the STARTUP.COM file initially supplied in the software distribution kit. Usually, site-specific modifications are added to SYS$MANAGER:SYSTARTUP_V5.COM. See Chapter 2 for more information on startup command procedures.

If, however, you want to specify an alternate site-independent startup command procedure, you can do so by entering the SYSGEN command SET/STARTUP. Use the SHOW/STARTUP command to display the current site-independent startup command procedure.

Enter the following command sequence to display the current startup procedure and to specify an alternate site-independent startup command procedure:

```
SYSGEN>  USE CURRENT
SYSGEN>  SHOW/STARTUP
Startup command file = SYS$SYSTEM:STARTUP.COM
SYSGEN>  SET/STARTUP SYS$SYSTEM:XSTARTUP.COM
SYSGEN>  WRITE CURRENT
%OPCOM, 15-APR-1988 16:04:06.30, message from user SYSTEM
%SYSGEN-I-WRITECUR, CURRENT system parameters modified by process
ID 00160030 into file SYS$SYSROOT:[SYSEXE]VAXVMSSYS.PAR
SYSGEN>  SHOW/STARTUP
Startup command file = SYS$SYSTEM:XSTARTUP.COM
SYSGEN>  EXIT
```

7 Connecting to a LAT Network

The Local Area Transport (LAT) communications protocol is a protocol that the VMS operating system uses within a local area network to communicate with terminal servers. Terminal servers are communication devices that connect terminals, modems, or printers to an Ethernet network.

Terminal servers provide a cost-effective method of connecting many user terminals to a computer. Terminal servers save on cable requirements, and they maximize the number of devices that can access a computer.

7.1 Function of the Local Area Transport (LAT) Protocol

The Local Area Transport (LAT) Protocol is the software that allows terminal server devices and computers to communicate within a local area network. LAT protocol is concerned with matching terminals and other devices to the computing resources of a local area network. Because LAT terminals no longer connect directly to a computer, a terminal server has to listen for terminal requests and must be able to match up user terminals with computers that provide the desired services.

Using the LAT protocol, a VMS operating system advertises its available services over the Ethernet. Terminal servers listen to the Ethernet advertisements and build a database of service information so that they can locate an appropriate VMS system when a user terminal requests computing services. For example, a user terminal might request general processing service or a data entry program. The terminal server uses LAT protocol to establish and maintain a connection between the requesting terminal and the VMS operating system.

Sometimes a VMS operating system can request services from a terminal server. The LAT protocol allows VMS systems to ask for connections to printers or other devices attached to a terminal server.

7.2 Advantages of the LAT Protocol

Instead of being restricted to the resources of one computer, users connected to a local area network by way of the LAT protocol can access the resources of any computer on the network. In addition, users can establish multiple computing sessions with different computers and switch easily from one session to another.

The LAT protocol provides load balancing features and recovery mechanisms so users get the best, most consistent service possible. In their broadcast messages, VMS systems rate the availability of their services so that terminal servers can establish connections to computing resources on the least busy node. If a node becomes unavailable for any reason, the servers attempt to provide alternate services.

System managers can also establish special computing environments using the LAT protocol. Application programs running under the VMS operating system, such as data entry programs or news services, can be configured as specific resources. When a user terminal requests a connection to the resource, the LAT protocol sets up a connection directly to the application program. No login procedure is necessary.

Another advantage of the LAT protocol is an improvement in system performance. Because the servers bundle messages onto a single Ethernet interface, a terminal server interface decreases the network traffic and reduces the number of computer interrupts realized in systems where terminals, modems, and printers each have a physical connection to the computer.

7.3 The LAT Network

A LAT network is any local area network where terminal servers and operating systems use the Local Area Transport (LAT) protocol. A LAT network can coexist on the same Ethernet with other protocols. The LAT protocol, which operates on both terminal servers and the VMS operating system, is designed to ensure the safe transmission of data over the Ethernet.

The LAT network consists of the following components:

- VMS service nodes
- Terminal server nodes
- Ethernet coaxial cable

VMS service nodes supply computing resources for the local network, while terminal server nodes port their terminals, modems, or printers to those resources upon request from a user terminal or an application program.

7.3.1 VMS Service Nodes

A VMS service node is one type of node in a LAT network. (Non-VMS nodes can also be used along with VMS nodes in a LAT network.) A service node is an individual computer in a local area network that offers its resources to users and devices. Because the VMS operating system contains the LAT protocol, any VMS system can be configured as a service node within a LAT network.

7.3.1.1 Types of Services

Each VMS node offers its resources as a *service*. Most often, a node offers a general processing service, but it can offer special application services as well. A system manager can create up to eight services. Any or all of the services can be specialized applications.

For example, a VMS service node might offer three services: one service for general processing, another for data entry, and a third for stock quotations. The general processing service would allow the use of the general computing environment. The data entry and stock services, on the other hand, would be restricted environments, with connections to the application service but to no other part of the service node.

Each service is distinguished by the name the system manager assigns to it. In a VMS cluster, DIGITAL recommends that the service name be the same as the cluster name. In an independent node, DIGITAL recommends that the service name be the same as the node name. With special service applications, the service holds the name of the application.

7.3.1.2 Service Advertisements

A VMS service node advertises its services over the Ethernet at regular intervals so that terminal servers know where network resources are available. The service announcement mentions the physical node name, the service names, a description of services, and a rating of service availability. Terminal servers listen to the Ethernet advertisements and record information in a database.

Whenever a user terminal requests service, the terminal server node requests connection to the appropriate VMS service node.

7.3.1.3 Print Requests

In some cases, VMS service nodes can request services from terminal servers. The most common situation is when the VMS system wants to use a printer that is ported to a terminal server. VMS submits the print request to the terminal server print queue that is set up and initialized in the VMS startup procedure. Then the LAT symbiont (the process that transfers data to or from a mass storage device) requests the LAT port driver to create and terminate connections to the remote printer.

7.3.2 Terminal Server Nodes

A terminal server node is the second type of node in a LAT network. This node is usually located near the terminals and printers it supports. These devices are physically cabled to the terminal server; the terminal server is physically connected to the Ethernet.

7.3.2.1 Locating VMS Service Nodes

Terminal servers build and maintain a directory of services from announcements they hear advertised over the network. Then, when terminal servers receive requests from terminal users, they can scan their service database and locate the computer that offers the requested service.

Terminal servers not only look for the VMS node that provides the requested service, they can also evaluate the service rating of that node. If a requested service is offered by more than one node, then the service rating is used to select the node that is least busy. A server establishes a logical connection between the user terminal and the VMS service node.

