OpenVMS System Manager's Manual: Essentials

December 1995

This manual is part one of a task-oriented guide to managing an OpenVMS system.

Revision/Update Information:	This manual supersedes the *OpenVMS System Manager's Manual: Essentials*, OpenVMS Alpha Version 6.2 and OpenVMS VAX Version 6.2.
Software Version:	OpenVMS Alpha Version 7.0 OpenVMS VAX Version 7.0

Digital Equipment Corporation
Maynard, Massachusetts

December 1995

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

Digital conducts its business in a manner that conserves the environment and protects the safety and health of its employees, customers, and the community.

© Digital Equipment Corporation 1995. All rights reserved.

The following are trademarks of Digital Equipment Corporation: ALL–IN–1, AXP, CI, DEC, DEC C, DECdtm, DECevent, DECnet, DECwindows, Digital, HSC, HSJ, IAS, InfoServer, LAT, LN03, LP27, MicroVAX, MicroVAX II, MSCP, OpenVMS, POLYCENTER, RSX–11M, RSX–11M–PLUS, RSX–11D, RZ, StorageWorks, TK, VAX, VAX C, VAXcluster, VAXstation, VMS, VMScluster, VT, and the DIGITAL logo.

The following are third-party trademarks:

Microsoft is a registered trademark and Windows is a trademark of Microsoft Corporation.

Motif and Open Software Foundation are registered trademarks of the Open Software Foundation, Inc.

IEEE and POSIX are registered trademarks of the Institute of Electrical and Electronics Engineers.

Oracle is a registered trademark and Oracle RDB is a trademark of Oracle Corporation.

POSTSCRIPT is a registered trademark of Adobe Systems Incorporated.

All other trademarks and registered trademarks are the property of their respective holders.

ZK6015

This document is available on CD–ROM.

Contents

Preface xxi

1 Overview of This Manual

1.1 Using the *OpenVMS System Manager's Manual* 1–1
1.2 How This Manual Relates to Other System Management Documentation 1–2
1.3 Finding Information About Managing Complex Environments 1–2
1.4 Finding Information About Managing Small Systems 1–3

2 Using OpenVMS System Management Utilities and Tools

2.1 Understanding OpenVMS System Management Tools 2–2
2.1.1 OpenVMS Management Station 2–2
2.1.1.1 Managing Resources 2–4
2.1.1.2 Managing Operations 2–4
2.1.2 DCL Commands 2–5
2.1.3 System Messages 2–5
2.1.4 DCL Command Procedures 2–5
2.1.5 System Management Utilities 2–7
2.1.6 MGRMENU.COM Command Procedure 2–9
2.2 Logging In to the SYSTEM Account 2–9
2.3 Using the System Management Utility (SYSMAN) to Centralize System Management 2–10
2.3.1 Understanding SYSMAN 2–11
2.3.1.1 Privileges Required 2–11
2.3.1.2 Tools and Commands 2–11
2.3.2 Enabling a Remote System to Execute SYSMAN Commands 2–12
2.3.3 Understanding a SYSMAN Management Environment 2–12
2.3.4 Defining the SYSMAN Management Environment 2–13
2.3.4.1 Defining Another Node as the Environment 2–14
2.3.4.2 Using Logical Names to Organize Management Environments . . . 2–14
2.3.4.3 Defining a VMScluster Environment 2–15
2.3.5 Understanding Your SYSMAN Profile 2–16
2.3.6 Adjusting Your SYSMAN Profile 2–16
2.3.6.1 Changing Your Current Privileges 2–17
2.3.6.2 Changing Your Default Device and Directory 2–17
2.3.7 Setting DCL Verification 2–17
2.3.8 Executing DCL Commands from SYSMAN 2–18
2.3.9 Creating SYSMAN Command Procedures 2–18
2.3.10 Setting Up SYSMAN with an Initialization File 2–19
2.4 Using OPCOM to Communicate with System Users 2–19
2.4.1 Starting OPCOM 2–22
2.4.2 Sending Messages to Users 2–22

2.4.3 Controlling the Use of OPA0: as an Operator Terminal 2–23
2.4.4 Designating Operator Terminals 2–23
2.4.5 Sending Requests to an Operator 2–25
2.4.6 Replying to Operator Requests 2–25
2.5 Using VMSKITBLD.COM to Modify a System Disk 2–26
2.5.1 Using VMSKITBLD.COM to Build a New System Disk 2–27
2.5.1.1 Completing a System Disk Built with VMSKITBLD.COM 2–29
2.5.2 Using VMSKITBLD.COM to Copy System Files to an Existing Disk 2–29
2.5.3 Using VMSKITBLD.COM to Add an Alternate System Root Directory 2–31
2.5.3.1 Configuring a System Root Added with VMSKITBLD 2–32

3 Installing, Upgrading, and Updating Software

3.1 Installing, Upgrading, and Updating the Operating System 3–2
3.1.1 Installing or Upgrading the OpenVMS Alpha Operating System 3–2
3.1.2 Installing or Upgrading from a Running Alpha System 3–3
3.1.3 Installing the OpenVMS VAX Operating System 3–4
3.1.4 Upgrading the OpenVMS VAX Operating System 3–5
3.1.5 Updating the OpenVMS Alpha or OpenVMS VAX Operating System 3–5
3.2 Installing or Upgrading Layered Products 3–6
3.3 Preparing Your System to Run VMSINSTAL.COM 3–6
3.3.1 Performing Preliminary Operations 3–7
3.3.2 Registering and Loading Licenses 3–8
3.3.3 Preventing Nodes from Sharing PAKs 3–8
3.4 Running VMSINSTAL.COM 3–9
3.4.1 Selecting a Product List 3–10
3.4.2 Selecting the Source 3–11
3.4.3 Selecting Options 3–11
3.4.4 Selecting the Destination 3–12
3.4.5 Verifying, Logging, and Confirming the Operation 3–13
3.4.6 Completing the Installation 3–13
3.5 Recovering from a System Failure 3–13
3.6 Selecting VMSINSTAL.COM Options 3–13
3.6.1 Using the Autoanswer Option (A) (Layered Products Only) 3–14
3.6.2 Using the Alternate Working Device Option (AWD=) 3–14
3.6.3 Using the Get Save Set Option (G) (Layered Products Only) 3–15
3.6.3.1 Storing a Product Save Set 3–15
3.6.3.2 Installing a Product 3–16
3.6.3.3 Specifying Backup Qualifiers 3–16
3.6.4 Using the File Log Option (L) 3–17
3.6.5 Using the Release Notes Option (N) 3–17
3.6.6 Using the Alternate Root Option (R) 3–18
3.7 Using the POLYCENTER Software Installation Utility 3–18
3.7.1 Interfaces to Use 3–18
3.7.2 Operations to Perform 3–19
3.7.3 Product Files and Databases 3–20
3.7.4 Format of Software Product Kits 3–20
3.7.5 Software Product Name Conventions 3–20
3.7.5.1 Version Identification Format 3–21
3.7.5.2 Looking at Software Product Name Examples 3–22

3.7.6 Creating a Product Configuration File (PCF) 3–22
3.7.6.1 Configuration Options 3–23
3.7.6.2 Configuration Commands 3–23
3.7.6.3 Recording Configuration Choices 3–24
3.7.6.4 Modifying an Existing PCF 3–26
3.7.7 Using a Product Database 3–26
3.7.7.1 Adding Information to the Database 3–26
3.7.7.2 Registering a Noncompliant Product 3–27
3.7.7.3 Detecting and Tracking Software Dependencies 3–27
3.8 Installing with the POLYCENTER Software Installation Utility 3–27
3.8.1 Performing Preliminary Steps 3–28
3.8.1.1 Assigning Privileges 3–28
3.8.1.2 Installing Prerequisite Software 3–29
3.8.1.3 Identifying Postinstallation Procedures 3–29
3.8.2 Extracting a Product's Release Notes 3–29
3.8.3 Installing a Product 3–29
3.8.3.1 Using an Existing PCF 3–30
3.8.3.2 Creating a New PCF During the Installation 3–30
3.8.4 Responding to Installation Questions 3–30
3.8.4.1 Requesting an Explanation to Questions 3–30
3.8.4.2 Accepting Default Answers 3–31
3.8.5 Confirming Your Answers 3–31
3.8.6 Performing the Installation as a Batch Job 3–32
3.9 Performing Other Operations on Installed Software Products 3–32
3.9.1 Reconfiguring an Installed Product 3–32
3.9.2 Recording a Change in Volume Label in the Product Database 3–33
3.9.3 Copying a Software Kit to a New Location 3–33
3.9.4 Converting a Software Kit from One Format to Another 3–33
3.9.5 Retrieving Product Information 3–33
3.9.5.1 Displaying Information About Objects 3–34
3.9.5.2 Displaying Information About the Products 3–34
3.10 Removing Installed Software Products and Kits 3–34
3.11 Using the DECwindows Motif Interface 3–34
3.11.1 The POLYCENTER Software Installation Utility Main Window 3–35
3.11.2 Specifying Operations to Perform 3–37
3.11.3 Selecting Products and Product Options 3–37
3.11.4 Performing an Operation 3–39
3.11.5 Customizing the DECwindows Motif Interface 3–39

4 Starting Up and Shutting Down the System

4.1 Understanding Booting and System Startup 4–2
4.1.1 Booting and Startup Processes 4–2
4.1.2 Types of Booting Operations 4–3
4.1.2.1 Nonstop Boot: The Most Common Booting Operation 4–3
4.1.2.2 Conversational Boot: For Special Booting Functions 4–3
4.1.3 System Startup and STARTUP.COM 4–4
4.1.4 Messages Indicating Booting and Startup Progress 4–5
4.2 Booting with Modified System Parameter Values 4–5
4.2.1 Booting After Showing or Modifying Individual System Parameter Values 4–6
4.2.2 Booting with an Alternate System Parameter File 4–7
4.3 Booting in an Emergency 4–7
4.3.1 Booting with Default System Parameters 4–8

4.3.2 Booting Without Startup and Login Procedures 4–8
4.3.3 Booting Without the User Authorization File 4–10
4.4 Booting with Controlled Startup 4–12
4.4.1 Booting with an Alternate Site-Independent Startup Procedure 4–12
4.4.2 Specifying an Alternate Default Startup Command Procedure 4–13
4.4.3 Booting with Minimum Startup 4–14
4.4.4 Booting While Displaying Startup Procedure Commands 4–14
4.4.5 Displaying Startup Procedure Commands with SYSMAN 4–15
4.5 Solving Booting Problems 4–16
4.6 Writing a New Boot Block on the System Disk 4–17
4.7 Using the Snapshot Facility to Reboot Nonclustered Systems (VAX Only) 4–19
4.7.1 Preparing Your Workstation for a System Image Snapshot (VAX Only) 4–20
4.7.2 Taking the Saved Image Snapshot (VAX Only) 4–22
4.7.3 Problem Solving When Using the Snapshot Facility (VAX Only) 4–23
4.8 Shutting Down the System 4–27
4.8.1 Performing an Orderly Shutdown with SHUTDOWN.COM 4–27
4.8.2 Understanding the Order of Shutdown Events 4–31
4.8.3 Customizing SHUTDOWN.COM to Perform Site-Specific Operations 4–32
4.8.3.1 Defining Logical Names 4–33
4.8.3.2 Modifying the Site-Specific Shutdown Command Procedure 4–34
4.8.4 Performing an Orderly Shutdown with the System Management Utility (SYSMAN) 4–34
4.8.5 Performing an Emergency Shutdown with the OPCCRASH.EXE Program 4–36
4.8.6 Performing an Emergency Shutdown Using Console Commands 4–37

5 Customizing the Operating System

5.1 Adding and Deleting Optional Files 5–1
5.2 Modifying Site-Specific Startup Command Procedures 5–2
5.2.1 Understanding Site-Specific Startup Command Procedures 5–2
5.2.2 Understanding the Order of Startup Events 5–4
5.2.3 Modifying SYPAGSWPFILES.COM to Install Page and Swap Files . . 5–6
5.2.4 Modifying SYCONFIG.COM to Configure Devices 5–7
5.2.4.1 Connecting Nonstandard Devices 5–7
5.2.4.2 Suppressing Autoconfiguration of Devices 5–8
5.2.5 Modifying SYLOGICALS.COM to Define Systemwide Logical Names 5–8
5.2.6 Modifying SYSECURITY.COM to Set Up Security Auditing 5–9
5.2.7 Modifying SYSTARTUP_VMS.COM to Perform General Operations 5–10
5.2.7.1 Mounting Public Disks 5–11
5.2.7.2 Mounting Disks That Must Be Available Early in Startup 5–12
5.2.7.3 Setting Terminal and Printer Characteristics 5–12
5.2.7.4 Starting Queues and Enabling Autostart for Queues 5–12
5.2.7.5 Installing Known Images 5–12
5.2.7.6 Installing Resident Images (Alpha Only) 5–13
5.2.7.7 Setting Up the OpenVMS InfoServer Client Software 5–13
5.2.7.8 Running the System Dump Analyzer 5–13
5.2.7.9 Purging the Operator Log File 5–14
5.2.7.10 Submitting Batch Jobs to Run at Startup Time 5–14

5.2.7.11 Creating Systemwide Announcements 5–14
5.2.7.12 Starting Up and Customizing the LAT Protocol Software 5–15
5.2.7.13 Starting the DECnet for OpenVMS Network 5–15
5.2.7.14 Starting the DIBOL Message Manager 5–16
5.2.7.15 Defining the Number of Interactive Users 5–16
5.3 Modifying Login Command Procedures to Customize User Environments 5–16
5.4 Customizing Startup Databases with the System Management Utility (SYSMAN) 5–18
5.4.1 Understanding Startup Databases 5–18
5.4.2 Understanding the Layered Product Startup Database 5–19
5.4.3 Specifying the Current Startup Database 5–20
5.4.4 Showing the Name of the Target Startup Database 5–20
5.4.5 Showing the Contents of a Startup Database 5–20
5.4.6 Adding Startup Files to a Startup Database 5–20
5.4.7 Changing Information Associated with a Startup File 5–21
5.4.8 Deleting a Record from a Startup Database 5–21
5.4.9 Preventing a Startup File from Executing 5–22
5.4.10 Allowing a Previously Disabled Startup File to Execute 5–22
5.5 Registering Images That Have System Version Dependencies (VAX Only) 5–22
5.5.1 Understanding System Version Dependency and the Image Registry (VAX Only) 5–23
5.5.2 Using the Image Registry Facility (VAX Only) 5–24
5.6 Customizing the Help Message Database 5–25
5.6.1 Accessing $STATUS Values for Uninstalled Messages 5–26
5.6.2 Creating System-Level Database Search Paths 5–28
5.6.3 Deleting Digital-Supplied Messages from the Database 5–29
5.6.4 Adding Comments to Digital-Supplied Messages 5–30
5.6.5 Changing Digital-Supplied Data 5–31
5.6.6 Adding Messages to Digital-Supplied Database Files 5–32
5.7 Customizing MAIL 5–32
5.8 Setting Correct Time Zone Information on Your System 5–33
5.8.1 Understanding Time-Setting Concepts 5–34
5.8.1.1 Coordinated Universal Time 5–34
5.8.1.2 Time Differential Factor 5–35
5.8.1.3 Daylight Saving Time and Standard Time 5–35
5.8.1.4 Time Zones 5–35
5.8.2 Determining Your System's Time Differential Factor 5–35
5.8.3 Using UTC$TIME_SETUP.COM 5–37
5.8.3.1 Setting the Time Zone on Your System 5–37
5.8.3.2 Setting the TDF on Your System 5–39
5.8.4 Adjusting for Daylight Saving Time and Standard Time 5–41
5.8.5 Setting Time in a VMScluster Environment 5–42
5.9 Setting Time Using the Battery-Backed Watch (BBW) (Alpha Only) 5–43
5.10 Choosing Languages, and Date and Time Formats 5–43
5.10.1 Specifying Languages Other Than English 5–44
5.10.2 Invoking LIB$DT_STARTUP.COM 5–45
5.10.3 Defining System Default Date and Time Formats 5–46
5.10.3.1 Defining Your Own Format 5–46
5.10.3.2 Using Predefined Formats 5–48
5.10.4 User Definitions of Language, and Date and Time Formats 5–50
5.11 Saving Your Customization 5–50

6 Managing User Accounts

6.1 Understanding the User Authorization File (UAF) 6–2
6.1.1 Priority 6–2
6.1.2 Limits and Quotas 6–2
6.1.3 Privileges 6–3
6.2 Understanding the Protection of Authorization Files 6–4
6.3 Understanding UAF Login Checks 6–5
6.4 Managing System-Supplied UAF Accounts 6–6
6.4.1 Understanding System-Supplied UAF Accounts 6–6
6.4.2 Creating Accounts on Alpha Systems (Alpha Only) 6–7
6.4.2.1 Creating Field Service Accounts (Alpha Only) 6–8
6.4.2.2 Creating SYSTEST and SYSTEST_CLIG Accounts (Alpha Only) 6–9
6.4.3 Maintaining System-Supplied Accounts (VAX Only) 6–9
6.4.4 Using the SYSTEM Account 6–10
6.4.5 Using AUTHORIZE to Maintain UAF Accounts 6–10
6.5 Preparing to Add User Accounts 6–12
6.5.1 Choosing an Account Type 6–12
6.5.2 Performing Additional Tasks 6–13
6.5.2.1 Selecting a User Name and Password 6–13
6.5.2.2 Assigning the User Identification Code (UIC) 6–14
6.5.2.3 Adding a Disk Quota Entry 6–15
6.5.2.4 Setting the User Default Device for an Interactive Account 6–15
6.5.2.5 Setting the User Default Device for a Captive Account 6–15
6.5.3 Understanding Account Security 6–16
6.6 Adding User Accounts 6–16
6.6.1 Adding a User Account with AUTHORIZE 6–17
6.6.2 Adding a User Account with a Command Procedure 6–18
6.7 Maintaining User Accounts 6–19
6.7.1 Using Command Procedures for Interactive Accounts 6–19
6.7.2 Modifying a User Account 6–23
6.7.3 Listing User Accounts 6–23
6.7.4 Maintaining the User Environment 6–24
6.7.5 Deleting a User Account 6–25
6.7.6 Using BACKUP to Remove User Files 6–26
6.7.7 Disabling a User Account 6–27
6.8 Restricting the Use of Accounts 6–27
6.8.1 Setting Day Types 6–27
6.8.2 Restricting Logins to Specific Times 6–28
6.8.3 Restricting Login Functions 6–28
6.8.4 Using Login Command Procedures for Restricted or Captive Accounts 6–29
6.8.5 Setting Priorities for User Processes 6–31
6.9 Setting Up Special Accounts 6–31
6.9.1 Setting Up an Automatic Login Account with SYSMAN 6–32
6.9.2 Setting Up a Project Account with ACL Identifiers 6–33
6.9.3 Understanding Network Proxy Accounts 6–34
6.9.4 Creating Network Proxy Authorization Files 6–35
6.9.5 Adding Proxy Accounts 6–36
6.9.6 Removing Proxy Accounts 6–37
6.9.7 Displaying Proxy Accounts 6–37
6.9.8 Controlling Proxy Logins 6–38
6.10 Managing MAIL 6–38
6.10.1 Modifying a User Record 6–38

6.10.2 Removing a User Record 6–39
6.10.3 AUTHORIZE Flags and MAIL 6–39
6.11 Managing System Resources 6–39
6.11.1 Understanding Pages and Pagelets 6–39
6.11.2 Setting Limits on System Resources 6–40

7 Managing Peripheral Devices

7.1 Understanding Device Names 7–1
7.2 Getting Information About Devices on the System 7–2
7.2.1 Determining If Volumes Need Rebuilding 7–4
7.2.2 Getting Information About ISO 9660-Formatted Devices 7–5
7.3 Setting Security Protection Characteristics on Devices 7–5
7.3.1 Granting Access to a Specific Device 7–6
7.3.2 Granting Access to All Devices 7–6
7.4 Connecting Devices and Loading Device Drivers 7–6
7.4.1 Manually Connecting Devices and Loading Device Drivers (VAX Only) 7–7
7.4.2 Manually Connecting Devices and Loading Device Drivers (Alpha Only) 7–8
7.4.3 Suppressing Autoconfiguration of Devices 7–9
7.5 Managing Terminals 7–10
7.5.1 Setting Terminal Characteristics 7–10
7.5.1.1 Setting Default Characteristics with System Parameters 7–10
7.5.1.2 Setting Characteristics in System Startup 7–11
7.5.2 Managing Virtual Terminals 7–11
7.5.2.1 Using Virtual Terminals for Dynamic Asynchronous DECnet for OpenVMS (VAX Only) 7–12
7.5.2.2 Determining the Physical Terminal Type of a Virtual Terminal . . . 7–13
7.6 Managing Printers 7–13
7.6.1 Setting Printer Characteristics 7–13
7.6.2 Using Spooled Printers 7–14
7.6.2.1 Spooling Printers 7–15
7.6.2.2 Despooling a Spooled Printer 7–16
7.6.2.3 Testing a Spooled Printer 7–16
7.7 Managing Tape Drives 7–16
7.7.1 Getting Magnetic Tape Device Information 7–17
7.7.2 Modifying Magnetic Tape Device Characteristics 7–17
7.8 Managing a Card Reader (VAX Only) 7–17
7.8.1 Distinguishing the Type of Card Deck (VAX Only) 7–18
7.8.1.1 Batch Job Card Deck (VAX Only) 7–18
7.8.1.2 Data Card Deck (VAX Only) 7–18
7.8.1.3 Setting Card Reader Translation Modes (VAX Only) 7–19
7.8.2 Running the Input Symbiont Interactively (VAX Only) 7–19

8 Managing Storage Media

8.1 Understanding Storage Media Concepts 8–2
8.1.1 Disk Concepts 8–2
8.1.1.1 Disk Terminology 8–2
8.1.1.2 Disk File Structures 8–3
8.1.1.3 CD–ROM File Structures 8–4
8.1.2 Tape Concepts 8–6

8.1.3 Public and Private Disk Volumes 8–7
8.1.3.1 Public Disk Volumes 8–8
8.1.3.2 Private Disk Volumes 8–9
8.2 Allocating and Deallocating Drives 8–9
8.2.1 Allocating Drives 8–9
8.2.2 Deallocating Drives 8–10
8.3 Initializing Volumes 8–10
8.3.1 Using INITIALIZE Command Qualifiers 8–12
8.3.2 Assisting Users in Accessing and Initializing Volumes 8–14
8.4 Protecting Volumes 8–14
8.4.1 Protecting Disk Volumes 8–15
8.4.1.1 Specifying Protection When You Initialize Disk Volumes 8–16
8.4.1.2 Changing Protection After Disk Volumes Are Mounted 8–18
8.4.1.3 Displaying UIC- and ACL-Based Protection 8–18
8.4.2 Protecting Tape Volumes 8–19
8.4.2.1 Using the /PROTECTION Qualifier with Tape Volumes 8–19
8.4.2.2 Protecting Tape Volumes for Interchange Environments 8–20
8.4.3 Auditing Volume Access 8–20
8.5 Mounting Volumes 8–20
8.5.1 Using MOUNT Command Qualifiers When You Mount Disks 8–22
8.5.2 Using MOUNT Command Qualifiers When You Mount Tapes 8–24
8.5.3 Assisting Users in Mounting Volumes 8–26
8.5.4 Mounting a Volume with Protected Subsystems 8–28
8.5.5 Modifying Disk Volume Characteristics 8–29
8.5.6 Speeding Up Disk Mounting 8–29
8.6 Setting Up Disk Volume Sets 8–30
8.6.1 Understanding Disk Volume Sets 8–30
8.6.1.1 Guidelines for Creating Disk Volume Sets 8–30
8.6.1.2 Using the /BIND Qualifier 8–31
8.6.2 Creating a Disk Volume Set from New Volumes 8–32
8.6.3 Creating a Shadowed Disk Volume Set 8–33
8.6.4 Creating a Disk Volume Set from an Existing Volume 8–33
8.6.5 Adding Volumes to an Existing Disk Volume Set 8–34
8.7 Mounting ISO 9660 Volume Sets and Groups 8–34
8.7.1 Mounting ISO 9660 Volume Sets 8–35
8.7.2 Mounting ISO 9660 Volume Groups 8–35
8.7.3 Handling Partially Mounted ISO 9660 Volume Sets 8–35
8.7.4 Handling ISO 9660 Restrictions 8–36
8.8 Mounting Tape Volume Sets 8–37
8.8.1 Creating a Tape Volume Set 8–37
8.8.2 Mounting Continuation Volumes in a Tape Volume Set 8–38
8.8.2.1 Creating Labels 8–39
8.8.2.2 Enabling Automatic Volume Switching 8–39
8.8.2.3 Disabling Automatic Switching 8–40
8.8.2.4 Sending Messages Back to Users 8–41
8.8.3 Modifying Magnetic Tape Characteristics 8–42
8.9 Dismounting Volumes and Volume Sets 8–42
8.9.1 Dismounting a Single Volume 8–44
8.9.1.1 Dismounting Before Unloading a Volume 8–44
8.9.1.2 Dismounting Allocated Devices 8–44
8.9.1.3 Using DISMOUNT Command Qualifiers 8–45
8.9.2 Dismounting a Volume Set 8–45
8.9.3 Dismounting Foreign Volumes 8–45
8.9.4 Dismounting a Volume in a VMScluster System 8–45

8.10 Using Command Procedures for Media Setup 8–46
8.10.1 Sample Command Procedure for Setting Up Disk Volumes 8–46
8.10.2 Sample Command Procedure for Setting Up Tape Volumes 8–47
8.11 Managing Disk Space 8–48
8.11.1 Understanding Disk Quotas 8–49
8.11.2 Establishing Disk Quotas 8–50
8.11.2.1 Creating a Quota File 8–51
8.11.2.2 Monitoring Disk Quotas 8–52
8.11.2.3 Suspending Quota Operations 8–52
8.11.3 Purging Files 8–53
8.11.4 Setting Version Limits on Files 8–53
8.11.5 Setting File Expiration Dates 8–54
8.12 Using the Analyze/Disk_Structure Utility to Check and Repair Disks . . . 8–55
8.12.1 Reporting Errors 8–56
8.12.2 Reporting and Repairing Errors 8–56
8.12.3 Recovering Lost Files 8–57
8.12.4 Erasing Old Home Blocks 8–58
8.12.5 Creating a Disk Usage File 8–58
8.13 Using Mount Verification for Recovery 8–58
8.13.1 Understanding Mount Verification 8–59
8.13.2 Using Mount Verification 8–60
8.13.2.1 Enabling Mount Verification 8–60
8.13.2.2 Controlling Timeout Periods for Mount Verification 8–60
8.13.2.3 Recovering from Offline Errors 8–61
8.13.2.4 Recovering from Write-Lock Errors 8–61
8.13.2.5 Canceling Mount Verification 8–62
8.14 Using the Bad Block Locator Utility to Detect Media Errors 8–64

9 Using Files and Directories

9.1 Using DCL Commands with Files 9–1
9.2 Getting File Information 9–2
9.3 Protecting Files 9–4
9.3.1 Understanding File Protection Concepts 9–4
9.3.2 Displaying File Ownership and Protection 9–5
9.3.3 Protecting Disk Files 9–6
9.3.3.1 Setting Default Disk File Protection 9–7
9.3.3.2 Setting Explicit Disk File Protection 9–7
9.3.3.3 Modifying Disk File Protection Characteristics 9–8
9.3.3.3.1 Changing File Protection Characteristics 9–9
9.3.3.3.2 Using the SET FILE Command 9–10
9.3.4 Protecting Disk Directories 9–10
9.3.4.1 Changing Directory UIC Protection Characteristics 9–12
9.3.4.2 Changing Default ACL Protection 9–12
9.3.5 Protecting Magnetic Tape Files 9–12
9.4 Accessing Disk Files 9–13
9.5 Accessing Tape Files 9–14
9.5.1 Understanding Tape File Names 9–14
9.5.2 Locating Standard-Labeled Tape Files 9–15
9.5.3 Using Wildcard Characters with Tape Volumes 9–16
9.5.4 Reading Files on Tape Volumes 9–17
9.5.5 Writing Files to Tape Volumes 9–17
9.5.5.1 Writing New Files That Overwrite Existing Files 9–17
9.5.5.2 Appending or Updating Files 9–18

9.6 Copying and Transferring Files 9–18
9.6.1 Copying Files to Disk Volumes 9–19
9.6.2 Copying Files to Tape Volumes 9–21
9.6.3 Continuing to Copy at the End of a Tape 9–24
9.6.4 Using the Exchange Utility (EXCHANGE) 9–24
9.6.5 Using the EXCHANGE/NETWORK Command 9–24

10 Using BACKUP

10.1 Overview of BACKUP Tasks 10–2
10.2 Understanding Types of Backups 10–3
10.3 Formulating a Backup Strategy 10–4
10.4 Understanding the Backup Interfaces 10–4
10.4.1 The BACKUP Command Line 10–5
10.4.2 The Backup Manager 10–5
10.4.2.1 Backup Manager Features 10–5
10.4.2.2 Getting Started with Backup Manager 10–6
10.5 Understanding Save Sets 10–6
10.5.1 Magnetic-Tape Save Sets 10–6
10.5.2 Files–11 Disk Save Sets 10–7
10.5.3 Network Save Sets 10–7
10.5.4 Sequential-Disk Save Sets 10–8
10.6 Understanding BACKUP File Formats 10–9
10.7 Setting Process Quotas for Efficient Backups 10–9
10.8 Using Disks and Tapes 10–12
10.8.1 Understanding Volume Initialization 10–13
10.8.1.1 When to Initialize Volumes 10–13
10.8.1.2 Initializing Tapes 10–13
10.8.1.3 Initializing Disks 10–14
10.8.2 Mounting a Volume 10–15
10.8.3 Dismounting a Volume 10–16
10.9 Understanding OPCOM and Volumes 10–17
10.9.1 Requesting Operator Assistance 10–17
10.10 Listing the Contents of a BACKUP Save Set 10–18
10.11 Understanding Multivolume BACKUP Operations 10–20
10.11.1 Multivolume Tape Labeling 10–20
10.12 Understanding BACKUP Tape Label Processing 10–21
10.13 Backing Up Files and Directories 10–22
10.13.1 Copying Files to Other Files 10–23
10.13.2 Backing Up Files and Directories to a Save Set 10–23
10.13.3 Comparing Files 10–25
10.13.4 Creating and Listing BACKUP Journal Files 10–26
10.14 Restoring Files and Directories 10–28
10.14.1 Accessing Files in Deep Directory Structures 10–29
10.15 Backing Up User Disks 10–30
10.15.1 Preparing to Back Up User Disks 10–30
10.15.2 Performing Image Backups to Tape 10–31
10.15.3 Performing Image Backups to Disk 10–32
10.15.4 Performing Incremental Backups to Tape 10–33
10.15.5 Performing Incremental Backups to Disk 10–34
10.15.6 Backing Up Your Workstation Disk 10–36
10.15.6.1 Using a Command Procedure for Nightly Image Backups 10–36
10.15.6.2 Using a Command Procedure for Nightly Incremental Backups . . 10–37
10.15.6.3 Using an Interactive Command Procedure for Backups 10–39

10.15.7 Backing Up Volume Shadow Sets 10–40
10.15.7.1 Mounting a Disk in a Host-Based Shadow Set 10–41
10.15.7.2 Assisted Merging in Mixed-Architecture Clusters 10–41
10.16 Restoring User Disks 10–42
10.16.1 Restoring Image Backups 10–42
10.16.2 Restoring Incremental Backups 10–44
10.16.2.1 Restoring to Target Disk Structures 10–46
10.16.3 Restoring Volume Shadow Sets 10–47
10.17 Backing Up and Restoring the System Disk 10–47
10.17.1 Starting the Menu-Driven Procedure 10–48
10.17.1.1 How to Perform This Task 10–48
10.17.1.2 Example 10–49
10.17.2 Understanding Standalone BACKUP (VAX Only) 10–50
10.17.2.1 Building Standalone BACKUP on a Disk (VAX Only) 10–51
10.17.2.2 Booting Standalone BACKUP from a Disk (VAX Only) 10–52
10.17.2.3 Building Standalone BACKUP on a Tape Cartridge (VAX Only) 10–53
10.17.2.4 Booting Standalone BACKUP from a Tape Cartridge (VAX Only) 10–54
10.17.3 Backing Up the System Disk to Tape 10–55
10.17.4 Restoring the System Disk from Tape 10–57
10.17.5 Backing Up the System Disk to a Disk 10–58
10.17.6 Using InfoServer Tapes to Back Up and Restore System Disks 10–60
10.18 Ensuring Data Integrity 10–62
10.18.1 /CRC Qualifier 10–62
10.18.2 /GROUP_SIZE Qualifier 10–62
10.18.3 /IGNORE Qualifier 10–63
10.18.4 /LOG Qualifier 10–63
10.18.5 /VERIFY Qualifier 10–63
10.19 Troubleshooting 10–64
10.19.1 BACKUP Fatal Error Options 10–64
10.19.2 Tape Label Errors 10–65

11 Security Considerations

11.1 Understanding Security Management 11–1
11.2 Managing Passwords 11–2
11.2.1 Initial Passwords 11–2
11.2.2 System Passwords 11–3
11.2.3 Primary and Secondary Passwords 11–4
11.2.4 Enforcing Minimum Password Standards 11–4
11.2.5 Guidelines for Protecting Passwords 11–5
11.2.6 Password History 11–6
11.3 Using Intrusion Detection Mechanisms 11–6
11.4 Understanding Ways to Protect Objects 11–7
11.4.1 Interpreting a User Identification Code 11–7
11.4.2 Understanding Protection Codes 11–8
11.5 Creating Access Control Lists (ACLs) 11–9
11.5.1 Kinds of Entries in an ACL 11–9
11.5.2 Types of Identifiers 11–10
11.6 Assigning ACLs 11–11
11.7 Using the ACL Editor 11–11
11.7.1 Adding an Identifier ACE 11–12
11.7.2 Setting a Default Protection Code 11–12

11.7.3 Generating Security Alarms and Audits 11–13
11.8 Auditing Security-Relevant Events 11–13
11.8.1 Enabling Classes of Security Alarms 11–13
11.9 Analyzing Audit Log Files 11–14

12 Managing the Queue Manager and Queue Database

12.1 Understanding the Queue Manager 12–1
12.1.1 Queue Manager Failover 12–3
12.1.2 Multiple Queue Managers 12–3
12.2 Understanding the Queue Database 12–4
12.3 Specifying the Location of Queue Database Files 12–5
12.3.1 Master File 12–6
12.3.2 Queue and Journal Files 12–6
12.4 Starting the Queue Manager and Creating the Queue Database 12–7
12.5 Customizing Queue Manager Failover 12–9
12.6 Stopping the Queue Manager 12–9
12.7 Restarting the Queue Manager 12–9
12.8 Creating an Additional Queue Manager 12–10
12.9 Displaying Information About the Queue Manager 12–11
12.10 Saving the Queue Database 12–11
12.11 Restoring the Queue Database 12–12
12.12 Maximizing Queue System Performance 12–13
12.13 Solving Queue Manager Problems 12–14
12.13.1 Avoiding Common Problems: A Troubleshooting Checklist 12–14
12.13.2 If the Queue Manager Does Not Start 12–14
12.13.2.1 Investigating the Problem 12–14
12.13.3 If the Queuing System Stops or the Queue Manager Does Not Run on Certain Nodes 12–15
12.13.3.1 Investigating the Problem 12–16
12.13.3.2 Cause 12–16
12.13.3.3 Correcting the Problem 12–17
12.13.4 If the Queue Manager Becomes Unavailable 12–17
12.13.4.1 Correcting the Problem 12–17
12.13.5 If the Queuing System Does Not Work on a Certain VMScluster Node 12–17
12.13.5.1 Investigating the Problem 12–17
12.13.5.2 Cause 12–18
12.13.5.3 Correcting the Problem 12–18
12.13.6 If You See Inconsistent Queuing Behavior on Different VMScluster Nodes 12–18
12.13.6.1 Investigating the Problem 12–19
12.13.6.2 Cause 12–19
12.13.6.3 Correcting the Problem 12–19
12.14 Reporting a Queuing System Problem to Digital 12–20

13 Managing Batch and Print Queues

13.1 Managing Queues on Small Systems 13–2
13.2 Understanding the Queuing Process 13–2
13.3 Understanding Types of Queues 13–2
13.3.1 Batch Execution Queues 13–3
13.3.2 Output Execution Queues 13–3
13.3.3 Generic Batch Queues 13–3