7.3.2.2 Setting up Connections

One logical connection carries all the data directed from one terminal server node to a VMS service node. That is, the server combines data from all terminals communicating with the same VMS node onto one connection. A terminal server only establishes a logical connection with a VMS service node if none exists.

If a connection fails for any reason, a terminal server attempts to find another node offering the same service and "rolls over" the connection so users can continue their computing sessions.

Even though terminal connections are bundled together, each terminal can be uniquely identified by its name. A terminal name consists of two parts. The first part is the name of the port on the terminal server that the terminal line plugs into; the second part is the name of the terminal server node.

7.3.2.3 Servicing VMS Nodes

Although terminal servers are usually the requesting nodes in a LAT network, sometimes VMS service nodes ask for service from terminal servers. Most commonly, a VMS service node queues print requests to remote printers connected to terminal servers.

7.4 Configuring a VMS Service Node

This section describes how to start up a LAT service node and assign characteristics to it in such a way that it compliments other networking software, such as VMS clusters and DECnet.

7.4.1 System Management Tasks

A VMS system becomes a LAT service node as soon as the LAT protocol starts advertising its services over the network.

As system manager, you start up the LAT protocol from the site-specific command procedure SYSTARTUP_V5.COM. The VMS startup procedure executes LTLOAD.COM.

LTLOAD.COM is the command procedure that invokes the LAT Control Program (LATCP) Utility. LATCP allows you to establish a LAT node name, a service name, and a message that is advertised over the network. In cases where a VMS system has two Ethernet controllers, LATCP allows you to configure your VMS service node to operate similarly on both local area networks or to offer different services on each.

Once the node is defined, LATCP starts the LAT port driver so that the VMS service node becomes part of the network.

LATCP also serves as a command interface to the LAT node. Its commands allow you to stop the LAT driver and modify characteristics of the VMS service node.

In order to set up a LAT network, your primary system management task is to edit SYSTARTUP_V5.COM to invoke LTLOAD.COM. Chapter 2 describes how to edit SYSTARTUP_V5.COM. Additional tasks are necessary in order to use remote printers connected to terminal servers or to create application services:

- If you are going to use remote printers connected to terminal servers or if you are creating special application services, edit LTLOAD.COM, as described in Chapter 2.
- To use printers that are connected to terminal servers, you must create a command procedure to set up and initialize queues to the remote printers. For information on setting up queues, see *Guide to Maintaining a VMS System*.

7.4.2 A Sample LAT Configuration

Figure 7–1 demonstrates the components of a LAT network. The network consists of an Ethernet cable connecting VMS service nodes and terminal server nodes.

The three VMS service nodes in Figure 7–1, named MOE, LARRY, and ALEXIS, each offer services to terminal server nodes on the network.

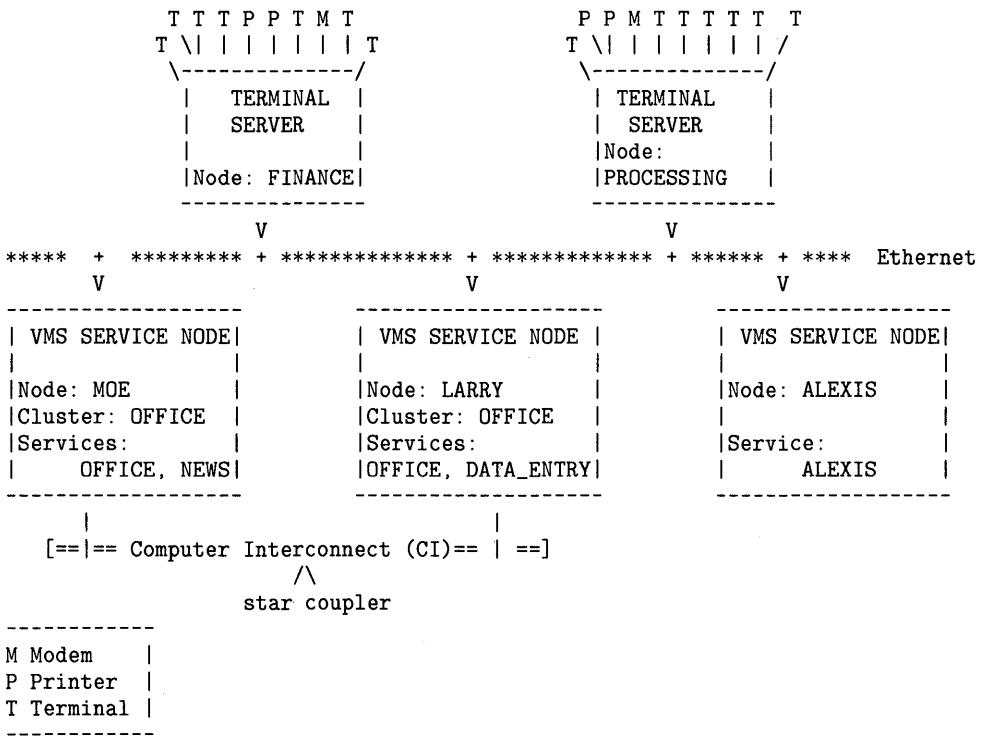
Two of the VMS service nodes, MOE and LARRY, belong to the OFFICE cluster. (The cluster is distinguished by its computer interconnect (CI) and star coupler.) Because MOE and LARRY are clustered, their service names are the same as their cluster names. Because both VMS service nodes offer an OFFICE service, terminal server nodes can assess the work load on both OFFICE nodes and establish a connection to a node that offers the service that is least busy.

The third VMS service node, ALEXIS, is an independent node in the LAT network so its service name is the same as its node name.

In addition to its primary "OFFICE" service, node MOE offers an application service called "NEWS". With this specialized service, user terminals can connect directly to the online news program, without any login procedure but also without general access to the general computer resources of the node.

The terminal server nodes, shown in Figure 7–1, are called FINANCE and PROCESSING. Each of them supports a number of interactive terminals as well as a modem and a printer. The terminal servers can accept print requests from any of the three VMS service nodes, provided each of the VMS nodes has set up print queues to support remote printers on the terminal server.

Figure 7–1 A LAT Network Configuration



7.4.3 LAT Relationship to VMS clusters and DECnet

Although the LAT protocol works independently of VMS cluster software, DIGITAL recommends that you configure a VMS service node to compliment the cluster concept. You achieve this by creating a service on each node in a cluster and assigning the cluster name to this service. A terminal server assesses the availability of cluster services and establishes a connection to the node that is least busy. Thus, the LAT protocol helps balance the cluster load. If one node in the cluster fails, the terminal server can transfer the failed connections to another service node within the cluster.