13.3.4 Generic Output Queues 13–3
13.3.4.1 Logical Queues 13–4
13.4 Understanding the Autostart Feature 13–4
13.5 Designing Your Batch Queue Environment 13–4
13.5.1 Simple Batch Queue Configuration: For Limited Batch Needs 13–5
13.5.2 Specialized Batch Queues: Meeting Special Needs 13–6
13.5.3 Generic Batch Queues in a VMScluster: Distributing the Work Load 13–7
13.6 Designing Your Output Queue Environment 13–8
13.6.1 Simple Output Queue Configuration: For Limited Printing Needs . . . 13–8
13.6.2 Mixed Printers: Specialized Output Execution Queues 13–9
13.6.3 PostScript Printing 13–9
13.6.4 LAT Printers: Sharing Printers Among Multiple Systems 13–10
13.6.5 Generic Output Queues: Distributing the Work Load 13–11
13.6.6 VMScluster Queues: Sharing Devices in a Cluster 13–12
13.6.7 Spooled Printers: Printing from Applications 13–13
13.6.8 Distributed Printing 13–13
13.7 Setting Up and Starting Queues 13–14
13.7.1 Setting Up Output Devices 13–14
13.7.2 Creating and Starting Queues 13–15
13.7.2.1 Autostart Execution Queues 13–15
13.7.2.2 Nonautostart Execution Queues 13–16
13.7.2.3 Generic Queues 13–17
13.7.3 Creating a Command Procedure to Restart Execution Queues on Reboot 13–17
13.8 Using Queue Options 13–18
13.8.1 Controlling Access to Queues 13–22
13.8.1.1 Understanding UIC-Based Queue Protection 13–22
13.8.1.2 Setting and Showing UIC-Based Queue Protection 13–23
13.8.1.3 Understanding ACL-Based Queue Protection 13–24
13.8.1.4 Setting and Showing ACL-Based Queue Protection 13–24
13.8.1.5 Understanding How Privileges Affect Queues 13–26
13.8.2 Understanding Job Retention Options 13–26
13.8.3 Understanding Queue Characteristics 13–28
13.8.4 Specifying Queue Characteristics Options 13–28
13.8.5 Specifying Batch-Processing Options 13–28
13.8.5.1 Base Process Priority 13–30
13.8.5.2 Job Limit 13–30
13.8.5.3 Working Set Default, Quota, and Extent 13–30
13.8.5.4 CPU Default and Maximum 13–31
13.8.5.5 Swapping 13–31
13.8.5.6 Setting Up Batch Queues on Memory-Constrained Systems 13–31
13.8.5.7 Optimizing Batch Queues for the Sort/Merge Utility 13–31
13.8.6 Specifying Job Scheduling Options 13–32
13.8.7 Understanding Banner Pages 13–33
13.8.7.1 Flag and Burst Pages 13–33
13.8.7.2 Trailer Pages 13–37
13.8.8 Specifying Banner Page Options 13–42
13.8.9 Understanding Forms 13–42
13.8.10 Specifying Forms Options 13–43
13.8.11 Specifying Options for Controlling Page and Line Overflow 13–43
13.8.11.1 Controlling Line Overflow with Forms 13–43
13.8.11.2 Controlling Page Overflow with the Form-Feed Character 13–44
13.8.12 Suppressing Initial Form Feed 13–44

13.8.13 Understanding Device Control Libraries 13–44
13.8.14 Specifying Device Control Library Options 13–45
13.8.15 Understanding the Order of Device Control Module Output 13–45
13.9 Maintaining the Queues 13–46
13.9.1 Managing Queues 13–46
13.9.1.1 Initializing Queues 13–48
13.9.1.2 Starting Nonautostart Queues 13–48
13.9.1.3 Enabling Autostart on a Node to Start Autostart Queues 13–48
13.9.1.4 Activating an Autostart Queue 13–49
13.9.1.5 Monitoring Queue Information 13–49
13.9.1.6 Modifying a Queue 13–53
13.9.1.7 Pausing a Queue 13–53
13.9.1.8 Closing a Queue 13–53
13.9.1.9 Stopping a Queue 13–54
13.9.1.10 Preventing Autostart Queues from Starting 13–54
13.9.1.11 Disabling Autostart on a Node 13–54
13.9.1.12 Stopping All Queues on a Node 13–55
13.9.1.13 Stopping Queues Before Shutting Down a System 13–55
13.9.1.14 Assigning a Logical Queue 13–56
13.9.1.15 Moving All Jobs from One Queue to Another 13–57
13.9.1.16 Deleting a Queue 13–57
13.9.2 Managing Characteristics 13–57
13.9.2.1 Defining Characteristics 13–58
13.9.2.2 Displaying Characteristics Defined on a System 13–59
13.9.2.3 Assigning Characteristics to a Queue 13–59
13.9.2.4 Displaying Characteristics Assigned to a Queue 13–59
13.9.2.5 Canceling Characteristics Assigned to a Queue 13–59
13.9.2.6 Deleting Characteristics 13–59
13.9.3 Managing Banner Pages 13–60
13.9.4 Managing Forms 13–60
13.9.4.1 Defining a Form 13–61
13.9.4.2 Displaying Forms Defined on a System 13–62
13.9.4.3 Changing the Systemwide Default Form 13–63
13.9.4.4 Assigning a Default Form for a Queue 13–63
13.9.4.5 Mounting a Form on a Queue 13–64
13.9.4.6 Displaying the Form Assigned to a Queue 13–64
13.9.4.7 Deleting a Form 13–64
13.9.5 Managing Device Control Libraries 13–65
13.9.5.1 Creating a Device Control Library and Inserting Modules 13–65
13.9.5.2 Assigning a Library to a Queue 13–66
13.9.5.3 Creating Forms for Setup and Page Setup Modules 13–67
13.9.5.4 Assigning a Reset Module to a Queue 13–67
13.9.6 Managing Jobs 13–68
13.9.6.1 Monitoring Jobs 13–68
13.9.6.2 Modifying Job Processing Options 13–70
13.9.6.3 Holding and Releasing a Job 13–70
13.9.6.4 Changing the Scheduling Priority of a Job 13–72
13.9.6.5 Requeuing an Executing Job 13–72
13.9.6.6 Requeuing a Pending Job 13–73
13.9.6.7 Retaining Jobs in a Queue 13–73
13.9.6.8 Deleting a Job 13–74
13.9.6.9 Pausing an Output Queue to Control Print Job Position and Alignment 13–75
13.9.6.9.1 Specifying the Position of Print 13–75

13.9.6.9.2 Aligning Print Forms 13–76
13.10 Solving Queue Problems 13–77
13.10.1 Determining the Cause of General Printer Problems 13–77
13.10.2 Making Pending Jobs Eligible for Scheduling 13–78
13.10.2.1 Fixing Print Jobs That Are Pending Due to Stock Mismatch 13–80
13.10.2.2 Fixing Jobs That Are Pending Because of Characteristics Mismatch 13–80
13.10.3 Fixing a Stalled Output Queue 13–81
13.10.4 Determining Why an Autostart Queue Will Not Start 13–81
13.10.5 Solving Problems Deleting a Queue, Form, or Characteristic 13–81
13.10.6 Adding or Deleting a Device Control Library Module 13–83
13.10.7 Fixing a Disabled Queue 13–84
13.10.8 Reporting Queue Problems to Digital 13–84

Index

Examples

3–1 Sample Procedure for Creating a PCF 3–25
3–2 Sample Command Procedure for Installing a Product 3–32
6–1 Sample Systemwide SYS$MANAGER:SYLOGIN.COM Login Command Procedure 6–20
6–2 Sample Login Command Procedure (LOGIN.COM) for a User Account 6–21
6–3 Command Procedure Template for Deleting an Account's Files 6–26
6–4 Sample Captive Command Procedure 6–30
8–1 Command Procedure to Set Up Tape Volumes 8–47
10–1 System Disk Backup to an InfoServer Tape 10–61
11–1 SHOW INTRUSION Output 11–7

Figures

1–1 Associated Documentation 1–3
2–1 Sample OpenVMS Management Station Screen 2–3
2–2 Sample SYSMAN Management Environment 2–13
2–3 Operator Communication Manager (OPCOM) 2–20
3–1 POLYCENTER Software Installation Utility Main Window 3–36
3–2 Selected Products Window 3–38
5–1 Time Differential Factor Map 5–36
8–1 Files–11 On-Disk Structure Hierarchy 8–3
8–2 Record Blocking 8–7
12–1 OpenVMS Batch and Print Queuing System 12–2
13–1 Default Batch Queue 13–5
13–2 Multiple Batch Queues with Special Resource and Performance Options 13–6
13–3 Batch Queue Configuration with Clusterwide Generic Queue 13–7
13–4 Simple Output Queue 13–8
13–5 Queue Configuration with Mixed Printers 13–9
13–6 Configuration for Remote Printers on a Terminal Server 13–10

13–7 Queue Configuration with Three Like Printers and a Generic Queue 13–11
13–8 Output Queue Configuration in a VMScluster 13–12
13–9 Queue Configuration with Spooled Devices 13–13
13–10 Determining Job Retention 13–27
13–11 Job Flag and Burst Pages 13–34
13–12 File Flag and Burst Pages 13–36
13–13 Job Trailer Page 13–38
13–14 File Trailer Page 13–39

Tables

1–1 Documentation for Managing Complex Environments 1–2
1–2 Documentation for Managing Small Standalone Systems 1–4
2–1 System Management Command Procedures 2–7
2–2 System Management Utilities 2–8
3–1 If You Are Installing 3–4
3–2 If You Are Upgrading 3–4
3–3 Format of *facvvu* Save-Set File Name 3–10
3–4 VMSINSTAL.COM Options 3–12
3–5 Operations and How to Initiate Them from Each Interface 3–19
3–6 PDF Kit Types and Values 3–21
3–7 Format of *tmmnn-ue* Version Identification 3–21
3–8 Features You Can Request During an Installation 3–29
3–9 SHOW OBJECT Command: Displaying Managed Object Information 3–34
3–10 SHOW PRODUCT and SHOW HISTORY Commands: Displaying Product Information 3–34
3–11 Operations Listed on the Mode Menu 3–37
4–1 Emergency Boot Procedures 4–7
4–2 Startup Logging Options 4–15
4–3 Node Notification During Shutdown 4–34
5–1 Site-Specific Startup Command Procedures 5–3
5–2 Commonly Defined System Logical Names 5–8
5–3 REGISTER_PRIVILEGED_IMAGE.COM Keywords 5–24
5–4 Time Zone Acronyms 5–38
5–5 Available Languages 5–45
5–6 Format Mnemonics 5–46
5–7 Predefined Output Date Formats 5–48
5–8 Predefined Output Time Formats 5–49
6–1 Resource Types 6–2
6–2 System Privileges 6–3
6–3 System Login Flow 6–5
6–4 System-Supplied UAF Accounts 6–7
6–5 Account Types 6–12
6–6 Qualifiers Used with AUTHORIZE 6–17
6–7 MAIL User Profile Record 6–38

6–8	Types of Limits	6–40
6–9	Limits and Suggested Values for SYSTEM and DEFAULT Accounts	6–40
6–10	Descriptions of SYSTEM and DEFAULT Accounts	6–42
8–1	Files–11 On-Disk Structure Terminology	8–2
8–2	Reserved Files	8–4
8–3	CD–ROM On-Disc Formats	8–5
8–4	CD–ROM Volume Structure	8–5
8–5	Terms Related to Magnetic Tapes	8–6
8–6	Tasks Users Can Perform Unassisted	8–11
8–7	INITIALIZE Command Qualifiers	8–13
8–8	Access Types for Disk and Tape Volumes	8–14
8–9	INITIALIZE Command Qualifiers for Protection	8–17
8–10	Protection Granted with INITIALIZE Command Qualifiers	8–17
8–11	Access Types for Tape Volume Protection	8–19
8–12	MOUNT Command Qualifiers for Mounting Disks	8–22
8–13	MOUNT Command Qualifiers for Mounting Tapes	8–24
8–14	ISO 9660 Restrictions	8–36
8–15	REPLY Command Qualifiers for Continuation Volumes	8–41
8–16	Contents of a Quota File	8–50
9–1	DCL Commands to Display Ownership and Protection	9–5
9–2	Access Types with Disk File Protection	9–6
9–3	DCL Commands to Modify File Protection Characteristics	9–8
9–4	Access Types for Directory Protection	9–11
9–5	Comparison of OpenVMS Extended Names and Standard Names	9–14
9–6	Wildcard Character Support with Tape Volumes	9–16
9–7	Methods of Transferring Information	9–19
10–1	Comparison of Image and Incremental Backups	10–4
10–2	BACKUP Command Qualifier Types	10–5
10–3	Save-Set Types	10–6
10–4	Recommended Process Quotas for Efficient Backups	10–10
10–5	Sample Process Quotas for Efficient Backups	10–11
10–6	Methods of Volume Initialization	10–13
10–7	Number of Shadow Sets Supported	10–40
10–8	Valid Standalone BACKUP Qualifiers	10–50
10–9	BACKUP Error Options and Results	10–64
11–1	Kinds of Security Events OpenVMS Can Report	11–13
13–1	Qualifiers for Specifying Queue Options	13–19
13–2	Effects of Queue Commands on All Queues	13–46
13–3	Effects of Queue Commands on Autostart and Nonautostart Queues	13–47
13–4	Queue Statuses Displayed in the SHOW QUEUE Command	13–50
13–5	Job Statuses Returned by SHOW ENTRY	13–69
13–6	SET ENTRY Qualifiers for Changing Jobs	13–70
13–7	Common Causes and Solutions for Pending Job Status	13–79

Preface

Intended Audience

The intended audience for this manual is OpenVMS system managers.

Document Structure

The *OpenVMS System Manager's Manual: Essentials* consists of the following chapters:

- Chapter 1, Overview of This Manual
- Chapter 2, Using OpenVMS System Management Utilities and Tools
- Chapter 3, Installing, Upgrading, and Updating Software
- Chapter 4, Starting Up and Shutting Down the System
- Chapter 5, Customizing the Operating System
- Chapter 6, Managing User Accounts
- Chapter 7, Managing Peripheral Devices
- Chapter 8, Managing Storage Media
- Chapter 9, Using Files and Directories
- Chapter 10, Using BACKUP
- Chapter 11, Security Considerations
- Chapter 12, Managing the Queue Manager and Queue Database
- Chapter 13, Managing Batch and Print Queues

For more information about the structure of the *OpenVMS System Manager's Manual*, see Section 1.1.

Related Documents

The following books are helpful when you use them in conjunction with the *OpenVMS System Manager's Manual*:

- *OpenVMS System Management Utilities Reference Manual*
- *OpenVMS User's Manual*
- *OpenVMS Software Overview*
- The current version of the *Upgrade and Installation Manual* for your system
- *OpenVMS Guide to System Security*
- *Guide to OpenVMS Performance Management*
- *VMScluster Systems for OpenVMS*

- The manuals in the networking kit of the OpenVMS Standard Documentation Set:
  - *DECnet for OpenVMS Guide to Networking*
  - *DECnet for OpenVMS Networking Manual*
  - *DECnet for OpenVMS Network Management Utilities*

For additional information on OpenVMS products and services, access the Digital OpenVMS World Wide Web site. Use the following URL:

`http://www.openvms.digital.com`

Reader's Comments

Digital welcomes your comments on this manual.

Print or edit the online form SYS$HELP:OPENVMSDOC_COMMENTS.TXT and send us your comments by:

Internet	**openvmsdoc@zko.mts.dec.com**
Fax	603 881-0120, Attention: OpenVMS Documentation, ZK03-4/U08
Mail	OpenVMS Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How To Order Additional Documentation

Use the following table to order additional documentation or information. If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Location	Call	Fax	Write
U.S.A.	DECdirect 800–DIGITAL 800–344–4825	Fax: 800–234–2298	Digital Equipment Corporation P.O. Box CS2008 Nashua, NH 03061
Puerto Rico	809–781–0505	Fax: 809–749–8300	Digital Equipment Caribbean, Inc. 3 Digital Plaza, 1st Street, Suite 200 P.O. Box 11038 Metro Office Park San Juan, Puerto Rico 00910–2138
Canada	800–267–6215	Fax: 613–592–1946	Digital Equipment of Canada, Ltd. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: DECdirect Sales
International	—	—	Local Digital subsidiary or approved distributor
Internal Orders	DTN: 264–4446 603–884–4446	Fax: 603–884–3960	U.S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063–1260

ZK–7654A–GE

Conventions

The name of the OpenVMS AXP operating system has been changed to OpenVMS Alpha. Any references to OpenVMS AXP or AXP are synonymous with OpenVMS Alpha or Alpha.

The following conventions are used to identify information specific to OpenVMS Alpha or to OpenVMS VAX:

Alpha	The Alpha icon denotes the beginning of information specific to OpenVMS Alpha.
VAX	The VAX icon denotes the beginning of information specific to OpenVMS VAX.
◆	The diamond symbol denotes the end of a section of information specific to OpenVMS Alpha or to OpenVMS VAX.

In this manual, every use of DECwindows and DECwindows Motif refers to DECwindows Motif for OpenVMS software.

The following conventions are also used in this manual:

Ctrl/*x*	A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 *x* or GOLD	A sequence such as PF1 *x* or GOLD *x* indicates that you must first press and release the key labeled PF1 or GOLD and then press and release another key or a pointing device button. GOLD key sequences can also have a slash (/), dash (–), or underscore (_) as a delimiter in EVE commands.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
. . .	Horizontal ellipsis points in examples indicate one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	Vertical ellipsis points indicate the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.

[ ]	In command format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in an OpenVMS file specification or in the syntax of a substring specification in an assignment statement.)
{ }	In command format descriptions, braces indicate a required choice of options; you must choose one of the options listed.
boldface text	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason (user action that triggers a callback). Boldface text is also used to show user input in Bookreader versions of the manual.
italic text	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system messages (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *device-name* contains up to five alphanumeric characters).
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`struct`	Monospace type in text identifies the following C programming language elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.
-	A hyphen in code examples indicates that additional arguments to the request are provided on the line that follows.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

1
Overview of This Manual

Together, the two parts of this manual explain tasks and concepts related to managing a system. This chapter describes this manual and how to use it.

The *OpenVMS System Manager's Manual* explains system management tasks so that both new and experienced system managers can perform them. However, before performing these tasks, you should be familiar with the following:

- User-level tasks such as creating and editing files and command procedures. For more information, see the *OpenVMS User's Manual*.
- DCL commands. For more information, see the *OpenVMS DCL Dictionary*.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Using the *OpenVMS System Manager's Manual*	Section 1.1
Finding information about managing complex environments	Section 1.3
Finding information about managing small systems	Section 1.4

This chapter explains the following concept:

Concept	Section
How this manual relates to other system management documentation	Section 1.2

1.1 Using the *OpenVMS System Manager's Manual*

The *OpenVMS System Manager's Manual* is made up of two parts:

- This book, *OpenVMS System Manager's Manual: Essentials*
- A second book, *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems*

Use these two books to get step-by-step instructions for general system management tasks.

The first page of each chapter in these books provides two tables to help you find information within the chapter: a task table and a concept table.

The Task Table

The first table lists the major *tasks* described in the chapter. To perform a task quickly, go directly to the section that explains that task. For example, in this chapter the task table lists the following tasks:

Task	Section
Using the *OpenVMS System Manager's Manual*	Section 1.1
Finding information about managing complex environments	Section 1.3
Finding information about managing small systems	Section 1.4

The Concept Table

The second table lists the major *concepts* explained in the chapter. If you want to learn more about an underlying concept, go to the appropriate concept section. For example, the concept table in this chapter lists the following concept:

Concept	Section
How this manual relates to other system management documentation	Section 1.2

1.2 How This Manual Relates to Other System Management Documentation

This manual is intended to be used as a companion to other OpenVMS system management manuals. Figure 1–1 shows the books you should be prepared to use along with the *OpenVMS System Manager's Manual*.

1.3 Finding Information About Managing Complex Environments

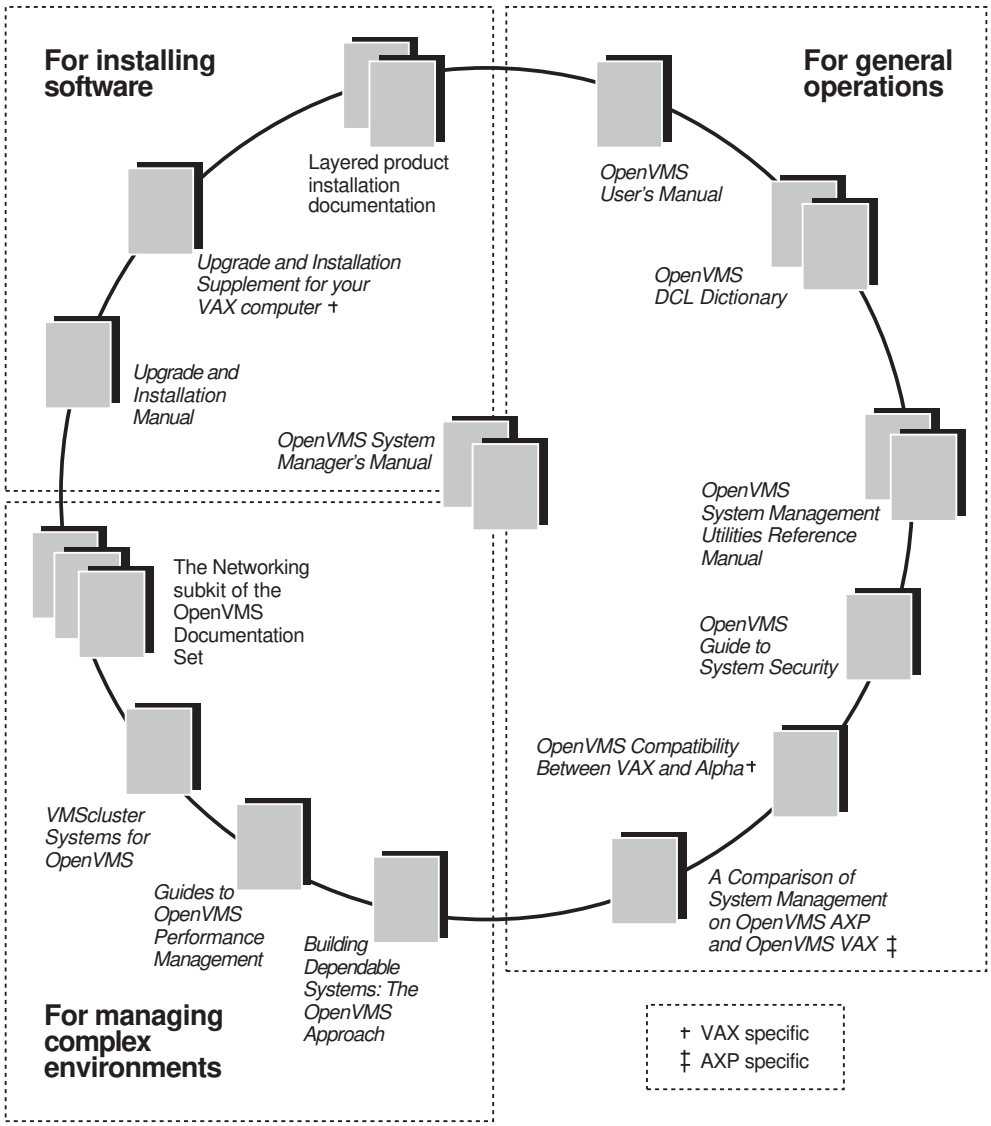
If you manage large or complex configurations, you may want additional specialized information. Table 1–1 lists some typical environments and OpenVMS manuals containing specialized information for managing those environments.

Table 1–1 Documentation for Managing Complex Environments

Topic	Manual
Networked environments	*DECnet for OpenVMS Guide to Networking* *DECnet for OpenVMS Networking Manual* *DECnet for OpenVMS Network Management Utilities* *DECnet/OSI Network Management*
VMScluster environments	*VMScluster Systems for OpenVMS*
Environments with both VAX and Alpha computers	‡*A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* †*OpenVMS Compatibility Between VAX and Alpha*
Highly available systems	*Building Dependable Systems: The OpenVMS Approach*

†VAX specific
‡Alpha specific

Figure 1–1 Associated Documentation



1.4 Finding Information About Managing Small Systems

If you manage a small standalone system—for example, a desktop workstation—you probably need to perform only basic system management tasks.

Table 1–2 lists the tasks you are likely to perform, and where to find instructions for performing these tasks.

Table 1–2 Documentation for Managing Small Standalone Systems

Task	Chapter, Section, or Other Manual
Installing and upgrading the operating system	The *Upgrade and Installation Manual*
Installing layered product software	Section 3.3
Loading software licenses	Section 3.3.2
Booting the system	Section 4.1.2.1
Shutting down the system	Section 4.8.1
Adding and deleting optional files from the system disk	Section 5.1
Modifying site-specific startup command procedures	Section 5.2
Modifying login command procedures	Section 5.3
Setting up user accounts	Section 6.5
Backing up workstation disks	Section 10.15.6
Backing up and restoring the system disk	Section 10.17
Starting the queue manager and creating the queue database	Section 12.4
Managing queues on small systems	Section 13.1
Modifying system parameters with AUTOGEN	Section 14.5
Tuning the system	Chapter 16

2

Using OpenVMS System Management Utilities and Tools

This chapter provides *general* information about system management utilities and tools that are provided with the OpenVMS operating system.

Procedures for using utilities and tools to perform specific tasks are provided in the respective chapters that describe those tasks. For example, this chapter contains a general description of the System Management utility (SYSMAN). Section 8.11.2 describes how to use SYSMAN to manage disk quotas. Section 14.7 describes how to use SYSMAN to manage system parameters.

To use system management tools, you can also refer to the following:

- Online help for online information about commands and qualifiers for DCL and system management utilities
- The *OpenVMS System Management Utilities Reference Manual* for detailed information about commands and qualifiers for system management utilities
- The *OpenVMS DCL Dictionary* for detailed information about DCL commands and qualifiers

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Logging in to the SYSTEM account	Section 2.2
Using SYSMAN to centralize system management	Section 2.3
Using OPCOM to communicate with system users	Section 2.4
Using VMSKITBLD.COM to modify a system disk	Section 2.5

This chapter explains the following concepts:

Concept	Section
OpenVMS system management tools	Section 2.1
DCL commands for system management	Section 2.1.2
System messages	Section 2.1.3
DCL command procedures for system management	Section 2.1.4
System management utilities	Section 2.1.5
MGRMENU.COM command procedure	Section 2.1.6
System Management utility (SYSMAN)	Section 2.3.1

Concept	Section
Understanding a SYSMAN management environment	Section 2.3.3
Understanding a SYSMAN profile	Section 2.3.5
Understanding OPCOM	Section 2.4

2.1 Understanding OpenVMS System Management Tools

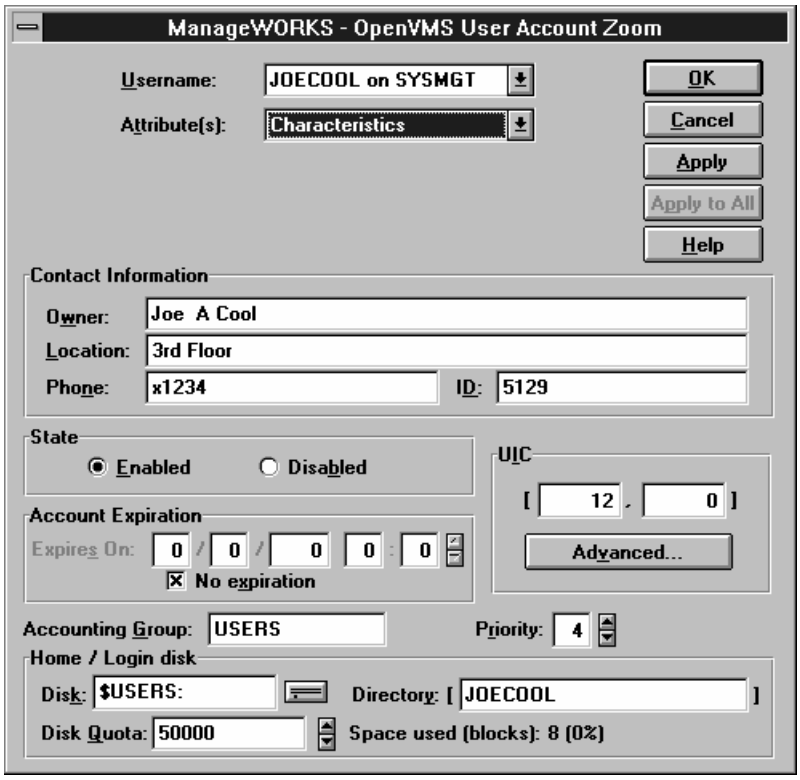
Digital supplies the following software tools to monitor and control system operations and resources:

Tool	For More Information
OpenVMS Management Station	Section 2.1.1
DIGITAL Command Language (DCL) commands; for example, COPY and MOUNT	Section 2.1.2
System messages	Section 2.1.3
Command procedures; for example, AUTOGEN.COM and STARTUP.COM	Section 2.1.4
System management utilities; for example, the Authorize utility (AUTHORIZE) and the Backup utility (BACKUP)	Section 2.1.5
MGRMENU.COM command procedure	Section 2.1.6
OPCOM	Section 2.4

2.1.1 OpenVMS Management Station

OpenVMS Management Station is a powerful, Microsoft Windows based management tool for system managers and others who perform account management tasks on OpenVMS systems. The OpenVMS Management Station software provides a comprehensive user interface to OpenVMS account management across multiple systems. You can manage multiple systems from a single source. OpenVMS Management Station software coexists with all of the existing OpenVMS system management utilities. Figure 2–1 shows a sample OpenVMS Management Station screen.

Figure 2–1 Sample OpenVMS Management Station Screen



OpenVMS Management Station addresses the problem of having to use multiple utilities to manage accounts. For example, creating an account usually involves the following steps:

1. Add a UAF entry
2. Grant rights identifiers
3. Create a directory
4. Create disk quotas
5. Grant network proxies

These steps require that you use DCL, the Authorize utility, and the DISKQUOTA component of the Sysman utility. OpenVMS Management Station provides an easy-to-use interface to this process.

OpenVMS Management Station consists of two components:

- The client. This is Microsoft Windows based software that you install on a PC and use to perform management operations.
- The server. This software must be installed on all OpenVMS systems to be managed. You do not interact directly with the server, the client does that.

Documentation for OpenVMS Management Station

The Microsoft Windows help files completely describe features, functions, instructions, and examples of using the OpenVMS Management Station. The *OpenVMS Management Station Overview and Release Notes* document provides an overview of OpenVMS Management Station and describes how to get started using the software.

Information about installing OpenVMS Management Station on your Alpha or VAX computer and your PC is located in the following manuals:

- *OpenVMS Alpha Version 7.0 Upgrade and Installation Manual*
- *OpenVMS VAX Version 7.0 Upgrade and Installation Manual*

2.1.1.1 Managing Resources

OpenVMS Management Station allows you to organize the systems you need to manage in ways that are meaningful to you and your environment, and allows you to manage user accounts on those systems.

You can easily manage user accounts across multiple OpenVMS systems, depending on your needs. The systems might be some of the clusters in a network, all of the systems on one floor of a building, a mix of clusters and nonclustered nodes, and so forth.

You can use OpenVMS Management Station to manage OpenVMS user accounts in a convenient, easy manner. For example, when creating an account on multiple systems, OpenVMS Management Station can add a user authorization file (UAF) entry, grant rights identifiers, create an OpenVMS directory, set a disk quota, set up OpenVMS Mail characteristics, and so forth, for each instance of the account.

OpenVMS Management Station manages the following OpenVMS resources:

- The SYSUAF.DAT user authorization file
- The RIGHTSLIST.DAT user rights file
- The network proxy database
- Account login-directory trees
- User account disk quotas
- The OpenVMS Mail VMSMAIL_PROFILE.DATA file

2.1.1.2 Managing Operations

OpenVMS Management Station supports the following account management operations:

- Creating user accounts
- Modify user accounts (any aspect)
- Deleting user accounts
- Renaming user accounts
- Displaying user account attributes

2.1.2 DCL Commands

You perform many system management tasks by entering DCL (DIGITAL Command Language) commands. For example, you enter the DCL command MOUNT to make disks and tapes available to the system. Most of the DCL commands used by system managers require special privileges (such as OPER privilege). See the *OpenVMS DCL Dictionary* or DCL Help for the required privileges and syntax for each command.

The general format of a DCL command is as follows:

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

Because a command can be continued on more than one line, the term **command string** is used to define the entire command. A command string is the complete specification of a command, including the command name, command qualifiers, parameters, and parameter qualifiers.

For complete descriptions of each DCL command, see the online DCL help or the *OpenVMS DCL Dictionary*. If you are not familiar with DCL command syntax, see the *OpenVMS User's Manual*.

2.1.3 System Messages

When you enter commands in DCL or in utilities, the system returns messages to help you understand the result of each command. **System messages** can indicate the following:

- Successful completion of a command
- Information about the effect of the command
- Warning about the effect of the command
- Failure to successfully complete the command

At times, you might need to interpret a system message, for example, to find out how to recover from a warning or failure. The Help Message utility allows you and system users to quickly access online descriptions of system messages from the DCL prompt.

For more information about the Help Message utility, see the *OpenVMS System Messages: Companion Guide for Help Message Users*. In addition, the *OpenVMS System Messages and Recovery Procedures Reference Manual* provides detailed descriptions of system messages.

2.1.4 DCL Command Procedures

You can use command procedures to efficiently perform routine tasks. A **command procedure** is a file containing DCL commands and, optionally, data used by those DCL commands. When you execute a command procedure, the system reads the file and executes the commands it contains. This eliminates the need for you to enter each command interactively. You can create command procedures to automate some of the routine system management tasks specific to your site.

A simple command procedure can contain a sequence of commands that you use frequently. For example, you could include the following commands in a command procedure called GO_WORK.COM:

```
$ SET DEFAULT [PERRY.WORK]
$ DIRECTORY
$ EXIT
```

When you execute this command procedure with the command @GO_WORK, you set your default directory to [PERRY.WORK] and display a list of files in that directory.

With complex command procedures, you can use DCL instead of a high-level programming language. For more information on creating command procedures, see the *OpenVMS User's Manual*.

Executing Command Procedures in Batch Mode

You can execute command procedures in **batch mode** by submitting the procedure to a batch queue. When resources are available, the system creates a batch process to execute the commands in the procedure. Usually, processes running in batch mode execute at a lower process priority to avoid competing with interactive users for system resources.

You might execute a command procedure in batch mode for the following reasons:

- To automate a task
- To process work at a lower scheduling priority, so as not to compete with interactive users for system resources
- To perform a task during off hours, such as at night or on weekends
- To allow an operation to continue without having a terminal logged in, thereby increasing the security of the system

A batch-oriented command procedure can include a command to resubmit itself to a batch queue, thereby repetitively performing the task with no user intervention. For example, you might create a batch-oriented command procedure to run the Analyze/Disk_Structure utility to report disk errors. If you include a command to resubmit the procedure to a batch queue, the procedure will automatically execute when scheduled, unless errors cause the procedure to fail. The following is a simple example of this kind of command procedure, named SYSTEM-DAILY.COM:

```
$ SET NOON
$! Resubmit this procedure to run again tomorrow.
$!
$ SUBMIT/KEEP/NOPRINT/QUEUE=SYS$BATCH/AFTER="TOMORROW+1:00"/USER=SYSTEM -
  SYS$MANAGER:SYSTEM-DAILY.COM;
$!
$! Purge the log files
$ PURGE/KEEP=7 SYS$MANAGER:SYSTEM-DAILY.LOG
$!
$! Analyze public disks
$!
$ ANALYZE/DISK/LIST=SYS$MANAGER:WORK1.LIS; WORK1:
$ ANALYZE/DISK/LIST=SYS$MANAGER:WORK2.LIS; WORK2:
$!
$! Print listings
$!
$ PRINT/QUEUE=SYS$PRINT SYS$MANAGER:WORK1.LIS;,SYS$MANAGER:WORK2.LIS;
$ EXIT
```

Digital-Supplied Command Procedures for System Management

Digital provides several command procedures that you can use when managing a system. Table 2–1 lists some commonly used command procedures.

Table 2–1 System Management Command Procedures

Command Procedure	Function
SYS$SYSTEM:STARTUP.COM	The system uses this command procedure to automatically perform certain tasks that are required to start up an OpenVMS system. This procedure is executed when the system boots. *Do not modify this command procedure*.
SYS$STARTUP:SYSTARTUP_VMS.COM	STARTUP.COM executes this procedure when the system boots. Add commands to this procedure to perform site-specific tasks each time the system boots.
SYS$SYSTEM:SHUTDOWN.COM	Use to shut down the system in an orderly fashion.
SYS$UPDATE:AUTOGEN.COM	Use to automatically set system parameters and page, swap, and dump file sizes to values appropriate for the system configuration and work load.
SYS$UPDATE:VMSINSTAL.COM	Use to install software on a running system.

2.1.5 System Management Utilities

With the operating system, Digital supplies a number of **system management utilities** to help perform system management tasks. A system management utility is a program that performs a set of related operations. For example, you use the Mount utility (MOUNT) to make disks and tapes available to the system, and you use the Backup utility (BACKUP) to save and restore files.

Most system management utilities require special privileges. Generally, you run these utilities from the SYSTEM account, which has all privileges by default. Section 2.2 describes logging in to the SYSTEM account.

You invoke some utilities using the following command format:

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

To invoke other utilities, such as MOUNT and ANALYZE/DISK_STRUCTURE, you enter a DCL command. For example:

```
$ ANALYZE/DISK_STRUCTURE
```

For information on invoking utilities and the commands you can use in these utilities, see the *OpenVMS System Management Utilities Reference Manual*.

Table 2–2 lists the system management utilities and their purposes. This manual describes how to use most of these utilities. For detailed information on utility commands and qualifiers, see the *OpenVMS System Management Utilities Reference Manual*.

Table 2–2 System Management Utilities

Utility	Purpose
Accounting utility (ACCOUNTING)	To produce reports of resource use.
ACL editor (access control list editor)	To create and maintain ACLs.
Analyze/Disk_Structure utility (ANALYZE/DISK_STRUCTURE)	To check the validity of Files–11 Structure Level 1 and Structure Level 2 disk volumes, and to report errors and inconsistencies. Also used to repair these inconsistencies.
Audit Analysis utility (ANALYZE /AUDIT)	To produce reports and summaries of security events from the system security audit log file. Use this utility to interpret the large amounts of auditing information that the system might generate.
Authorize utility (AUTHORIZE)	To add and modify records in the existing user authorization and network authorization files, or to create new files. Also used to add and modify records in the rights database.
Backup utility (BACKUP)	To copy or save files and disk volumes. Also used to restore saved files and volumes.
Bad Block Locator utility (BAD)	To analyze block-addressable devices and record the location of blocks that cannot reliably store data.
†Crash Logger Utility Extractor (CLUE)	On VAX systems, to obtain information about crash dumps. ‡On Alpha systems, some commands in the System Dump Analyzer facility (SDA) contain CLUE functionality.
‡DECevent Event Management utility	On Alpha systems, to produce ASCII reports derived from entries in system event log files.
Error Log utility (ERROR LOG)	To report the contents of a system error log file.
Exchange utility (EXCHANGE)	To transfer data to and from mass storage volumes that are written in formats other than standard formats recognized by the operating system.
Help Message utility (MSGHLP)	To quickly access information about system messages returned by DCL commands.
Install utility (INSTALL)	To improve performance or enhance privileges of images.
LAT Control Program utility (LATCP)	To set up and control the LAT software on OpenVMS host systems. LAT software allows you to connect terminals and printers to multiple remote systems.
†Local Area Disk Control Program utility (LADCP)	On VAX systems, to set up and control the local area disk (LAD) software on OpenVMS host systems. Use LAD software with InfoServer systems.
Log Manager Control Program utility (LMCP)	To create and manage the transaction logs used by DECdtm services.
Monitor utility (MONITOR)	To monitor systemwide performance.
Mount utility (MOUNT)	To make a disk or magnetic tape volume available for processing.
Network Control Program (NCP)	To set up, control, monitor, and test a DECnet network.
Operator Communication Manager (OPCOM)	To communicate with system users.
System Generation utility (SYSGEN)	To create and install page, swap, and dump files and to manage system parameters. †On VAX systems, to load and connect device drivers.