LAT does not use DECnet as a message transport facility, but instead uses its own virtual circuit layer to implement a transport mechanism. Essentially LAT and DECnet work independently in a common, Ethernet environment. For compatibility, if a VMS service node is also a DECnet node, the VMS service node name should be the same as the DECnet node name.

A Files on a VMS System Disk

This appendix describes the contents and organization of the directories and many of the files supplied by DIGITAL in the VMS operating system distribution medium. The files are located in a series of directories, some of which are reserved for system use.

The following list includes a brief description of each directory and its corresponding logical name:

1 [SYSCBI] (logical name SYS$INSTRUCTION)

This directory is reserved for EDTCAI files.

2 [SYSERR] (logical name SYS$ERRORLOG)

This directory is reserved for the error log file (ERRLOG.SYS).

3 [SYSEXE] (logical name SYS$SYSTEM)

This directory, listed in Table A–1, contains executable images commonly used by the operating system.

4 [SYSHLP] (logical name SYS$HELP)

This directory, listed in Table A–2, contains text libraries for the Help Utility and other components. The [SYSHLP] directory contains a subdirectory named [SYSHLP.EXAMPLES] with the logical name SYS$EXAMPLES. This subdirectory, listed in Table A–3, contains sample driver programs, user-written system service programs, and other source code examples of interest.

5 [SYS$LDR] (logical name SYS$LOADABLE_IMAGES)

This directory, listed in Table A–4, contains the set of images that are loaded during the bootstrap of the system. The Executive loaded images, device drivers, and other images loaded into the system reside in this directory. These loadable images are unique in that they cannot be executed with RUN or other DCL commands.

6 [SYSLIB] (logical name SYS$LIBRARY or SYS$SHARE)

This directory, listed in Table A–5, contains various macro and object libraries as well as shareable image files. The logical names SYS$LIBRARY and SYS$SHARE are synonymous.

7 [SYSMAINT] (logical name SYS$MAINTENANCE)

This directory is reserved for system hardware diagnostic programs.

8 [SYSMGR] (logical name SYS$MANAGER)

This directory, listed in Table A–6, contains files used in managing the operating system. This directory is the default directory for the system manager's account.

9 [SYSMSG] (logical name SYS$MESSAGE)

This directory, listed in Table A–7, contains system message text files.

10 [SYS$STARTUP] (logical name SYS$STARTUP)

This directory, listed in Table A–8, is used for starting up the VMS operating system. The logical name SYS$STARTUP is a searchlist that points to the [SYS$STARTUP] and [SYSMGR] directories.

11 [SYSTEST] (logical name SYS$TEST)

This directory, listed in Table A–8, contains files used to run the User Environment Test Package (UETP).

12 [SYSUPD] (logical name SYS$UPDATE)

This directory, listed in Table A–9, contains files used in applying system updates.

Table A–1 Files Contained in Directory [SYSEXE]

File Name	Description
ACC.EXE	Accounting Utility
ACLEDT.EXE	Access Control List (ACL) Editor
AGEN$FEEDBACK.EXE	AUTOGEN feedback image file
AGEN$FEEDBACK.DATA	AUTOGEN feedback data file
AGEN$FEEDBACK.REPORT	AUTOGEN feedback report file
ANALIMDMP.EXE	ANALYZE/PROCESS_DUMP image
ANALYZBAD.EXE	ANALYZE/MEDIA image
ANALYZOBJ.EXE	ANALYZE/IMAGE and ANALYZE/OBJECT image
ANALYZRMS.EXE	ANALYZE/RMS_FILE image
AUTHORIZE.EXE	User authorization program
BACKUP.EXE	Backup Utility
BADBLOCK.EXE	Dynamic bad block Files–11 ACP subprocess
BOOT58.EXE	Reserved for future use
BOOTBLOCK.EXE	Reserved for future use
CDU.EXE	Command Definition Utility
CHECKSUM.EXE	Used during installation of VMS updates
CIA.EXE	SHOW INTRUSION and DELETE/INTRUSION_RECORD image
CLUSTER_AUTHORIZE.EXE	Cluster authorization program
CNDRIVER.EXE	DECnet CI data link driver
CONFIGURE.EXE	Dynamic device configure process
CONVERT.EXE	Convert Utility
COPY.EXE	Copy utility
CREATE.EXE	File and directory creation utility
CREATEFDL.EXE	CREATE/FDL image
CSP.EXE	Cluster server process image
CVTNAFV5.EXE	Convert NETPROXY.DAT utility
CVTUAF.EXE	Convert SYSUAF.DAT utility

Table A–1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
DBLMSGMGR.EXE	DIBOL message manager
DCL.EXE	Command interpreter
DCLDEF.STB[1]	Global definitions for DCL structures
DELETE.EXE	File deletion/purge utility
DIFF.EXE	File compare utility
DIRECTORY.EXE	Directory utility
DISKQUOTA.EXE	Disk Quota Utility
DISMOUNT.EXE	Volume dismount utility
DSRINDEX.EXE	RUNOFF/INDEX image
DSRTOC.EXE	RUNOFF/CONTENTS image
DTR.COM	DTRECV.EXE server initiating procedure
DTRECV.EXE	DTSEND server
DTSEND.EXE	DECnet logical links test program
DUMP.EXE	Dump utility
EDF.EXE	File definition language editor
EDT.EXE	EDT text editor
ERF.EXE	ANALYZE/ERROR image
ERF*.EXE	ANALYZE/ERROR routines
ERRFMT.EXE	Error logging facility
EVL.COM	Command file used by DECnet error logging
EVL.EXE	DECnet event logging program
EXCHANGE.EXE	RT–11/DOS file transfer utility
F11AACP.EXE	Files–11 Structure Level 1 ancillary control process
F11BXQP.EXE	File system extended QIO processor
FAL.COM	FAL startup procedure
FAL.EXE	DECnet file access listener
FILESERV.EXE	File system cache flush server
HLD.COM	Command procedure used by HLD.EXE
HLD.EXE	Downline task loading program
HSCPAD.EXE	SET HOST/HSC command processor
IMGDEF.STB[1]	Global definitions for image activator structures
INIT.EXE	Disk device initialization utility
INPSMB.EXE	Card reader input symbiont
INSTALL.EXE	Utility that installs known images
JOBCTL.EXE	Job controller/symbiont manager
LADRIVER.EXE	LPA–11 driver