†VAX specific
‡Alpha specific

(continued on next page)

Table 2–2 (Cont.) System Management Utilities

Utility	Purpose
System Management utility (SYSMAN)	To centralize system management. Allows you to perform system management tasks simultaneously on one or more nodes. ‡On Alpha systems, to load and connect device drivers.

‡Alpha specific

This manual does not describe the following utilities in detail:

Utility	For More Information
Bad Block Locator utility (BAD)	*OpenVMS Bad Block Locator Utility Manual*, Online help
Exchange utility (EXCHANGE)	*OpenVMS Exchange Utility Manual*, Online help
LASTCP (LASTCP) and LADCP utilities (LADCP)	*InfoServer Client for OpenVMS LASTCP and LADCP Utilities Manual*
Network Control Program utility (NCP)	*DECnet for OpenVMS Network Management Utilities*, Online help

2.1.6 MGRMENU.COM Command Procedure

To help you perform basic system management tasks, Digital provides a command procedure named SYS$EXAMPLES:MGRMENU.COM. This procedure displays a menu that you can use to perform the following tasks:

- Add a user account
- Build a standalone BACKUP kit
- Shut down your system

You can use this command procedure as is, or modify it to serve your own site-specific needs. If you modify this procedure, Digital recommends you first copy the procedure to another directory (for example, SYS$MANAGER), so that an original version of MGRMENU.COM is always available in the SYS$EXAMPLES directory.

To see and use the menu, enter the following command:

```
$ @SYS$EXAMPLES:MGRMENU
```

2.2 Logging In to the SYSTEM Account

To use system management utilities and perform system management functions, log in to the system manager's account (SYSTEM).

Caution

Digital recommends that you change the password for the SYSTEM account frequently to maintain system security. Because the SYSTEM account has full privileges by default, exercise caution when using it.

If your site has strong security requirements, Digital recommends that you disable all but batch use of the SYSTEM account and set up separate privileged accounts for individuals who must perform privileged activities

on the system. This will allow you to more closely account for privileged activity on the system.

How To Perform This Task

1. Press the Return key on the console terminal.
2. In response to the system's request for your *username*, enter SYSTEM.
3. In response to the system's request for your *password*, enter the password that you chose for the SYSTEM account when you installed or upgraded the operating system, or the current password if you changed it since then.
4. After you enter your password, the system prints a welcome message on the console terminal. If you have logged in previously, the system also prints the time of your last login. When the dollar sign ($) prompt appears, login is complete and you can enter commands.

Example

VAX

On VAX systems:

```
Username: SYSTEM
Password:
        Welcome to OpenVMS VAX Version n.n on node x
    Last interactive login on Thursday, 20-FEB-1995 16:41
    Last non-interactive login on Friday, 21-FEB-1995 17:06 ♦
```

Alpha

On Alpha systems:

```
Username: SYSTEM
Password:
        Welcome to OpenVMS Alpha (TM) Operating System, Version n.n on node x
    Last interactive login on Thursday, 20-FEB-1995 16:41
    Last non-interactive login on Friday, 21-FEB-1995 17:06 ♦
```

2.3 Using the System Management Utility (SYSMAN) to Centralize System Management

If you manage more than one computer, you can use the System Management utility to centralize system management.

The following table lists some major SYSMAN features and points to sections in this chapter that contain more information.

Feature	For More Information
Enable a system to execute SYSMAN commands from remote nodes	Section 2.3.2
Define your SYSMAN management environment	Section 2.3.4
Adjust your SYSMAN profile to set privileges, default device and directory, and DCL verification	Section 2.3.6
Execute DCL commands from SYSMAN	Section 2.3.8
Create SYSMAN command procedures	Section 2.3.9
Set up SYSMAN with an initialization file	Section 2.3.10

2.3.1 Understanding SYSMAN

The System Management utility (SYSMAN) centralizes system management, so that you can manage nodes or VMSclusters from one location. Rather than logging in to individual nodes and repeating a set of management tasks, SYSMAN enables you to define your management environment to be a particular node, a group of nodes, or a VMScluster environment. With a management environment defined, you can perform traditional system management tasks from your local node; SYSMAN executes these tasks on all nodes in the target environment.

2.3.1.1 Privileges Required

You must have the following to run SYSMAN:

- OPER and TMPMBX privileges
- A separate account with no more than 125 rights, or enough identifiers removed from the current account so the total number of rights falls within the appropriate range.

  The rights limitation of 125 includes a minimum of three identifiers that are granted during login when the process rights list is created:

  - A UIC identifier
  - A system identifier
  - At least one environmental identifier, depending upon the environment in which the process is operating

2.3.1.2 Tools and Commands

SYSMAN uses many of the same software tools that you traditionally use to manage a system. It can process most DCL commands, such as MOUNT and INITIALIZE. It can also execute many system management utilities and command procedures, such as AUTHORIZE and AUTOGEN.

SYSMAN also includes its own commands that let you perform the following tasks:

Command	Task	For More Information
ALF (automatic login facility)	Associate a terminal or port with a user name	Section 6.9.1
CONFIGURATION	Inspect or modify VMScluster parameters	Section 20.4
DISKQUOTA	Control and monitor disk usage	Section 8.11.2
‡IO	Control and display the I/O configuration of an Alpha system	Section 7.4.2
LICENSE	Load and unload licenses	Section 3.3.2
PARAMETERS	Inspect and modify system parameters	Section 14.7
STARTUP	Customize startup databases by inspecting and modifying software startup components	Section 5.4

‡Alpha specific

2.3.2 Enabling a Remote System to Execute SYSMAN Commands

The SMISERVER process must be running on a remote node for SYSMAN commands to execute on that node. SMISERVER is the detached process responsible for executing SYSMAN commands on remote nodes.

Any node that is part of a VMScluster normally starts the SMISERVER process in the system startup procedure SYS$SYSTEM:STARTUP.COM. (The system parameter CLUSTER on the node must have a value of 1 or more.)

To start the SMISERVER process on a workstation that is not part of a VMScluster, include the following command line in the site-specific startup command procedure SYSTARTUP_VMS.COM:

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

For more information about SYSTARTUP_VMS.COM, see Section 5.2.7.

You can also enter this command interactively to restart the SMISERVER process without rebooting the system.

2.3.3 Understanding a SYSMAN Management Environment

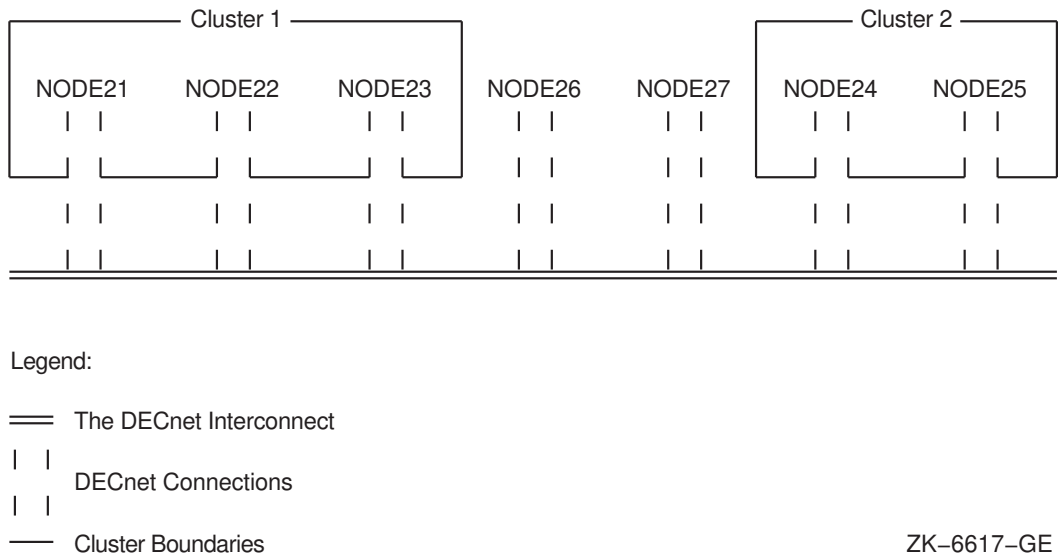
When you use SYSMAN, it is important to define the management environment you will be working in. The **management environment** is the node or nodes on which subsequent commands will execute.

By default, the management environment is the **local node** (the node from which you execute SYSMAN). To execute commands on one or more other nodes, you can redefine the management environment to be any of the following:

- Your own VMScluster
- A subset of nodes in your VMScluster
- A nonclustered node available through DECnet for OpenVMS
- Another VMScluster
- A subset of nodes in another VMScluster
- Any group of individual nodes

Refer to Figure 2–2 during the following discussion of management environments.

Figure 2–2 Sample SYSMAN Management Environment



You can use NODE21 as the management environment, or you can define the environment to be any node, group of nodes, or VMScluster shown in Figure 2–2.

If you execute SYSMAN from NODE21, then NODE21 is the local node; it is the management environment when SYSMAN starts. All other nodes are remote nodes.

2.3.4 Defining the SYSMAN Management Environment

To define the management environment, use the SYSMAN command SET ENVIRONMENT. Whenever you redefine an environment, SYSMAN displays the new context. You can always verify the current environment with the command SHOW ENVIRONMENT.

When you are not working on your local node or within your own cluster, your environment is a **nonlocal environment**. SYSMAN makes this distinction for security reasons; when you are defining a nonlocal environment, such as a different VMScluster, SYSMAN prompts for a password. SYSMAN also prompts for a password when you attempt to manage a system under a different user name. You can change your user name by including the /USERNAME qualifier on the SET ENVIRONMENT command.

A SYSMAN environment remains in effect until you change it or exit from SYSMAN.

2.3.4.1 Defining Another Node as the Environment

You can define a management environment to be any node available through DECnet. To define one or more nodes to be your management environment, use the SET ENVIRONMENT/NODE command. For example, the following SET ENVIRONMENT command defines the management environment to be NODE22:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/NODE=NODE22
%SYSMAN-I-ENV, current command environment:
        Individual nodes: NODE22
        Username ALEXIS will be used on nonlocal nodes
```

The following command defines the management environment to be a group of nodes — NODE23, NODE24, and NODE25:

```
SYSMAN> SET ENVIRONMENT/NODE=(NODE23,NODE24,NODE25)
Remote Password:

%SYSMAN-I-ENV, Current Command Environment:
        Individual nodes: NODE23,NODE24,NODE25
        At least one node is not in local cluster
        Username ALEXIS   will be used on nonlocal nodes
```

2.3.4.2 Using Logical Names to Organize Management Environments

If you want to organize the nodes in your VMScluster according to specific categories (for example, all CI-based nodes or all nodes with C installed), you can define logical names to use with the SET ENVIRONMENT/NODE command, as follows:

1. Create the logical name table SYSMAN$NODE_TABLE by putting the following command into the file SYS$MANAGER:SYLOGICALS.COM, which is executed during system startup:

   ```
   $ CREATE/NAME_TABLE/PARENT=LNM$SYSTEM_DIRECTORY SYSMAN$NODE_TABLE
   ```

2. Define one or more logical names to be a node or list of nodes by putting a command similar to the following into SYS$MANAGER:SYLOGICALS.COM:

   ```
   $ DEFINE CI_NODES NODE21,NODE22,NODE23/TABLE=SYSMAN$NODE_TABLE
   ```

3. When you set your SYSMAN environment from the DCL level, specify one of the logical names you created for this purpose. For example:

   ```
   $ RUN SYS$SYSTEM:SYSMAN
   SYSMAN> SET ENVIRONMENT/NODE=(CI_NODES)

   Remote Password:

   %SYSMAN-I-ENV, current command environment:
           Individual nodes: NODE21,NODE22,NODE23
           At least one node is not in the local cluster.
           Username SYSTEM       will be used on nonlocal nodes.
   ```

You can also define logical names for VAX and Alpha nodes in a dual-architecture VMScluster, as explained in Section 20.6.

Example

The following example demonstrates how you can define multiple logical names to organize several management environments:

```
$ CREATE/NAME_TABLE/PARENT=LNM$SYSTEM_DIRECTORY SYSMAN$NODE_TABLE
$ DEFINE CI_NODES SYS2,SYS8/TABLE=SYSMAN$NODE_TABLE
$ DEFINE C NODE21,NODE22,NODE23/TABLE=SYSMAN$NODE_TABLE
$ DEFINE PASCAL NODE23,NODE18,CI_NODES/TABLE=SYSMAN$NODE_TABLE
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/NODE=(C,PASCAL)
Remote Password:

%SYSMAN-I-ENV, current command environment:
        Individual nodes: NODE21,NODE22,NODE23,NODE18,SYS2,SYS8
        At least one node is not in the local cluster.
        Username SYSTEM       will be used on nonlocal nodes.
```

2.3.4.3 Defining a VMScluster Environment

To define your management environment to be a VMScluster, use the SET ENVIRONMENT/CLUSTER command.

In SYSMAN, VMScluster environments can be one of two types:

VMScluster Environment	Definition
Local	VMScluster from which you are using SYSMAN
Nonlocal	Any VMScluster other than the one from which you are executing SYSMAN

To expand the management environment in Figure 2–2 from NODE21 to Cluster 1, enter the following command from NODE21:

```
SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, Current Command Environment:
        Clusterwide on local cluster
        Username ALEXIS   will be used on nonlocal nodes
```

In the VMScluster environment shown in Figure 2–2, SYSMAN executes commands on all nodes in Cluster 1, namely NODE21, NODE22, and NODE23.

To manage a nonlocal VMScluster with SYSMAN, use the /NODE qualifier to identify the cluster. If you define a VMScluster alias, the /NODE qualifier can use the alias rather than the node name.

If you use the /CLUSTER and /NODE qualifiers together, the environment becomes the VMScluster where the given node is a member. For example, to perform management tasks on Cluster 2 in Figure 2–2, enter the SET ENVIRONMENT command with the /CLUSTER qualifier and name one node within Cluster 2 using the /NODE qualifier:

```
SYSMAN> SET ENVIRONMENT/CLUSTER/NODE=NODE24
Remote Password:

%SYSMAN-I-ENV, Current Command Environment:
        Clusterwide on remote node NODE24
        Username ALEXIS   will be used on nonlocal nodes
```

For information about using SYSMAN to manage a VMScluster that contains both Alpha and VAX nodes, see Section 20.6.

2.3.5 Understanding Your SYSMAN Profile

When you use SYSMAN across VMSclusters, SYSMAN establishes a profile that contains your rights, privileges, and defaults, and verifies that you are an authorized user. If you encounter privilege problems when using SYSMAN, it helps to know how SYSMAN determines your profile.

SYSMAN looks for three possible scenarios when determining your profile:

- If the environment is a VMScluster that has common SYSUAF and RIGHTSLIST databases, SYSMAN assigns the profile in effect on the local node to the SMISERVER process on the target node(s). This profile includes both authorized and current privileges.
- If the environment is a VMScluster and does not have common SYSUAF and RIGHTSLIST databases, SYSMAN checks the SYSUAF on the target node(s) to see if you are an authorized user. If you are an authorized user, SYSMAN copies your profile from the SYSUAF on the target node(s) to the SMISERVER process on the target node(s).
- If the environment has nodes that are not part of your local VMScluster, or if you have recently changed your user name, SYSMAN prompts you for a password before it checks the SYSUAF on the target node. If you enter the correct password and the SYSUAF shows that you are an authorized user, SYSMAN copies your profile from the SYSUAF on the target node(s) to the SMISERVER process on the target node(s).

The profile does not include symbolic names, logical names, preset terminal characteristics, or key definitions established through a login command procedure. The only environment that has the attributes defined in a login command procedure is the local node from which you are executing SYSMAN.

2.3.6 Adjusting Your SYSMAN Profile

Use the SYSMAN command SET PROFILE to change your SYSMAN management profile. The qualifiers /PRIVILEGES, /DEFAULT, and /VERIFY enable you to change the following attributes of the SMISERVER process:

Attribute	Qualifier	For More Information
Current privileges	/PRIVILEGES	Section 2.3.6.1
Default device and directory	/DEFAULT	Section 2.3.6.2
DCL verification of DO commands	/VERIFY	Section 2.3.7

This profile is in effect until you change it with the SET PROFILE command, reset the environment (which may change your profile automatically), or exit from SYSMAN.

The SET PROFILE command temporarily changes the attributes of your current local process. However, when you exit from SYSMAN, all attributes are restored to the values that were current when SYSMAN was invoked.

2.3.6.1 Changing Your Current Privileges

The SYSMAN command SET PROFILE/PRIVILEGES temporarily changes your current privileges in an environment.

Frequently, system management commands require special privileges. You might need to add privileges before you execute certain commands in an environment. System managers usually have the same privileges on all nodes; if you do not have the required privileges on a node, SYSMAN cannot execute the command and returns an error message.

Example

The following example makes SYSPRV one of your current privileges:

```
SYSMAN> SET PROFILE/PRIVILEGES=SYSPRV
SYSMAN> SHOW PROFILE
%SYSMAN-I-DEFDIR, Default directory on node NODE21  -- WORK1:[MAEW]
%SYSMAN-I-DEFPRIV, Process privileges on node NODE21 --
        TMPMBX
        OPER
        NETMBX
        SYSPRV
```

2.3.6.2 Changing Your Default Device and Directory

Use the SET PROFILE/DEFAULT command to reset the default device and directory specification for your process and all server processes in the environment.

Most often, the default device and directory specified in your UAF record is a first-level directory in which you create and maintain files and subdirectories. SYSMAN uses this default device and directory name when resolving file specifications. It also assigns the default device and directory name to any files that you create during a session.

In some cases, you might need to change the default device and directory in your SYSMAN profile. For example, you might have a directory containing command procedures as well as some system management utilities that require the default directory to be SYS$SYSTEM.

Example

The following example sets the default device and directory to DMA1:[SMITH.COM]:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET PROFILE/DEFAULT=DMA1:[SMITH.COM]
```

2.3.7 Setting DCL Verification

Using the SET PROFILE/VERIFY command, you can turn on DCL verification, which displays DCL command lines and data lines as they execute.

SYSMAN can execute DCL commands using the DO command. By default, SYSMAN DCL verification is turned off.

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET PROFILE/VERIFY
```

2.3.8 Executing DCL Commands from SYSMAN

The SYSMAN command DO executes DCL commands, command procedures, and SYSMAN commands on all nodes in a VMScluster environment. In VMSclusters or in any environment with multiple nodes, you enter a set of commands once, and SYSMAN executes the commands sequentially on every node in the environment. SYSMAN displays the name of each node as it executes commands, or an error message if the command fails.

If a node does not respond within a given timeout period, SYSMAN displays a message before proceeding to the next node in the environment. You can specify a timeout period with the SET TIMEOUT command.

Each DO command executes as an independent subprocess, so no process context is retained between DO commands. For this reason, you must express all DCL commands in a single command string, and you cannot run a procedure that requires input.

In a VMScluster environment, SYSMAN executes DO commands sequentially on all nodes in the VMScluster. After a command completes or times out on one node, SYSMAN sends it to the next node in the environment. Any node that is unable to execute a command returns an error message.

For more information about using the DO command to manage a VMScluster, see Section 20.6. You can also refer to the *OpenVMS System Management Utilities Reference Manual* for a complete description of the DO command.

Example

In the following example, SYSMAN runs the INSTALL utility and makes a file known on all nodes in the VMScluster when you enter the commands from the local node:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> SET PROFILE/PRIVILEGE=CMKRNL
SYSMAN> DO INSTALL ADD/OPEN/SHARED WORK4:[CENTRAL]STATSHR
.
.
.
%SYSMAN-I-OUTPUT, Command execution on node NODE21
%SYSMAN-I-OUTPUT, Command execution on node NODE22
```

2.3.9 Creating SYSMAN Command Procedures

The SYSMAN command @ executes SYSMAN command procedures on each node in the environment.

Example

The following example creates and executes a SYSMAN command procedure to display the current date and system time for each VMScluster node:

```
$ CREATE TIME.COM
SET ENVIRONMENT/CLUSTER
CONFIGURATION SHOW TIME Ctrl/Z
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> @TIME
%SYSMAN-I-ENV, Current command environment:
            Clusterwide on local cluster
            Username SYSTEM   will be used on nonlocal nodes
```

```
System time on node NODE21: 19-JUN-1995 13:32:19.45
System time on node NODE22: 19-JUN-1995 13:32:27.79
System time on node NODE23: 19-JUN-1995 13:32:58.66
SYSMAN>
```

2.3.10 Setting Up SYSMAN with an Initialization File

You can create an initialization file that is used each time you invoke SYSMAN. In the initialization file, you can perform tasks such as defining keys and setting up your environment.

The default file specification for the SYSMAN initialization file is SYS$LOGIN:SYSMANINI.INI. If you want your SYSMAN initialization file to have a different file specification, you must define the logical name SYSMANINI to point to the location of the file. The following is a sample initialization file that defines several keys:

```
$ TYPE SYSMANINI.INI
DEFINE/KEY/TERMINATE KP0 "SET ENVIRONMENT/CLUSTER/NODE=(NODE21,NODE22)"
DEFINE/KEY/TERMINATE KP1 "CONFIGURATION SHOW TIME"
DEFINE/KEY/TERMINATE KP2 "SHOW PROFILE"
   .
   .
   .
```

2.4 Using OPCOM to Communicate with System Users

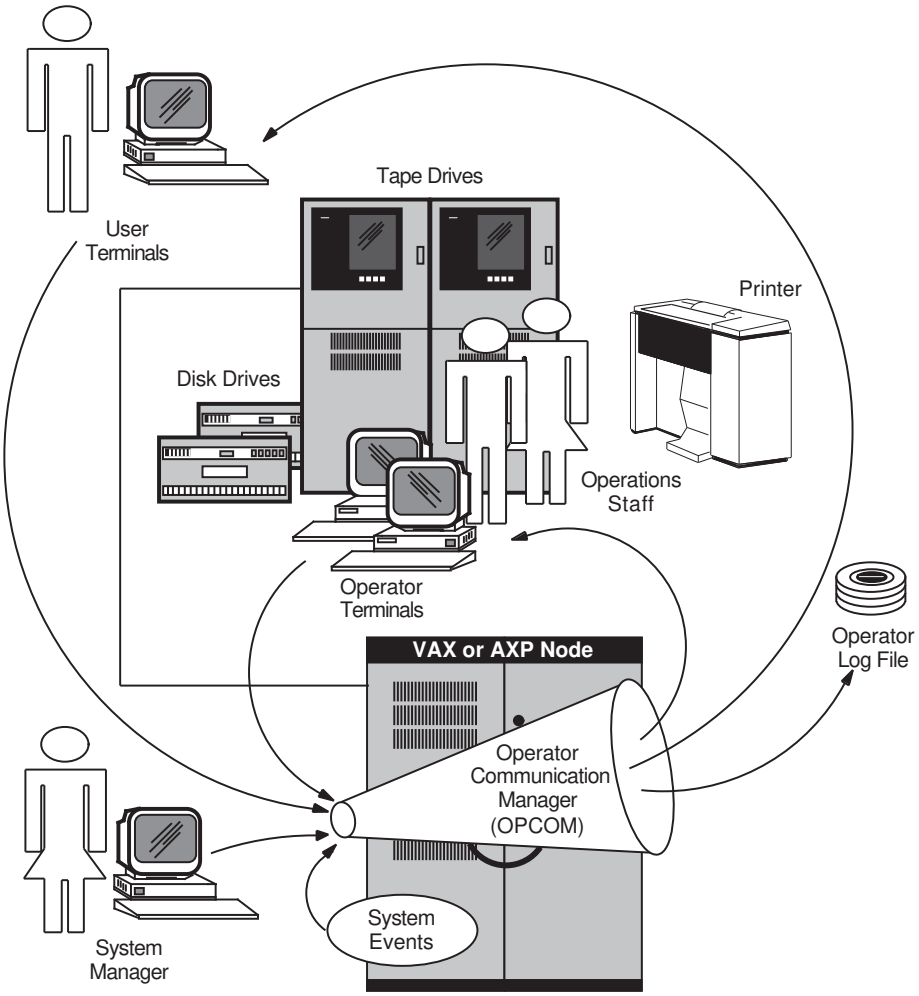
The operator communication manager (OPCOM) is a tool for communicating with users and operators on the system. OPCOM allows you to perform the following functions.

Function	For More Information
To broadcast messages to users who are logged in	Section 2.4.2
To control the use of OPA0: as an operator terminal	Section 2.4.3
To designate terminals as operator terminals, enabling them to display messages broadcast by OPCOM	Section 2.4.4
To record messages broadcast by OPCOM in a log file	Section 18.6.3
To send requests to an operator[1]	Section 2.4.5
To reply to operator requests[1]	Section 2.4.6

[1]These functions are used in sites where operators are assigned to help users mount disk or tape volumes and printer forms

Figure 2–3 illustrates the function of OPCOM.

Figure 2–3 Operator Communication Manager (OPCOM)



OPCOM Components

OPCOM uses the following components:

Component	Description	For More Information
OPCOM process	The system process that manages OPCOM operations. Unless you disable it, the OPCOM process starts automatically at system startup time.	Section 2.4.1
Operator terminals	Terminals designated to display messages broadcast by OPCOM. Usually, the console terminal (with the device name OPA0:) is the operator terminal. However, you can designate any user terminal as an operator terminal.	Section 2.4.4
Operator log file	A file that records messages broadcast by OPCOM. The file is named SYS$MANAGER:OPERATOR.LOG.	Section 18.6
OPCOM messages	Messages broadcast by OPCOM. These messages are displayed on operator terminals and written to the operator log file. The messages might be general messages sent by you, user requests, operator replies, or system events.	Section 18.6.2
REPLY and REQUEST commands	DCL commands that allow you to use and control OPCOM.	Section 2.4.2, Section 2.4.5, and Section 2.4.6

OPCOM Defaults

OPCOM uses the following defaults:

- OPCOM is started by default on all systems.
- Except for workstations in a VMScluster environment, OPCOM logs messages to OPA0:, which is enabled by default as an operator terminal. The log file SYS$MANAGER:OPERATOR.LOG is opened, and all OPCOM classes are enabled on both the operator terminal and the log file.

  Section 2.4.3 explains how to control the use of OPA0: as an operator terminal. Section 18.6.3.2 explains how to specify the default state of operator log files.

OPCOM Requirements

OPCOM has the following requirements:

- To execute an OPCOM REPLY command, OPCOM must be running, and you must enter the REPLY command from a terminal device designated as an operator terminal.
- The OPCOM REPLY command requires at least OPER privilege. You must have SHARE privilege if another process is logged in to the designated operator terminal. To enable or disable the security class, you must have SECURITY privilege.
- To designate an operator terminal in batch or SYSTARTUP, you must assign SYS$COMMAND to a valid terminal device.

2.4.1 Starting OPCOM

The OPCOM process must be running to execute the OPCOM REPLY command. The OPCOM process starts automatically during system startup, unless it is disabled. You might need to start OPCOM interactively if a software problem causes the process to fail and prevents OPCOM from restarting automatically.

To start OPCOM, enter the following command from the system manager's account (SYSTEM):

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

If a software problem causes OPCOM to fail, submit a software problem report (SPR) to report the problem to Digital. Include the process dump file named SYS$SYSTEM:OPCOM.DMP. When OPCOM fails, it creates this file to allow Digital representatives to determine the cause of the failure.

2.4.2 Sending Messages to Users

To broadcast a message to users, enter the REPLY command as follows:

REPLY [/qualifier...] ["message-text"]

For example:

```
$ REPLY/ALL/BELL "Please log off"
```

Use the following qualifiers to control OPCOM messages:

Qualifier	Description
/ALL	Broadcasts a message to all terminals that are attached to the system or VMScluster. These terminals must be turned on and have broadcast-message reception enabled.
/BELL	Rings a bell at the terminal receiving a message when entered with the /ALL, the /TERMINAL, or the /USERNAME qualifier; two bells when entered with the /URGENT qualifier; and three bells when entered with the /SHUTDOWN qualifier.
/NODE[=(*node-name*[,...])	Broadcasts a message to the local node only, or to a node or nodes you specify.
/SHUTDOWN	Sends a message beginning ''SHUTDOWN...''; if used with the /BELL qualifier, rings three bells at terminals receiving the message.
/TERMINAL=(*terminal-name*[,..])	Broadcasts the message to the specified terminals.
/URGENT	Broadcasts a message beginning ''URGENT...''; if used with the /BELL qualifier, rings two bells at terminals receiving the message.
/USERNAME=(*username*[,...])	Broadcasts a message to all terminals at which users are logged in to the system (or VMScluster), or only to the terminals of the specified users.

For more information, see the *OpenVMS DCL Dictionary*.

Examples

The REPLY command in the following example sends a message to all users logged in to node WLDWND. When the message is displayed, a bell rings at the terminal.

```
$ REPLY/ALL/BELL/NODE=WLDWND "Please log off"
```

The REPLY command in the following example sends a message to the user logged in at terminal TTC1. When the message is displayed, a bell rings at that terminal.

```
$ REPLY/BELL/TERMINAL=TTC1: "Your file has completed printing"
```

2.4.3 Controlling the Use of OPA0: as an Operator Terminal

You can control the use of OPA0: as an operator terminal, whether or not the node is part of a VMScluster system, by defining the following logicals in SYS$MANAGER:SYLOGICALS.COM:

Logical Name	Function
OPC$OPA0_ENABLE	Defined as True or False; if True, specifies that OPA0: is to be enabled as an operator terminal.
OPC$OPA0_CLASSES	Specifies the operator classes that are enabled for OPA0. The logical name can be a search list of the allowed classes, a comma-separated list, or a combination of the two.

The logicals take effect the next time you boot the system.

2.4.4 Designating Operator Terminals

Normally, the console terminal (with the device name OPA0:) is automatically an operator terminal except for workstations in a VMScluster environment. However, you can designate any terminal as an operator terminal. You can also disable a previously designated operator terminal.

Enabling Operator Terminals

To designate a terminal as an operator terminal, enter the REPLY/ENABLE command at the terminal. For example:

```
$ REPLY/ENABLE
$
%%%%%%%%%%  OPCOM  13-JUL-1995 11:30:30.56  %%%%%%%%%%
Operator _BHAK$FTA20: has been enabled, username SYSTEM
```

The REPLY/ENABLE command requires OPER privilege.

To designate an operator's terminal in batch or in startup command procedures, SYS$COMMAND must be assigned to a valid terminal device.

If your facility is very large, there may be several operators, each of whom is assigned to specific tasks. If this is the case, you can specify the classes of messages the operator terminal receives and responds to when you enable the operator terminal, as follows:

REPLY/ENABLE=(class[,...])

The following table describes each class of messages:

Class	Description
CARDS	Displays messages sent to the card readers.
CENTRAL	Displays messages sent to the central system operator.
CLUSTER	Displays messages from the connection manager pertaining to VMScluster state changes.
DEVICES	Displays messages pertaining to mounting disks.
LICENSE	Displays messages pertaining to licenses.
NETWORK	Displays messages pertaining to networks; the keyword CENTRAL must also be specified to inhibit network messages.
OPER1 to OPER12	Displays messages sent to operators identified as OPER1 to OPER12.
PRINTER	Displays messages pertaining to print requests.
SECURITY	Allows messages pertaining to security events; requires SECURITY privilege.
TAPES	Allows messages pertaining to mounting and dismounting tape volumes.

For example:

```
$ REPLY/ENABLE=(PRINTER,OPER3)
```

Disabling Operator Terminals

A terminal that you designate as an operator's terminal remains enabled even when the operator logs out. To return the terminal to normal (nonoperator) status, enter the REPLY/DISABLE command from the terminal.

Example

The following example designates terminal TTA3 as an operator terminal, enabling it to receive messages concerning printers, magnetic tapes and disks, and messages intended for the central operator. Later, it relinquishes terminal TTA3's ability to receive messages concerning tapes. The terminal still receives and can respond to messages about disks and printers and messages directed to CENTRAL.

```
$ REPLY/ENABLE=(PRINTER,DISKS,TAPES,CENTRAL)
$
%%%%%%%%%%  OPCOM  13-JUL-1995 11:37:09.52  %%%%%%%%%%
Operator TTA3 has been enabled, username SYSTEM

$
%%%%%%%%%%  OPCOM  13-JUL-1995 11:37:09.53  %%%%%%%%%%
Operator status for operator TTA3
CENTRAL, PRINTER, DISKS, TAPES

$ REPLY/DISABLE=TAPES
%%%%%%%%%%  OPCOM  13-JUL-1995 11:37:09.53  %%%%%%%%%%
Operator status for operator TTA3
CENTRAL, PRINTER, DISKS
$ REPLY/DISABLE
%%%%%%%%%%  OPCOM  13-JUL-1995 11:38:50.68  %%%%%%%%%%
Operator TTA3 has been disabled, username SYSTEM
```

2.4.5 Sending Requests to an Operator

In sites where operators are assigned to assist users by mounting volumes and changing printer forms, users can communicate with operators by entering REQUEST commands and the following qualifiers:

Qualifier	Description
/REPLY	Sends a request and requests a reply to the message. Requests sent with this command are issued a unique identification number to which the operator sends the response. The user cannot enter any commands until the operator responds.
/TO=(*operator*[,...])	If your facility is very large, there may be several operators, each of whom has specific tasks. The /TO qualifier lets users send requests to a specific operator. Options are as follows: CARDS, CENTRAL, CLUSTER, DEVICES, DISKS, LICENSE, NETWORK, OPER1 to OPER12, PRINTER, SECURITY, TAPES.

The DCL commands MOUNT/ASSIST and BACKUP/ASSIST also request operator assistance. For more information, see the following sections:

- For MOUNT requests, see Section 8.5.3.
- For BACKUP requests, see Section 10.9.1.

Example

Assume an operator is monitoring an operator terminal enabled for the PRINTER class. The following PRINT command submits an output job that requires a special print form (/FORM=LETTER). The REQUEST command sends a message to the operator. After completing the request, the operator would send a reply, as explained in Section 2.4.6.

```
$ PRINT/COPIES=2/QUEUE=LQ_PRINT  REPORT.OUT/FORM=LETTER
Job REPORT (queue LQA1, entry 401) pending
$ REQUEST/REPLY/TO=PRINTER -
_$ "Have queued job 401 as FORM=LETTER;  can you print it?"
%OPCOM-S-OPRNOTIF, operator notified, waiting...10:42:16.10
%OPCOM-S-OPREPLY, AFTER 11:00
19-APR-1995 10:25:32.40, request 3 completed by operator OPA0
```

2.4.6 Replying to Operator Requests

In sites where operators are assigned to assist users by mounting volumes and changing printer stock, operators can reply to user requests using REPLY commands. Operators can use the following qualifiers:

Qualifier	Description
/ABORT=*id-number*	Replies to the request specified by the identification number and cancels the request.
/PENDING	Replies to the request specified by the identification number and prevents the user from entering other commands until the operator fulfills or aborts the request. The current terminal must be enabled as an operator terminal.
/STATUS	Reports which classes are enabled, and all outstanding user requests for the terminal from which this command was entered. The current terminal must be enabled as an operator terminal.

Qualifier	Description
/TO=*id-number*	Replies to the request specified by the identification number and completes the request. The current terminal must be enabled as an operator terminal. Note that you can also use a variation of the REPLY/TO command in response to a MOUNT/ASSIST and BACKUP /ASSIST commands. For more information, see Section 8.5 and Section 10.9.1.

An operator working with magnetic tapes would also use additional REPLY qualifiers specific to magnetic tape operations. For more information, see Section 8.8.2.4. For detailed information on the REPLY command and its qualifiers, see the *OpenVMS DCL Dictionary*.

Example

In the following example, the REPLY/TO command replies to operator request number 5, issued by user ROBINSON. The MOUNT device is switched to DUA4, and the user is notified.

```
%%%%%%%%%%  OPCOM, 19-APR-1995 10:20:50.39  %%%%%%%%%%
           request 5 from user ROBINSON
           Please mount volume GRAPHIC_FILES in device _DUA11:
           Shelf 4 - slot B
$ REPLY/TO=5 "SUBSTITUTE  DUA4"
```

2.5 Using VMSKITBLD.COM to Modify a System Disk

VAX

On VAX systems, the command procedure SYS$UPDATE:VMSKITBLD.COM allows you to duplicate system files from an existing system disk on another disk. (On Alpha systems, procedures similar to those in the VMSKITBLD.COM procedure are performed by the AXPVMS$PCSI_INSTALL.COM procedure. See Section 3.1.2 and also the *OpenVMS Alpha Version 7.0 Upgrade and Installation Manual*.)

The SYS$UPDATE:VMSKITBLD.COM procedure offers the following options:

Option	Description	For More Information
BUILD	Builds a new common system disk after destroying all existing files on the disk.	Section 2.5.1
COPY	Copies the operating system files to an existing disk without destroying nonsystem files that are currently on the disk.	Section 2.5.2
ADD	Adds a new system root directory to an existing system disk.	Section 2.5.3

VMSKITBLD uses two disks:

Disk	Description
Source disk	The disk from which you copy system files. The source disk must be an existing system disk.
Target disk	The disk to which you move the system files.

Caution

Do not attempt to use VMSKITBLD with the current system disk as the target disk. VMSKITBLD.COM deletes files that are required for a running system.

2.5.1 Using VMSKITBLD.COM to Build a New System Disk

At some point, you might want to create a new system disk. For example, suppose that your existing system disk is an RA81 disk. If you purchase a larger RA90 disk and want to use it as your system disk, you could use the VMSKITBLD BUILD option to build a new system disk on the RA90 disk.