[1]Not supported by DIGITAL

Table A–1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
LALOAD.EXE	Accepts commands from or sends requests to LALOADER to load LPA–11 microcode
LALOADER.EXE	Loads LPA–11 microcode upon power recovery or upon request from LALOAD
LATCP.EXE	LAT–11 control program
LATSYM.EXE	LAT
LIBRARIAN.EXE	Librarian
LINK.EXE	Linker
LMF$LURT.DAT	License management facility file
LMF.EXE	License management facility program
LOGINOUT.EXE	Login/logout utility
MACRO32.EXE	VAX MACRO assembler
MAIL.COM	Command procedure used by DECnet mail
MAIL.EXE	Mail Utility
MAILEDIT.COM	Default MAIL editing command procedure
MAIL_SERVER.EXE	Mail server image
MESSAGE.EXE	Message compiler
MIRROR.COM	MIRROR startup procedure
MIRROR.EXE	DECnet node loopback server
MOM.COM	Maintenance operations module DECnet command procedure
MOM.EXE	Maintenance operations module image
MONITOR.EXE	Monitor Utility
MP.EXE	VAX–11/782 multiprocessing code
MP.MAP	Map to VAX–11/782 multiprocessing code
MP.STB[1]	Symbol table for MP.EXE
MP_8NN.EXE	VAX multiprocessing code
MP_8NN.MAP	Map to MP_8NN.MSKEXE
MP_8NN.MSKEXE	VAX multiprocessing code
MP_8NN.STB	Symbol table for MP_8NN.MSKEXE
MP_8SS.EXE	VAX multiprocessing code
MP_8SS.MAP	Map to MP_8SS.MSKEXE
MP_8SS.MSKEXE	VAX multiprocessing code
MP_8SS.STB	Symbol table for MP_8SS.MSKEXE
MSCP.EXE	MSCP server
MTAAACP.EXE	Magnetic tape ancillary control process
NCP.EXE	Network control program
NCS.EXE	National Character Set Utility
NETACP.EXE	DECnet ancillary control process

[1]Not supported by DIGITAL

Table A–1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
NETDEF.STB[1]	Symbol table for network definition
NETSERVER.COM	Network server DECnet command procedure
NETSERVER.EXE	Network server image
NICONFIG.COM	Ethernet configurator DECnet command procedure
NICONFIG.EXE	Ethernet configurator image
NML.COM	NML server startup procedure
NML.EXE	DECnet network management listener
NOTICE.TXT	Text file that can contain announcements to system users
OPCCRASH.EXE	System shutdown utility
OPCOM.EXE	Operator communications utility
PAGEFILE.SYS	System paging file
PARAMS.DAT	Data file for parameter values
PATCH.EXE	Patch Utility
PHONE.COM	PHONE startup procedure
PHONE.EXE	Phone Utility
PRTSMB.EXE	Print symbiont
QUEMAN.EXE	Queue managing utility
RECLAIM.EXE	CONVERT/RECLAIM image
RECOVER.EXE	RMS recovery utility
REMACP.EXE	Remote device ACP
RENAME.EXE	File rename utility
REPLY.EXE	Message broadcasting facility
REQUEST.EXE	Operator request facility
RMS.MAP	Map to RMS.EXE
RMS.STB[1]	RMS symbol table
RMSDEF.STB[1]	Global definitions for VAX RMS structures
RMSREC$RU_RECOVER.EXE	Recovery unit facility
RTB.EXE	Utility that writes an RT–11 bootstrap on disk
RTPAD.EXE	Remote terminal command interface
RUF*.*	Recovery unit facility
RUNDET.EXE	Facility that runs detached images
RUNOFF.EXE	DIGITAL Standard Runoff text formatting utility
SCSDEF.STB[1]	Symbol table for loadable routines
SDA.EXE	System Dump Analyzer
SDLNPARSE.EXE	SDL; Used for installing optional software
SEARCH.EXE	File search utility
SET.EXE	SET command processor

[1]Not supported by DIGITAL

Table A–1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
SETP0.EXE	SET MESSAGE command processor
SETRIGHTS.EXE	SET RIGHTS_LIST command processor
SETSHOACL.EXE	SET and SHOW ACCESS CONTROL LIST commands
SHOW.EXE	SHOW command processor
SHUTDOWN.COM	System shutdown command procedure
SHWCLSTR.EXE	SHOW CLUSTER command
SMGBLDTRM.EXE	Compiler for TERMTABLE definition file
SMGMAPTRM.EXE	Termtable global section, run at system startup
SMGTERMS.TXT	ASCII source file for DIGITAL terminal definitions
SMISERVER.EXE	System management server image
SMPUTIL.EXE	Multiprocessing utility
SORTMERGE.EXE	SORT and MERGE commands
SRTTRN.EXE	SORT specification file translator image
STABACCOP.EXE	Copy program for building standalone BACKUP kit
STABACKUP.EXE	Standalone Backup Utility
STACONFIG.EXE	HSC system disk configurator image
STANDCONF.EXE	Standalone BACKUP configure image
STARTUP.COM	System startup driver
STARTUP.INS	System startup
STARTUP.MAN	System startup
STASYSGEN.EXE	Standalone System Generation Utility
STOPREM.EXE	Stop REMACP utility
SUBMIT.EXE	Batch job submission utility
SUMSLP.EXE	Source file editor
SWAPFILE.SYS	System swap file
SYS.MAP	Map of the operating system
SYS.STB	Global symbol table of operating system
SYSBOOT.EXE	System bootstrap utility
SYSDEF.STB[1]	Global definitions for executive structures
SYSDUMP.DMP	System dump file
SYSGEN.EXE	System Generation Utility
SYSINIT.EXE	Operating system initialization image
SYSMAN.EXE	System management utility
SYSUAF.RL2	Unmodified copy of SYSUAF.DAT
TECO32.EXE	TECO text editor
TERMTABLE.EXE	Binary terminal definitions file
TERMTABLE.TXT	Terminal definitions source file

[1]Not supported by DIGITAL

Table A–1 (Cont.) Files Contained in Directory [SYSEXE]

File Name	Description
TKDRIVER.EXE	TK50 driver
TPU.EXE	VAXTPU text processing utility
TTRDVFY.EXE	Auxiliary terminal driver module
TYPE.EXE	Type utility
UNLOCK.EXE	File unlock utility
VAXVMSSYS.PAR	System parameter file
VERIFY.EXE	ANALYZE/DISK_STRUCTRURE image
VMB.EXE	VMS primary bootstrap
VPM.EXE	Cluster monitor
VMOUNT.EXE	Volume mount utility
VMSHELP.EXE	Help Utility
WRITEBOOT.EXE	System volume bootblock writing utility
WTDRIVER.EXE	VaxStation device driver
WTDRIVER.STB	Symbol table for VaxStation device driver
XADRIVER.EXE	Reserved for future use
XFDRIVER.EXE	DR32 system interconnect interface driver
XFLOADER.EXE	DR32 microcode loader utility
XWDRIVER.EXE	DUP11 device driver

Table A–2 Files Contained in Directory [SYSHLP]

File Name	Help Library
ACLEDT.HLB	Access Control List Editor
ANLRMSHLP.HLB	ANALYZE/RMS_FILE command
DEBUGHLP.HLB	Debugger
DISKQUOTA.HLB	Disk Quota Utility
EDFHLP.HLB	File Definition Language
EDTHELP.HLB	EDT
EDTVT100.DOC	EDT keypad layout for VT100
EDTVT52.DOC	EDT keypad layout for VT52
EVE$HELP.HLB	EVE help library
EVE$KEYHELP.HLB	EVE keypad help library
EXAMPLES.DIR	Examples directory
EXCHNGHLP.HLB	Exchange Utility
HELPLIB.HLB	Default (DCL) help library
INSTALHLP.HLB	Install Utility
LATCP.HLB	LAT Control Program
MAILHELP.HLB	Mail Utility

Table A–2 (Cont.) Files Contained in Directory [SYSHLP]

File Name	Help Library
MNRHELP.HLB	Monitor Utility
NCPHELP.HLB	Network Control Program
PATCHHELP.HLB	Patch Utility
PHONEHELP.HLB	Phone Utility
SDA.HLB	System Dump Analyzer
SHWCLHELP.HLB	Show Cluster Utility
SYSGEN.HLB	System Generation Utility
TECO.HLB	TECO
TFF$TFUHELP.HLB	Terminal fallback utility help library
TPUHELP	VAXTPU utility help file
SYSMANHLP.HLB	Sysman Utility help file
UAFHELP.HLB	User authorization file
VMSTLRHLP.HLB	Tailoring facility
WP.HLB	Watchpoint help library

Table A–3 Files Contained in Subdirectory [SYSHLP.EXAMPLES]

File Name	Description
ADDRIVER.MAR	Example device driver for AD11–K
ADDUSER.COM	Sample command procedure to add users to SYSUAF.DAT
BACKUSER.COM	Command procedure to back up user files and some system files
CONNECT.COM	Command procedure that connects device for LABIO system
DB*.*	Macro program for task-to-task communication with known network object
DOD_ERAPAT.MAR	Example loadable erase pattern generator
DRCOPY.PRM	Parameter file for DRCOPY routines
DRCOPYBLD.COM	Command procedure to build DRCOPY.EXE
DRMAST.MAR	VAX RMS interface for DRMASTER.FOR
DRMASTER.FOR	Master subroutines for DRCOPY
DRSLAVE.FOR	Slave subroutines for DRCOPY
DRSLV.MAR	VAX RMS interface for DRSLAVE.FOR
DTE_DF03.MAR	SET HOST/DTE modem support
EVE$*.TPU	EVE programs
GBLSECUFO.MAR	Opens file used as a global section for LABIO system
LABCHNDEF.FOR	Defines information associated with each A/D for LABIO system
LABIO.OPT	Linker options file for linking modules to be used in LABIO
LABIOACQ.FOR	Acquires data for LABIO system
LABIOCIN.MAR	Contains connect-to-interrupt call for LABIO system
LABIOCIN.OPT	Linker options file for linking LABIO_DATA_ACQ

Table A–3 (Cont.) Files Contained in Subdirectory [SYSHLP.EXAMPLES]

File Name	Description
LABIOCOM.FOR	Attaches a LABIO user program to the LABIO system modules of the LABIO system
LABIOCOMP.COM	Command procedure to compile and assemble the modules of the LABIO system
LABIOCON.FOR	Handles user requests and modifies the database for LABIO system
LABIOLINK.COM	Command procedure to link LABIO system
LABIOPEAK.FOR	Samples channel for peak data in LABIO system
LABIOSAMP.FOR	Samples channel in intervals, reporting date, time, and average value on logical device for LABIO system
LABIOSEC.FOR	Places LABIO_SECTION on page boundary
LABIOSTAT.FOR	Displays A/D channel status for LABIO system
LABIOSTRT.COM	Command procedure to start LABIO system
LABMBXDEF.FOR	Defines mailbox block for LABIO system
LBRDEMO.COM	Command procedure to create Librarian DEMO.EXE
LBRDEMO.FOR	Librarian demo (first part)
LBRMAC.MAR	Librarian demo (second part)
LOGIN.COM	Login command template
LPATEST.FOR	LPA11–K test program
LPMULT.B32	Example program for line printer
MAILCOMPRESS.COM	Sample procedure to compress mail files
MAILCVT.COM	Sample procedure to convert VMS Version 3.0 mail files
MAILUAF.COM	Sample procedure to manipulate SYS$SYSTEM:VMSMAIL.DAT
MGRMENU.COM	Command procedure for system manager menu
MONITOR.COM	Command procedure to generate MONITOR recording file
MONSUM.COM	Command procedure to generate cluster multi-file summaries
MSCPMOUNT.COM	Example cluster disk mount procedure
PEAK.FOR	Peak selection routine in LABIO system
RECOVERY_UNIT_ SERVICES.ADA	Sample source code for ADA
RESTUSER.COM	Command procedure to restore user and some system files from a backup done with BACKUSER.COM
RUFEXAMPLE.*	Sample programs using recovery unit facility code
SCRFT.MAR	Optional screen package (SCR$ . . . in RTL) extension to handle foreign terminals
SYSGTTSTR.MSG	Sample SYSGEN TERMINAL/ECHO message file
SUBMON.COM	Command procedure for using continuous MONITOR process
TDRIVER.MAR	Template for user-written driver
TESTLABIO.FOR	Tests LABIO system
USSDISP.MAR	Sample user system service dispatch and service examples
USSLNK.COM	Link command procedure for USSDISP

Table A–3 (Cont.) Files Contained in Subdirectory [SYSHLP.EXAMPLES]

File Name	Description
USSTEST.MAR	Sample program to invoke one of the example user services implemented in USSDISP
USSTSTLNK.COM	Link command procedure for USSTEST
XADRIVER.MAR	DR11 driver
XALINK.MAR	Sample DR11W-to-DR11W link program
XAMESSAGE.MAR	DR11 test program
XATEST.COM	Used to set up XALINK.MAR
XATEST.FOR	Companion program for XAMESSAGE
XIDRIVER.MAR	Example driver for parallel port on DMF32

Table A–4 Files Contained in Directory [SYS$LDR]