The existing system disk is the source disk. The new disk is the target disk.

Caution

The VMSKITBLD BUILD option initializes the target disk, deleting all of its previous contents. For information on copying files to an existing system disk without destroying files, see Section 2.5.2.

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 as described in the following procedure.

How to Perform This Task

1. If the source disk is not the current booted system disk, boot the operating system from the source disk.
2. Log in to the SYSTEM account.
3. Make sure the disk is spun up and on line. If you are using a removable disk, you must also place the disk into the appropriate drive.
4. Enter the following command to invoke VMSKITBLD:

   ```
   $ @SYS$UPDATE:VMSKITBLD
   ```

   VMSKITBLD prompts you to choose one of the following options:

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

5. Enter BUILD and press the Return key.

   VMSKITBLD displays messages that either prompt you for information needed to complete the operation or inform you of the procedure's status.

   a. In response to the following prompt, enter the name of the source disk:

      ```
      * Enter mounted SOURCE disk name (ddcu:):
      ```

   b. In response to the following prompt, enter the top-level system directory for the source disk:

      ```
      * Enter SOURCE disk top-level system directory [default = SYS0]:
      ```

      In most cases, you can choose the default value [SYS0].

   c. In response to the following prompt, enter the name of the target disk:

      ```
      * Enter TARGET disk name (ddcu:):
      ```

d. In response to the following prompt, enter the volume label of the target disk:

```
* Enter the TARGET disk's label [default = VAXVMSRL5]:
```

e. In response to the following prompt, enter the top-level system directory:

```
* Enter TARGET disk top-level system directory [default = SYS0]:
```

In most cases, you can choose the default value [SYS0].

f. The procedure displays the following message to warn you that the target disk will be initialized and to allow you to stop the procedure:

```
    The target disk will be initialized.
* Target disk, _DUA0:, ready to be initialized? (Y/N): Y
```

Make sure it is safe to destroy the contents of the target disk, and enter Y to continue.

When the system displays the dollar sign ($) prompt, the system disk is built. VMSKITBLD automatically dismounts the target disk. At this point, the target disk contains all the operating system files required for a complete system.

6. Complete the system disk by creating a rights database and network proxy database and configuring the system with appropriate system parameters. For instructions, see Section 2.5.1.1.

7. To use the new system disk, reboot the system with the new system disk.

Example

The following example runs VMSKITBLD.COM to build a new system disk. It copies the files on the current system disk to create a new system disk on the DUA0: disk.

```
* Enter mounted SOURCE disk name (ddcu:): SYS$SYSDEVICE:
* Enter SOURCE disk top level system directory [default = SYS0]: Return
* Enter TARGET disk name (ddcu:): DUA0: Return
* Enter the TARGET disk's label [default = VAXVMSRL5]: Return
* Enter TARGET disk top level system directory [default = SYS0]: Return
    The target disk will be initialized.
* Target disk, _DUA0:, ready to be initialized? (Y/N): Y
    Target disk, _DUA0:, has been initialized.
%MOUNT-I-MOUNTED, VAXVMSRL5   mounted on _DUA0:
    Creating system specific directories ...
    Creating cluster common directories ...
    Creating SYSGEN files ...
%SYSGEN-I-CREATED, _DUA0:<SYS0.SYSEXE>SWAPFILE.SYS;1 created
%SYSGEN-I-CREATED, _DUA0:<SYS0.SYSEXE>PAGEFILE.SYS;1 created
%SYSGEN-I-CREATED, _DUA0:<SYS0.SYSEXE>SYSDUMP.DMP;1 created
    Copying files from source disk ...
    Copying DECwindows file from source disk ...
    Writing a boot block ...
    System disk complete.
$
```

2.5.1.1 Completing a System Disk Built with VMSKITBLD.COM

After you create a new system disk using the VMSKITBLD BUILD option, use the following procedure to complete the new system disk:

1. Boot the new system disk using a conversational boot. For instructions, see the upgrade and installation supplement for your computer.
2. When the SYSBOOT> prompt appears, enter the USE DEFAULT command to boot with default values for all system parameters.
3. Enter the CONTINUE command to continue booting.
4. After the system boots, log in to the SYSTEM account. The password for the system account will be the default password, MANAGER. Make sure you change this password.
5. Use the Authorize utility to create a rights database and a network proxy database. For more information, see the *OpenVMS Guide to System Security*.
6. Run AUTOGEN from the SAVPARAMS phase to set appropriate values for system parameters. Be sure to specify the CHECK_FEEDBACK option. See Section 14.5 and the AUTOGEN section of the *OpenVMS System Management Utilities Reference Manual* for detailed information on running AUTOGEN.

   To reboot from the former system disk, specify REBOOT as the end phase when invoking AUTOGEN.

   To reboot the system from the new system disk, specify SHUTDOWN as the end phase and reboot manually, specifying the new system disk.

Example

```
SYSBOOT> USE DEFAULT
SYSBOOT> CONTINUE
   .
   .
   .
$ SET DEFAULT SYS$COMMON:[SYSEXE]
$ RUN AUTHORIZE
UAF> CREATE/RIGHTS
UAF> CREATE/PROXY
UAF> EXIT
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT CHECK_FEEDBACK
   .
   .
   .
```

2.5.2 Using VMSKITBLD.COM to Copy System Files to an Existing Disk

You can use VMSKITBLD to copy the operating system files to a target disk without deleting the files already existing on the target disk. For example, if you accidentally delete a large number of system files from a system disk, you can use VMSKITBLD to copy the system files from another system disk.

To do this, the operating system must be running and the source disk that you intend to copy from must be mounted.

When you use the COPY option of VMSKITBLD.COM, the user-modified files (including SYSUAF.DAT and site-specific command files) are *not* copied from the source disk; VMSKITBLD uses the unaltered TEMPLATE versions of these files. In addition, the procedure does not create the system-specific files SWAPFILE.SYS, PAGEFILE.SYS, or SYSDUMP.DMP.

Before VMSKITBLD copies each new system file, it deletes the older version of the file from the target disk.

How to Perform This Task

1. Log in to the SYSTEM account.
2. Place the target disk into the appropriate drive.
3. Note the device name of the target disk.
4. Enter the following command to invoke VMSKITBLD:

   ```
   $ @SYS$UPDATE:VMSKITBLD
   ```

   VMSKITBLD prompts you to choose one of the following options:

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

5. Enter COPY and press the Return key.

   VMSKITBLD displays messages that either prompt you for information needed to complete the copy operation or inform you of the procedure's status.

   a. In response to the following prompt, enter the name of the source disk.

      ```
      * Enter mounted SOURCE disk name (ddcu:):
      ```

   b. In response to the following prompt, enter the top-level system directory for the source disk:

      ```
      * Enter SOURCE disk top level system directory [default = SYS0]:
      ```

      In most cases, you can choose the default value [SYS0].

   c. In response to the following prompt, enter the name of the target disk:

      ```
      * Enter TARGET disk name (ddcu:):
      ```

   d. In response to the following prompt, enter the top-level system directory:

      ```
      * Enter TARGET disk top level system directory [default = SYS0]:
      ```

      In most cases, you can choose the default value [SYS0].

   When the system displays the dollar sign ($) prompt, the files have been copied and the system disk is complete. VMSKITBLD automatically dismounts the target disk.

Example

```
* Enter mounted SOURCE disk name (ddcu:): SYS$SYSDEVICE:
* Enter SOURCE top level system directory [default = SYS0]: Return
* Enter TARGET disk name (ddcu:): DUA0: Return
* Enter TARGET disk top level system directory [default = SYS0]: Return
%DCL-I-ALLOC, _DUA0: allocated
%MOUNT-I-MOUNTED, VAXVMSRL5 mounted on _DUA0:
    Copying files from source disk ...
    Copying DECwindows files from source disk ...
    Writing a boot block ...
    System disk complete.
$
```

2.5.3 Using VMSKITBLD.COM to Add an Alternate System Root Directory

Use the ADD option to create an alternate system root directory on a target system disk. You might use this option to create a test environment where you can test software without interfering with the current version of the system.

The system disk that you are adding to cannot be in use.

Note

Do not use the ADD option to create a system root to add a new system to a VMScluster environment. Instead, use the SYS$MANAGER:CLUSTER_CONFIG.COM procedure.

The ADD option creates only new specific root directories. The current common directory is linked to the new root.

How to Perform This Task

1. Log in to the SYSTEM account.
2. Check the number of free blocks on the system disk to make sure you have adequate space for the new files, including SWAPFILE.SYS, PAGEFILE.SYS, and SYSDUMP.DMP. The sizes of these files are determined by the type of computer you use. For information on calculating size for page, swap, and dump files, see Section 15.4.
3. Make sure the target system disk is dismounted and on line.
4. Enter the following command to invoke VMSKITBLD:

   ```
   $ @SYS$UPDATE:VMSKITBLD
   ```

   VMSKITBLD prompts you to choose one of the following options:

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

5. Enter ADD and press the Return key.

   VMSKITBLD displays messages that either prompt you for information needed to complete the operation or inform you of the procedure's status.

   a. In response to the following prompt, enter SYS$SYSDEVICE and press Return:

      ```
      * Enter mounted SOURCE disk name (ddcu:):
      ```

   b. In response to the following prompt, press Return to choose the default:

      ```
      * Enter SOURCE disk top level system directory [default = SYS0]:
      ```

   c. In response to the following prompt, enter the name of the target disk:

      ```
      * Enter TARGET disk name (ddcu:):
      ```

   d. In response to the following prompt, enter the new root directory specification.

      ```
      * Enter TARGET disk top level system directory [default = SYS0]:
      ```

      Do not specify directories SYSE or SYSF:

      - SYSE is reserved for storing standalone BACKUP.
      - SYSF is reserved for Digital use.

When the system displays the dollar sign ($) prompt, the target system directory contains the new system root directory. VMSKITBLD automatically dismounts the target disk.

6. Configure the new system root by booting the target disk and running AUTOGEN. For instructions, see Section 2.5.3.1.

Example

The following example adds an alternate system root directory named SYSA on the target disk SHEMP$DUA5:

```
* Enter mounted SOURCE disk name (ddcu:): SYS$SYSDEVICE:
* Enter SOURCE top level system directory [default = SYS0]: RETURN
* Enter TARGET disk name (ddcu:): SHEMP$DUA5: RETURN
* Enter TARGET disk top level system directory [default = SYS0]: SYSA RETURN
%DCL-I-ALLOC, _SHEMP$DUA5: allocated
%MOUNT-I-MOUNTED, VAXVMSRL5   mounted on _SHEMP$DUA5:
    Creating system specific directories ...
    Creating SYSGEN files ...
%SYSGEN-I-CREATED, _SHEMP$DUA5:<SYSA.SYSEXE>SWAPFILE.SYS;1 created
%SYSGEN-I-CREATED, _SHEMP$DUA5:<SYSA.SYSEXE>PAGEFILE.SYS;1 created
%SYSGEN-I-CREATED, _SHEMP$DUA5:<SYSA.SYSEXE>SYSDUMP.DMP;1 created
    System disk complete.
$
```

2.5.3.1 Configuring a System Root Added with VMSKITBLD

After you use VMSKITBLD to add an alternate system root directory to a system disk, you must configure system parameters for the new root. Perform the following steps:

1. Shut down the system and halt your computer. For instructions on shutting down your system, see Section 4.8.1.
2. Perform a conversational boot, as described in the upgrade and installation supplement for your computer.
3. When the conversational boot prompt (SYSBOOT>) appears, enter the following commands:

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

4. After the system boots, log in to the SYSTEM account and execute AUTOGEN from the SAVPARAMS phase to set appropriate values for system parameters.

   To reboot from the former root, specify REBOOT as the end phase when invoking AUTOGEN.

   To reboot from the new root directory, specify SHUTDOWN as the AUTOGEN end phase, and reboot manually.

   See Section 14.5 and the *OpenVMS System Management Utilities Reference Manual* (AUTOGEN) for detailed information on AUTOGEN.

Example

```
SYSBOOT> USE DEFAULT
SYSBOOT> CONTINUE
   .
   .
   .
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT CHECK_FEEDBACK
   .
   .
   .◆
```

3
Installing, Upgrading, and Updating Software

This chapter describes the concepts related to installing, upgrading, and updating OpenVMS operating system software and layered products. Two methods are available for installing or upgrading software: the POLYCENTER Software Installation utility and the VMSINSTAL.COM command procedure.

The POLYCENTER Software Installation utility is a new method for installing and upgrading software, and offers the following features:

- Typically faster installation and upgrade operations
- Removal (deinstallation) of previously installed software products
- A database of information on installed products that has query capabilities on the database
- Two interfaces to the utility: DCL and Motif
- Dependency checking of software products based on product version number

Operating System Software

To install or upgrade the OpenVMS Alpha operating system, use the POLYCENTER Software Installation utility. To install or upgrade the OpenVMS VAX operating system, use the VMSINSTAL.COM command procedure.

Layered Product Software

To install or upgrade layered products, use either the POLYCENTER Software Installation utility or the VMSINSTAL.COM command procedure method. Each layered product is packaged to use one method or the other. See the layered product's documentation for information about which method to use.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Installing the OpenVMS operating system	Section 3.1
Installing layered product software	Section 3.2
Using VMSINSTAL.COM to install layered software	Section 3.3 through Section 3.6
Using the POLYCENTER Software Installation utility with the DCL interface	Section 3.7 through Section 3.10
Using the POLYCENTER Software Installation utility with the Motif interface	Section 3.11

This chapter explains the following concepts:

Concept	Section
VMSINSTAL.COM	Section 3.3
POLYCENTER Software Installation utility	Section 3.7
Differences between the DCL and Motif interfaces when using the POLYCENTER Software Installation utility	Section 3.7.2
The DECwindows Motif interface	Section 3.11

3.1 Installing, Upgrading, and Updating the Operating System

The following sections explain how to install, upgrade, and update the OpenVMS operating system. For additional information, refer also to the following:

- On VAX systems, see the most recent versions of the *OpenVMS VAX Upgrade and Installation Manual* and the upgrade and installation supplement for your VAX computer.
- On Alpha systems, see the most recent version of the *OpenVMS Alpha Upgrade and Installation Manual*.

Understanding Installation, Upgrade, and Update Procedures

The installation of software can mean a complete installation, an upgrade, or an update. The terms have the following meanings:

Installation	Used to install a completely new version of the operating system.
Upgrade	Used to make major changes to an existing version of the operating system.
Update	Used to make minor changes to an existing version of the operating system.

3.1.1 Installing or Upgrading the OpenVMS Alpha Operating System

Alpha

To install or upgrade the OpenVMS Alpha operating system, boot the OpenVMS Alpha operating system CD–ROM from the local drive drive or from a drive connected to an InfoServer system. After you enter the appropriate boot command, a command procedure displays a menu that lists the following options:

- Install or upgrade OpenVMS Alpha, using the POLYCENTER Software Installation utility
- List layered product kits that this procedure can install
- Install or upgrade layered products, using the POLYCENTER Software Installation utility
- Execute DCL commands and procedures, from which you can perform preinstallation or maintenance tasks such as mounting or showing devices and backing up or restoring files on the system disk
- Shut down this system

When you choose the install or upgrade OpenVMS Alpha option, the system asks whether you want to initialize or preserve the system disk. Depending on the option you choose, the POLYCENTER Software Installation utility installs or upgrades the OpenVMS Alpha operating system. For example:

- If you choose to initialize the system disk (INITIALIZE), the utility performs these operations:
  - Removes all software and data files that were previously on the target disk
  - Installs the OpenVMS Alpha operating system

Caution

Initializing the target disk erases its contents. For this reason, use the installation procedure only on new computers or if you are sure you want to replace the entire contents of the target disk with new system files.

- If you choose to preserve the system disk (PRESERVE) and the OpenVMS Alpha operating system is *not* already installed on the target disk, the utility performs these operations:
  - Installs the OpenVMS Alpha operating system
  - Retains, or preserves, all other contents of the target disk
- If you choose PRESERVE and the OpenVMS Alpha operating system *is* installed on the target disk, the utility *upgrades* your system, as follows:
  - Merges and replaces old operating system files and new files
  - Retains, or preserves, all other contents of the target disk

Caution

The upgrade procedure does not work correctly if you have changed the names of system directories on your system disk. Restore the names of the original system directories on the system disk before attempting an upgrade. (Check the DEC-AXPVMS-VMS-*-2.PCSI$DESCRIPTION file in SYS$SYSDEVICE:[VMS$COMMON] for original system directory names.)

For information about booting the OpenVMS Alpha operating system CD–ROM and choosing the INITIALIZE or PRESERVE options, see the current *OpenVMS Alpha Upgrade and Installation Manual*.♦

3.1.2 Installing or Upgrading from a Running Alpha System

Alpha

You can install or upgrade from a running OpenVMS Alpha system to a target system disk. To do this, use AXPVMS$PCSI_INSTALL.COM in SYS$SYSTEM. This procedure is the same installation procedure that installs or upgrades OpenVMS Alpha from the OpenVMS Alpha operating system CD–ROM. (The procedure is similar to using VMSKITBLD.COM, which is no longer supported on Alpha systems.) Privileges required by the procedure are:

BYPASS	OPER
CMKRNL	SYSPRV
LOG_IO	VOLPRO

The procedure attempts to turn on these privileges but if it cannot, it reports an error and exits. Your original privilege settings are restored.

Use the following command to run the installation or upgrade procedure:

```
$ @SYS$SYSTEM:AXPVMS$PCSI_INSTALL.COM
```

The operation of AXPVMS$PCSI_INSTALL.COM is documented in the current *OpenVMS Alpha Upgrade and Installation Manual*. Before you execute this command procedure, read the appropriate chapters in that manual. Refer to Table 3–1 and Table 3–2.

Table 3–1 If You Are Installing

Read chapter...	To find information about...
Preparing to Install	What to do before running AXPVMS$PCSI_ INSTALL.COM
Installing	How to use AXPVMS$PCSI_INSTALL.COM
After Installing	What to do after an installation

Table 3–2 If You Are Upgrading

Read chapter...	To find information about...
Preparing to Upgrade and Before Upgrading	What to do before running AXPVMS$PCSI_ INSTALL.COM
Upgrading	How to use AXPVMS$PCSI_INSTALL.COM
After Upgrading	What to do after an upgrade

3.1.3 Installing the OpenVMS VAX Operating System

VAX

When you install the OpenVMS VAX operating system, the installation procedure does the following:

- Initializes the system disk, erasing its contents
- Creates a system directory structure on the new system disk
- Transfers system files from the distribution media to the new system disk

Caution

The installation procedure initializes the target disk, erasing its contents. For this reason, use the installation procedure only on new computers or if you are sure you want to replace the entire contents of the target disk with new system files.

You use the installation procedure under the following conditions:

- If your VAX computer is new; that is, if it has never had any version of the operating system running on it.
- If your computer is running a version of the OpenVMS VAX operating system and you are sure you are willing to destroy the entire contents of the system disk, including both system and user files.
- If you are running the OpenVMS VAX operating system, but are not able to perform an upgrade. For example, if you do not have a standard version of the OpenVMS VAX operating system on your system disk, the upgrade procedure might not work correctly.