File Name	Description
ERRORLOG.EXE	Executive loaded image
EVENT_FLAGS_AND_ ASTS.EXE	Executive loaded image
EXCEPTION.EXE	Executive loaded image
EXEC_INIT.EXE	Executive loaded image
IMAGE_MANAGEMENT.EXE	Executive loaded image
IO_ROUTINES.EXE	Executive loaded image
LOCKING.EXE	Executive loaded image
LOGICAL_NAMES.EXE	Executive loaded image
MESSAGE_ROUTINES.EXE	Executive loaded image
PAGE_MANAGEMENT.EXE	Executive loaded image
PRIMITIVE_IO.EXE	Executive loaded image
PROCESS_MANAGEMENT.EXE	Executive loaded image
RECOVERY_UNIT_ SERVICES.EXE	Executive loaded image
RMS.EXE	Record Management Services, Executive loaded image
SECURITY.EXE	Executive loaded image
SYS.EXE	Operating system image file, Executive loaded image
SYSDEVICE.EXE	Executive loaded image
SYSGETSYI.EXE	Executive loaded image
SYSLICENSE.EXE	Executive loaded image
SYSTEM_DEBUG.EXE	Executive loaded image
SYSTEM_PRIMITIVES.EXE	Executive loaded image
SYSTEM_ SYNCHRONIZATION.EXE	Executive loaded image
WORKING_SET_ MANAGEMENT.EXE	Executive loaded image

Table A–4 (Cont.) Files Contained in Directory [SYS$LDR]

File Name	Description
VMS$SYSTEM_IMAGES.DATA	Executive loaded image data file
CLUSTRLOA.EXE	Loadable VAXcluster support code
CONINTERR.EXE	Connect-to-interrupt driver
SCSLOA.EXE	Loadable routines used by SCS
FPEMUL.EXE	Floating point emulation for F-, D-, G-, and H-floating point
VAXEMUL.EXE	VAX–11 instruction emulator
SYSLOA410.EXE	Processor-specific system image
SYSLOA41W.EXE	Processor-specific system image
SYSLOA750.EXE	Processor-specific system image
SYSLOA790.EXE	Processor-specific system image
SYSLOA8SS.EXE	Processor-specific system image
SYSLOAUV2.EXE	Processor-specific system image
SYSLOAWS2.EXE	Processor-specific system image
SYSLOA41D.EXE	Processor-specific system image
SYSLOA730.EXE	Processor-specific system image
SYSLOA780.EXE	Processor-specific system image
SYSLOA8NN.EXE	Processor-specific system image
SYSLOAUV1.EXE	Processor-specific system image
SYSLOAWS1.EXE	Processor-specific system image
SYSLOAWSD.EXE	Processor-specific system image
SYSLOA650.EXE	Processor-specific system image
SYSLOA65D.EXE	Processor-specific system image
SYSLOA65W.EXE	Processor-specific system image
SYSLOA8PS.EXE	Processor-specific system image
SYSLOA9CC.EXE	Processor-specific system image
CNDRIVER.EXE	DECnet CI data link driver
CTDRIVER.EXE	CTERM driver
CWDRIVER.EXE	8800 console driver
DDDRIVER.EXE	TU58 driver
DMDRIVER.EXE	RK07 disk driver
DRDRIVER.EXE	RM03 disk driver
DYDRIVER.EXE	RX02 floppy diskette driver
FYDRIVER.EXE	DUP (Diagnostics/Utilities Protocol) driver
LIDRIVER.EXE	DMB32 line printer driver
LTDRIVER.EXE	LAT–11 driver
NDDRIVER.EXE	DECnet pseudo data link driver
NODRIVER.EXE	Asynchronous DECnet driver
PUDRIVER.EXE	CI UDA port driver

Table A–4 (Cont.) Files Contained in Directory [SYS$LDR]

File Name	Description
RXDRIVER.EXE	Console RX50 driver
TMDRIVER.EXE	Magnetic tape driver
TTDRIVER.EXE	Terminal driver
XEDRIVER.EXE	UNA driver
XMDRIVER.EXE	DMC11 Synchronous Communications Line Interface driver
YCDRIVER.EXE	DMF32 asynchronous port driver
YFDRIVER.EXE	DHU port driver
CRDRIVER.EXE	Card reader driver
CVDRIVER.EXE	8600/8650 console driver
DBDRIVER.EXE	RP05 and RP06 disk driver
DLDRIVER.EXE	RL02 disk driver
DQDRIVER.EXE	RB730 driver
DUDRIVER.EXE	UDA disk driver
DXDRIVER.EXE	RX01 console floppy diskette driver
DZDRIVER.EXE	DZ11 port driver
LCDRIVER.EXE	DMF32 line printer driver
LPDRIVER.EXE	Line printer driver
MBXDRIVER.EXE	Shared memory mailbox driver
NETDRIVER.EXE	DECnet logical link driver
PADRIVER.EXE	CI780 port driver
RTTDRIVER.EXE	Remote terminal driver
TFDRIVER.EXE	TU78 driver
TSDRIVER.EXE	TS11 magnetic tape driver
TUDRIVER.EXE	Magnetic tape class driver
XDDRIVER.EXE	DECnet DMP11 data link driver
XGDRIVER.EXE	DECnet DMF data link driver
XQDRIVER.EXE	QNA driver
YIDRIVER.EXE	DMB32 terminal port driver
DVDRIVER.EXE	Driver
DZVDRIVER.EXE	Driver
ETDRIVER.EXE	Driver
PBDRIVER.EXE	Driver
WPDRIVER.EXE	Driver
ESDRIVER.EXE	Driver
FBDRIVER.EXE	Driver
PDDRIVER.EXE	Driver
YEDRIVER.EXE	Driver

Table A–5 Files Contained in Directory [SYSLIB]

File Name	Description
ACLEDIT.INI	Access Control List Editor initialization file
ACLEDIT.TPU	VAXTPU Access Control List Editor initialization file
ACLEDTSHR.EXE	VAXTPU ACL Editor routines
ACLEDT$SECTION.TPU$SECTION	Shareable VAXTPU Access Control List Editor initialization file
ADARTL.EXE	VAX ADA Run-Time Library
BASRTL.EXE	VAX BASIC Run-Time Library
BASRTL2.EXE	VAX BASIC Run-Time Library (part 2)
CDDSHR.EXE	Dummy CDD image for layered products
CLIMAC.L32	BLISS interface to CLI$ routines
CLIMAC.REQ	BLISS interface to CLI$ routines
COBRTL.EXE	VAX COBOL Run-Time Library
CONVSHR.EXE	CONVERT and CONVERT/RECLAIM shareable image
CRFSHR.EXE	Cross-reference shareable image
DBGSSISHR.EXE	VAX DEBUG system service intercept handler
DBLRTL.EXE	DIBOL Run-Time Library
DCLTABLES.EXE	DCL command tables
DCXSHR.EXE	Data compression support
DEBUG.EXE	VMS Debugger
DEBUG.TPU	Debug support for VAXTPU
DELTA.EXE	DELTA multimode debugging tool image
DELTA.OBJ	Alternate debugging tool
DISMNTSHR.EXE	DISMOUNT shareable image
DTE_DF03.EXE	SET HOST/DTE support for DF03 modem
DTE_DF112.EXE	SET HOST/DTE support for DF112 modem
DTKSWITCH.EXE	DECtalk utility routines
DYNSWITCH.EXE	Asynchronous DECnet–VAX routines
EDTSECINI.TPU	EDT Keypad Emulator interface to the VAXTPU text processing utility
EDTSECINI.TPU$SECTION	Shareable EDT Keypad Emulator interface to the VAXTPU text processing utility
EDTSHR.EXE	EDT editor
ENCRYPSHR.EXE	Dummy VAX Encryption support module
ERFCOMMON.EXE	ANALYZE/ERROR common data structures
ERFCTLSHR.EXE	ANALYZE/ERROR routines
ERFLIB.TLB	ANALYZE/ERROR device descriptions
ERFSHR.EXE	ANALYZE/ERROR common routines
EVESECINI.TPU	EVE interface to the VAXTPU text processing utility
EVESECINI.TPU$SECTION	Shareable EVE interface to the VAXTPU text processing utility
EVE$SECTION.TPU$SECTION	EVE section file

Table A–5 (Cont.) Files Contained in Directory [SYSLIB]

File Name	Description
FDLSHR.EXE	FDL parsing shareable image
FORDEF.FOR	FORTRAN INCLUDE file: FOR$ symbols
FORIOSDEF.FOR	FORTRAN INCLUDE file: IOSTAT error codes
FORRTL.EXE	VAX FORTRAN Run-Time Library
IMAGELIB.OLB	System default shareable image library
IMGDMP.EXE	Image dump procedures
LBRSHR.EXE	Librarian shareable image
LIB.L32	Operating system BLISS library
LIB.MLB	Operating system macro library
LIB.REQ	Structure definitions of executive internals for use by BLISS programs
LIBDEF.FOR	FORTRAN program utility INCLUDE files
LIBRTL.EXE	VAX Common Run-Time Library
LIBRTL2.EXE	VAX Common Run-Time Library (part 2)
MOUNTSHR.EXE	MOUNT shareable image
MTHDEF.FOR	FORTRAN INCLUDE files: MATH$ symbols
MTHRTL.EXE	VAX math support Run-Time Library
NMLSHR.EXE	DECnet management listener shareable image
PASRTL.EXE	VAX Pascal Run-Time Library
PLIRTL.EXE	VAX PL/I Run-Time Library
RPGRTL.EXE	VAX RPG Run-Time Library
SCRSHR.EXE	RTL terminal screen procedures shareable image
SCNRTL.EXE	VAX SCAN Run-Time Library
SECURESHR.EXE	Rights database (RIGHTSLIST.DAT) service routines
SIGDEF.FOR	FORTRAN program utility INCLUDE files
SMBSRVSHR.EXE	Print symbiont service routines
SMGSHR.EXE	VAX Screen Management Run-Time Library
SORTSHR.EXE	VAX Sort/Merge Run-Time Library
STARLET.L32	BLISS system library
STARLET.MLB	System macro library
STARLET.OLB	System object library and Run-Time Library
STARLET.REQ	User interface structures for use by BLISS programs
STARLETSD.TLB	Text library of STARLET definitions; used during layered product installations.
SUMSHR.EXE	Source update merge shareable image
TPAMAC.L32	BLISS TPARSE macros
TPAMAC.REQ	Structure definitions for BLISS programs using TPARSE
TPUSECINI.TPU$SECTION	Shareable VAXTPU text processing utility

Table A–5 (Cont.) Files Contained in Directory [SYSLIB]

File Name	Description
TPUSHR.EXE	VAXTPU text processing utility
TRACE.EXE	VMS error traceback facility
VAXCCURSE.OLB	VAX C Run-Time Library
VAXCRTL.EXE	VAX C Run-Time Library
VAXCRTL.OLB	VAX C Run-Time Library
VMSRTL.EXE	Run-Time Library shareable image
XFDEF.FOR	Definitions available for programs using DR780 support routines

Table A–6 Files Contained in Directory [SYSMGR]

File Name	Description
ACCOUNTNG.DAT	Accounting data file
ALFMAINT.COM	Command Procedure to Maintain Sys$system:sysalf.dat
CLUSTER_CONFIG.COM	Command procedure for cluster configuration
DBLSTRTUP.COM	Command procedure to start up DIBOL message manager
EDTINI.TEMPLATE	Template for EDT initialization file
LOADNET.COM	Command Procedure to Create Network Acp Process
LPA11STRT.COM	LPA11 site-specific startup command procedure
LTLOAD.COM	Command Procedure to Load and Start Lat
MAKEROOT.COM	Command procedure to add new roots to cluster common system disk
NETCONFIG.COM	Command procedure to configure network database
RTTLOAD.COM	Remote terminal loader
SECAUDIT.COM	Command procedure to extract information from the operator's log
SMISERVER.COM	Command procedure for SMISERVER
STARTNET.COM	DECnet startup procedure
SYSHUTDWN.COM	Site-specific system shutdown command procedure
SYSTARTUP.COM	Site-specific system startup command procedure
VMSIMAGES.COM	Invoked at startup to install known images
VMSIMAGES.DAT	Data file for VMSIMAGES.COM

Table A–7 Files Contained In Directory [SYSMSG]

File Name	Description
ADAMSG.EXE	ADA message file
CLIUTLMSG.EXE	ANALYZE/MEDIA, EXCHANGE, MAIL, PHONE, PRINT, SUBMIT, RUN, SET, SHOW, SEARCH
DBGTBKMSG.EXE	DEBUG, TRACE
DBLRTLMSG.EXE	DIBOL message file
FILMNTMSG.EXE	ANALYZE/OBJECT, ANALYZE/IMAGE, EDIT/FDL, ANALYZE/DISK
LMF_MESSAGE.EXE	License Management Facility message library
NETWRKMSG.EXE	NCP, SET HOST
PASMSG.EXE	Pascal message library
PLIMSG.EXE	PL/I message library
PRGDEVMSG.EXE	CDU, DIFF, DUMP, LIBRARY, LINK, MACRO, MESSAGE, PATCH, ANALYZE/SYSTEM, ANALYZE/CRASH
RPGMSG.EXE	RPG message library
SHRIMGMSG.EXE	CONVSHR, DCXSHR, FDLSHR, SORTSHR, SMGSHR, EDTSHR
SYSMGTMSG.EXE	ACC, EDIT/ACL, BACKUP, INSTALL, MONITOR, AUTHORIZE, SYSMAN
SYSMSG.EXE	System message file
TECOMSG.EXE	TECO message library
TPUMSG.EXE	VAXTPU text processing utility message file
VAXCMSG.EXE	VAX C message file

Table A–8 Files Contained in Directory [SYSTEST]

File Name	Description
TCNTRL.CLD	Defines commands to invoke the UETP test controller
UETCLIG00.COM	Command procedure to run the cluster integration phase
UETCLIG00.DAT	Used by UETP test controller to start cluster integration phase
UETCLIG00.EXE	Cluster integration phase
UETCOMS00.EXE	DMC and DMR device test
UETDISK00.EXE	Disk device test
UETDMPF00.EXE	DMP and DMF32 device test
UETDNET00.COM	Command procedure for the DECnet phase
UETDNET00.DAT	Used by UETP test controller to start DECnet phase
UETDR1W00.EXE	DR11–W device test
UETDR7800.EXE	DR780 and DR750 device test
UETFORT01.DAT	FORTRAN data file used by UETFORT01
UETFORT01.EXE	Artificial load in load test
UETFORT02.EXE	Artificial load in load test
UETFORT03.EXE	Artificial load in load test
UETINIT00.EXE	Checks UETP environment and sets overall parameters
UETINIT01.EXE	Quick checks devices and builds UETINIDEV.DAT
UETLOAD00.DAT	Used by UETP test controller to start load test phase
UETLOAD02.COM	User script for load test
UETLOAD03.COM	User script for load test
UETLOAD04.COM	User script for load test
UETLOAD05.COM	User script for load test
UETLOAD06.COM	User script for load test
UETLOAD07.COM	User script for load test
UETLOAD08.COM	User script for load test
UETLOAD09.COM	User script for load test
UETLOAD10.COM	User script for load test
UETLOAD11.COM	User script for load test
UETLPAK00.EXE	LPA11–K device test
UETMA7800.EXE	MA780 device test
UETMEMY01.EXE	Artificial load for load test
UETNETS00.EXE	Reports nonzero counters as part of DECnet phase

Table A–8 (Cont.) Files Contained in Directory [SYSTEST]

File Name	Description
UETP.COM	Main command procedure for entire UETP
UETPHAS00.EXE	Test controller
UETRSXFOR.EXE	Artificial load for load test
UETSUPDEV.DAT	Supported device data file
UETTAPE00.COM	Command procedure for magnetic tape device test
UETTAPE00.EXE	Magnetic tape device test
UETTTYS00.EXE	Terminal and line printer device test
UETUNAS00.EXE	DEUNA device test

Table A–9 Files Contained in Directory [SYSUPD]

File Name	Description
730CNSL.DAT	Data file to build an 11/730 console medium
750CNSL.DAT	Data file to build an 11/750 console medium
780CNSL.DAT	Data file to build an 11/780 console meduim
AUTOGEN.COM	Command procedure to calculate parameter values and system file sizes
BASEINSTAL.COM	Command procedure to install the MicroVMS Common Utilities Option
BLISSREQ.TLR	List of files in the BLISS tailoring group
BOOTBLDR.COM	Multiprocessing console floppy diskette command procedure
BOOTUPD.COM	Command procedure to update VMS bootstrap file on console floppy diskette
CONSCOPY.COM	Command procedure that copies console floppy diskette
CONSOLBLD.COM	Command procedure to build a VMS console medium
CVTNAF.COM	Command procedure to convert NETPROXY.DAT
CVTUAF.COM	Command procedure to convert SYSUAF.DAT
DECNET.TLR	List of files in the DECnet tailoring group
DIRCLENUP.COM	Command procedure to clean up dangling entries on a Version 3 or Version 4 VMS system
DISKITBLD.COM	Command procedure to build a VMS or MicroVMS kit
DEVELOP.TLR	List of files in the DEVELOP tailoring group
DXCOPY.COM	Command procedure that copies files from console floppy diskette and restores files to floppy diskette
EXAMPLES.TLR	List of files in the EXAMPLE tailoring group
FILETOOLS.TLR	List of files in the TOOLS tailoring group
HELP.TLR	List of files in the HELP tailoring group
LIBDECOMP.COM	Command procedure to expand libraries shipped in data-reduced format
LIBRARY.TLR	List of files in the LIBRARY tailoring group
MAKE_IMAGE.CLD	Used in VMS layered product installations
MANAGER.TLR	List of files in the MANAGER tailoring group

Table A–9 (Cont.) Files Contained in Directory [SYSUPD]

File Name	Description
MISCTOOLS.TLR	List of files in the MISCTOOLS tailoring group
MOVE.COM	Command procedure to move files from one directory to another while maintaining the order of the version numbers
QUEUES.TLR	List of files in the QUEUES tailoring group
REMOVE.COM	Command procedure to remove system files in an orderly manner
REQUIRED.TLR	List of files in the REQUIRED tailoring group
SETDEFBOO.COM	Command procedure that sets default boot command file
SPKITBLD.COM	Command procedure to build software product kits
STABACKIT.COM	Command procedure that builds standalone BACKUP to media
STA_MSCPKIT.COM	Command procedure to build a standalone VMS system kit for MSCP operation
SWAPFILES.COM	Command procedure that creates swapping, paging, and system dump files of appropriate size for system being installed
TEXTTOOLS.TLR	List of files in the TEXTTOOLS tailoring group
UETP.TLR	List of files in the UETP tailoring group
UVINSTAL.COM	Command procedure to install the MicroVMS Common Utilities Option
UVKITBLD.COM	Command procedure to build MicroVMS kits
VMBUVAX1.COM	Command procedure to build a floppy disk which can be used on MicroVAX I's to boot from disks with floating CSRs
VMSINSTAL.COM	Command procedure to install EDTCAI and maintenance updates
VMSKITBLD.COM	Command procedure that builds and copies VMS distribution disk
VMSKITBLD.DAT	List of files in VMS system that drives VMSKITBLD.COM
VMSMEDIA.COM	Command procedure to build any VMS media kit
VMSOPT.COM	Command procedure to install the optional file set to the VMS operating system
VMSTAILOR.COM	Tailoring facility command procedure (supported only on VAX–11/730)
VMSUPDATE.COM	System update command procedure

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Reader's Comments

Guide to Setting Up a VMS System
AA–LA25A–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

Guide to Setting Up a VMS System
AA–LA25A–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 ——————————————————————