- If you want to run the OpenVMS VAX operating system in a C2 security environment.

To install the OpenVMS VAX operating system, follow the installation procedures in the current *OpenVMS VAX Upgrade and Installation Manual*.♦

3.1.4 Upgrading the OpenVMS VAX Operating System

VAX

When you upgrade the operating system, the upgrade procedure does the following:

- Makes room for the upgrade by purging and deleting most system files but leaves some system files and all user files intact. The upgrade also leaves most layered products intact. In some cases, reinstalling layered products after you upgrade may be necessary.
- Transfers the system files from the distribution media to the system disk.
- Merges old system libraries with new libraries.
- Cleans up files and directories used only during the upgrade.

In most cases, if you are already running a standard version of the operating system, you can use the upgrade procedure to obtain the next highest version. The upgrade procedure does not initialize the system disk.

Caution

The upgrade procedure does not work correctly if you have changed the names of system directories on your system disk. Restore the names of the original system directories on the system disk before attempting an upgrade. (Check the VMSKITBLD.DAT file in SYS$UPDATE for original system directory names.)

To perform an upgrade of the OpenVMS VAX operating system, follow the upgrade procedures in the current *OpenVMS VAX Upgrade and Installation Manual*.♦

3.1.5 Updating the OpenVMS Alpha or OpenVMS VAX Operating System

An update procedure makes minor changes to the operating system. When you update the operating system, the update procedure replaces some system files and applies patches to others (if necessary).

The two types of updates are:

- A required update you perform after installing or upgrading the operating system is referred to as a **mandatory update**. The directions for an installation or an upgrade indicate when to perform a mandatory update.
- Another type of update allows you to apply a special release of the operating system, which might include new features, support for new hardware, or refinements to specific software components. Instructions for applying such an update are included with the specific release.

Caution

The update procedure does not work correctly if you have changed the names of system directories on your system disk. Restore the names of system directories before attempting an

update. (On VAX systems, check the VMSKITBLD.DAT file in SYS$UPDATE for original system directory names. On Alpha systems, check the DEC-AXPVMS-VMS-*-2.PCSI$DESCRIPTION file in SYS$SYSDEVICE:[VMS$COMMON] for original system directory names.)

Follow the instructions provided with the update kit. The instructions will indicate whether to use VMSINSTAL.COM or the POLYCENTER Software Installation utility.

3.2 Installing or Upgrading Layered Products

To install and upgrade layered products, use either the software installation command procedure, SYS$UPDATE:VMSINSTAL.COM, or the POLYCENTER Software Installation utility. The installation documentation for each layered product states which procedure you should use.

Install or upgrade a layered product on your Alpha or VAX system as follows:

- If you are installing or upgrading a layered product using VMSINSTAL.COM, review the information in the following sections:

Task	Section
Preparing your system to run VMSINSTAL.COM	Section 3.3
Running VMSINSTAL.COM	Section 3.4
Recovering from a system failure	Section 3.5
Selecting VMSINSTAL.COM options	Section 3.6

Note that these sections do not describe specific VMSINSTAL.COM procedures. The examples used are for illustration only. For details of a particular product, refer to the installation documentation for the specific product.

- If you are installing or upgrading a layered product using the POLYCENTER Software Installation utility, refer to Section 3.7 and to the layered product installation documentation.

Task	Section
Using the POLYCENTER Software Installation utility	Section 3.7
Installing software	Section 3.8
Performing operations on installed software	Section 3.9
Removing installed software	Section 3.10
Using the DECwindows Motif interface	Section 3.11

3.3 Preparing Your System to Run VMSINSTAL.COM

This section provides guidelines for preparing your system for using VMSINSTAL.COM. Note that each software product that you install might not require you to follow all of the guidelines listed in this section.

3.3.1 Performing Preliminary Operations

Before you use VMSINSTAL.COM, perform the following operations (not necessarily in the order listed):

- Back up your system disk, as described in Section 10.17. Use the backup copy as a working copy for the installation.

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

- Log in to the SYSTEM account at the console terminal.

- Be sure all users have logged out and all batch jobs have completed by using, respectively, the SHOW USERS and SHOW SYSTEM/BATCH commands. Keep users off the system until VMSINSTAL.COM completes by using the following command:

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

- Shut down DECnet for OpenVMS. (Section 20.6.4 contains instructions for doing this.)

- Check system parameters. (GBLPAGES, GBLSECTIONS and NPAGEDYN often need to be adjusted.) Read the documentation supplied with each layered product to be installed, and find out if the product has any specific resource requirements.

  If you must change parameter values, increase the values by adding ADD_*parameter-name* symbols to MODPARAMS.DAT. (See Section 14.5.1.1.)

  Use AUTOGEN with feedback to size the system resources properly. (See Section 14.4.1.)

- Make sure the limits in the SYSTEM account authorization record are equal to or greater than the recommended limits.

  To check these limits, run the Authorize utility (AUTHORIZE) to display the current limits of the SYSTEM account's user authorization file.

  To run AUTHORIZE, enter the following commands:

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

  At the UAF prompt (UAF>), enter the following command:

  ```
  UAF> SHOW SYSTEM
  ```

  See Section 6.1.2 for details.

- If necessary, use the Authorize utility to modify the SYSTEM account limits. Changes you make do not take effect until you log out and log in again.

  For example, to increase the DIOLM limit to 100, enter the following command:

  ```
  UAF> MODIFY SYSTEM/DIOLM=100
  ```

  See Section 6.1.2 for details.

- Physically mount the first distribution media that contains the software product. See Section 8.5 for details.

- Register and load licenses, as explained in the next section.

3.3.2 Registering and Loading Licenses

A **license** refers to the authorization you have to use a product. The License Management Facility (LMF) enables you to register, manage, and track software licenses on line.

A **Product Authorization Key (PAK)** contains information that is provided for many Digital products. The data provided in the PAK allows you to register a software license in the license database on a system.

If you did not register and load your operating system license during the installation of the OpenVMS operating system, you must perform that task (and register other licenses, if necessary) before you install other software products, as explained in the following steps:

1. Log in to the system manager's account, SYSTEM.
2. Register the license in one of two ways:
   - Invoke the SYS$UPDATE:VMSLICENSE.COM procedure. When it prompts you for information, respond with data listed on your operating system Product Authorization Key (PAK). (You can register other software product licenses at this time as well.)
   - At the DCL prompt, enter the LICENSE REGISTER command with the appropriate qualifiers that correspond to License PAK information.
3. Use either of the following utilities to load a license:
   - License Management Facility (LMF), which loads the PAK only on the local node.
   - System Management utility (SYSMAN), which allows you to load the PAK throughout an entire cluster. (SYSMAN LICENSE commands are a subset of LMF commands.)

   Only the local node requires the installation of a license before you use VMSINSTAL.COM.

For more information about loading licenses, see the *OpenVMS License Management Utility Manual*.

3.3.3 Preventing Nodes from Sharing PAKs

The /NO_SHARE qualifier for the LICENSE MODIFY command lets you add the NO_SHARE option to a PAK registered in a license database (LDB). NO_SHARE PAKs are assigned to a single node in a VMScluster system. A NO_SHARE PAK cannot be shared with other VMScluster nodes.

The use of this qualifier remedies problems that occasionally occur when you attempt to use the PAK of a software product for which you already have other PAKs in your LDB. The PAK does not combine with the other PAKs for the same software product, resulting in LICENSE-W-NOCOMB warnings. Often, the license is not loaded on the nodes on which you want it loaded.

To remedy this problem, do the following:

1. Add the NO_SHARE option to the PAK or PAKs causing the NOCOMB warnings
2. Assign each PAK to a specific VMScluster node

3.4 Running VMSINSTAL.COM

Before you run VMSINSTAL.COM, note the following:

- Read the installation instructions for the specific product or update. If you need assistance during an installation, enter a question mark (?) for an explanation of possible responses.
- When you first start VMSINSTAL.COM, the procedure displays several prompts and messages that direct and explain the installation. These prompts and messages differ, depending on the software product that you are installing.
- When the procedure asks if you are satisfied with the backup of your system disk, back up your system disk before continuing with the installation if you do not have a recent backup of your system disk.
- If you do not satisfy all conditions required to start VMSINSTAL.COM, the procedure displays a warning message explaining the problem and asks you if you want to continue. Digital strongly recommends that you satisfy these conditions before you try to start VMSINSTAL.COM again.

Caution

If you continue the procedure without making the required corrections, Digital cannot guarantee that the installation will be supported.

How to Perform This Task

To run VMSINSTAL.COM, enter a command in the following format:

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

Example

```
$ @SYS$UPDATE:VMSINSTAL CALENDAR020 MUA0:
```

The command in this example installs the product CALENDAR, from save sets named CALENDAR020, on a magnetic tape on the MUA0: drive. (This command shows the simplest case, in which you use no options or qualifiers.)

The following sections explain required and optional parameters in the VMSINSTAL.COM command line:

Parameter	Section
Product list	Section 3.4.1
Source	Section 3.4.2
Options	Section 3.4.3
Destination	Section 3.4.4
Backup qualifiers	Section 3.4.5 and Section 3.6.3.3

Section 3.4.6 explains how to complete an installation.

3.4.1 Selecting a Product List

Products are stored in a **save set**, which is a specially formatted file that contains a group of files. Installation and upgrade procedures move the files from the save set to your system disk.

The *product-list* parameter lists the products that you want to install. You can use this parameter when you install layered products or update the operating system. (When you perform an upgrade procedure, you can list only one product, *VMSvvu*.)

Note

Do not use the wildcard character (*) if you are installing layered products and updating the operating system at the same time. In this case, update the system first. If you use the wildcard character, VMSINSTAL sorts the product list in alphabetical order; the operating system would probably not be installed first.

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

```
facvvu
```

Table 3–3 explains the format of *facvvu*. Using this format, you can install a specific version and update of a product from distribution media containing several versions and updates. If you do not include a version and update number, all versions and updates to the specified product from the source are installed in alphabetical order.

Table 3–3 Format of *facvvu* Save-Set File Name

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

If you are installing from a distribution kit, the list of products on your distribution media is included with the bill of materials for the distribution kit. If the list is not available, you can obtain one by using the DIRECTORY command; the distribution media then finds and displays the products that are included.

How to Perform This Task

To obtain the product list, enter commands in the following format:

```
MOUNT/OVERRIDE=ID device:
DIRECTORY device:[0,0]
```

Device is the drive that holds the distribution media.

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

```
DIRECTORY node::device:[directory]
```

Examples

1.
```
$ MOUNT/OVERRIDE=ID MUA0:
%MOUNT-I-MOUNTED, VMS060       mounted on _MUA0:
$ DIRECTORY MUA0:[0,0]
```

   The DIRECTORY command in this example directs the distribution media to find the products included on the MUA0: drive; for example:

```
Directory MUA0:[000,000]

000000.DIR;1        BACKUP.SYS;1        BADBLK.SYS;1        BADLOG.SYS;1
BITMAP.SYS;1        CONTIN.SYS;1        CORIMG.SYS;1        DECW060.C;1
DECW060.D;1         DECW060.E;1         DECW060.F;1         INDEXF.SYS;1
ISL_SCRIPT.ESS;1    SECURITY.SYS;1      SYS0.DIR;1          VMS060.A;1
VMS060.B;1          VMS060.C;1          VMS060.D;1          VMS060.E;1
VMS060.F;1          VOLSET.SYS;1

Total of 22 files.
```

2.
```
$ DIRECTORY BRAVO::DUA1:[0,0]
```

   The DIRECTORY command in this example directs the distribution media to find the products on the node BRAVO, which is on the DUA1: device.

Note

If you want to access a remote node, you must have read and execute access (R,E) to the directory.

3.4.2 Selecting the Source

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

- A drive that holds the distribution media; for example, a TK50 drive designated as the MUA0: drive.
- A disk directory to which the product save set has been transferred from the distribution media for later installation.

  You specify a disk directory as the source when you select the Get Save Set option. For more information about this option, see Section 3.6.3.
- A disk directory on another node.

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

```
* Where will the distribution volumes be mounted:
```

3.4.3 Selecting Options

The VMSINSTAL.COM command procedure permits the use of six options. Table 3–4 briefly describes each option. Section 3.6 contains a detailed description of each option.

Table 3–4 VMSINSTAL.COM Options

Letter Choice	Option	Description
A	Autoanswer	Makes it easier to *reinstall* a product after an upgrade by providing responses to the questions and prompts during the reinstallation. (Specify this option only when installing layered products.)
AWD=	Alternate Working Device	Lets you specify an alternate working device for the temporary working directory. (You can specify this option when installing layered products or applying updates.)
G	Get Save Set	Saves you time by allowing you to store product save sets temporarily on a magnetic tape or in a disk directory. (Specify this option only when installing layered products.)
L	File Log	Logs all file activity to the terminal during installation.
N	Release Notes	Displays or prints the online release notes file supplied by the layered product.
R	Alternate Root	Lets you install the product on a system disk other than that of the running system.

You specify each option by entering the appropriate option letter after the keyword OPTIONS in the command that starts VMSINSTAL.COM. The OPTIONS keyword is optional. However, if you have an option list, you must enter OPTIONS before it. If you enter an option list without the OPTIONS keyword, VMSINSTAL.COM displays an informational error message and the installation ends. (The *option-list* parameter lists the options requested.)

How to Perform This Task

To specify each option:

1. Invoke the VMSINSTAL.COM procedure, entering appropriate option letters after the keyword OPTIONS.
2. Press Return.

If you specify more than one option, separate the options with commas. Do not separate the options with spaces.

Example

```
$ @VMSINSTAL.COM NEWAID021 MTA0: OPTIONS A,N
```

3.4.4 Selecting the Destination

The *destination* parameter is optional. By default, VMSINSTAL.COM assumes that the product is to be installed in the system common directory SYS$COMMON on the system disk. However, you must use this parameter in the following two instances:

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

  device:[SYSn.]

where:

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

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

- To copy the product kit save sets into a storage directory for later installation. For more information on the Get Save Set option, see Section 3.6.3.

  Specify the destination directory in the following format:

  device:[directory]

  where:

device	The destination disk device
directory	Usually a directory dedicated to product save sets on the specified disk

3.4.5 Verifying, Logging, and Confirming the Operation

The VMSINSTAL.COM command procedure includes a BACKUP command that copies the save sets to the destination directory. You can verify, log, and confirm the copy operation by specifying BACKUP qualifiers with the VMSINSTAL.COM Get Save Set option. See Section 3.6.3 for more information.

3.4.6 Completing the Installation

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

- Performs an automatic shutdown of the system; you might be instructed to reboot manually.
- Returns you to the system prompt.

When the product is installed, back up the updated system disk. For instructions, see Section 10.17.

3.5 Recovering from a System Failure

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

- The system disk was not altered before the system failure. In this case, VMSINSTAL.COM instructs you to restart the installation.
- The system disk or a library used by the installation was corrupted. In this case, VMSINSTAL.COM instructs you to restore either the system disk or the corrupted library from the backup copy and to restart the installation.
- VMSINSTAL.COM continues the installation. In this case, the procedure performs most of the installation. In addition, VMSINSTAL.COM might tell you that you must perform some tasks manually to complete the installation.

3.6 Selecting VMSINSTAL.COM Options

When you use VMSINSTAL.COM, the command procedure allows you to select up to six options (summarized previously in Table 3–4.) The following sections describe those VMSINSTAL.COM options in detail.

3.6.1 Using the Autoanswer Option (A) (Layered Products Only)

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

If you specify the Autoanswer option when you install a product initially, an **answer file** is created in the form *product*.ANS in the SYS$UPDATE directory, where *product* is the product name parameter that you provide when you start VMSINSTAL.COM.

The answer file contains a record of your responses to questions and prompts from VMSINSTAL.COM. For example, if you install the product NEWAID010 with the Autoanswer option, VMSINSTAL.COM creates an answer file called NEWAID010.ANS.

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

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

Example

To use the Autoanswer option, specify the following command:

```
$ SYS$UPDATE:VMSINSTAL.COM NEW$PRODUCT010 CSA1: OPTIONS A
```

You can then examine the file:

```
$ TYPE SYS$COMMON:[SYSUPD]NEW$PRODUCT010.ANS;1
* Do you want to install the entire kit [Y]? \
* Are these selections correct [Y]? \
* Does this product have an authorization key registered and loaded? \Y
* Will you allow a system shutdown after this product is installed [YES]? \
* How many minutes for system shutdown [0]: \
* Do you want to do an automatic system reboot [YES]? \
$
```

3.6.2 Using the Alternate Working Device Option (AWD=)

Restriction

Before using this option, check with the product's installation guide to make sure the product is supported.

You can use the Alternate Working Device option to specify an alternate working device for the temporary working directory (defined as the logical name VMI$KWD). This option allows you to perform an installation with fewer free blocks on the system disk than are otherwise required.

If you do not specify this option, VMSINSTAL.COM creates the temporary working directory in the following location:

SYS$SPECIFIC:[SYSUPD.facvvu]

Table 3–3 explains the format of *facvvu*.

How to Perform This Task

Use the following format to specify this option:

AWD=dev:[dir]

where:

dev	Specifies the alternate working device
dir	Specifies the directory under which the *facvvu* subdirectory will be created

Specifying a directory is optional. If you do not specify a directory, VMSINSTAL.COM creates the working directory on the specified device with the following directory specification: [000000.*facvvu*]. If you specify a directory, VMSINSTAL.COM creates the working directory as a subdirectory within the directory that you specify (for example, [WORK.*facvvu*]). If the directory that you specify does not exist, VMSINSTAL.COM does not create it.

Example

To use the working directory [INSTALL] on the alternate device DUA2:, specify the following:

```
$ @SYS$UPDATE:VMSINSTAL.COM NEWAID010 CSA1: OPTIONS AWD=DUA2:[INSTALL]
```

3.6.3 Using the Get Save Set Option (G) (Layered Products Only)

Note

You cannot use this option to copy operating system kits.

Installing products either from a distribution tape or from console media directly onto your system disk is time-consuming. The Get Save Set option saves you time by allowing you to copy product save sets and store them temporarily while you do other work; you can then perform an update more quickly at a time that is convenient for you.

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

3.6.3.1 Storing a Product Save Set

To store a product save set on a disk directory using the Get Save Set option, enter a command using the following syntax:

@SYS$UPDATE:VMSINSTAL.COM product-list source OPTIONS G device:[directory]

The directory you specify must exist, and the device must be mounted.

Examples

If you specify just one option, enter the disk directory name immediately after the OPTIONS G parameter, leaving a space between G and the disk directory. For example, if you are storing save sets for a product named NEWAID010 from the console drive into disk directory USER1:[PRODUCTS], enter the following command:

```
$ @SYS$UPDATE:VMSINSTAL.COM NEWAID010 CSA1: OPTIONS G USER1:[PRODUCTS]
```

If you specify more than one option, place the disk directory name after the last option, leaving a space between the last option and the disk directory name. For example:

```
$ @SYS$UPDATE:VMSINSTAL.COM NEWAID010 CSA1: OPTIONS G, N USER1:[PRODUCTS]
```

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

facvvu.x

Table 3–3 explains the format of *facvvu*. The format of the file type (*x*) is:

x	A literal file type that is used to identify save-set files, where A is the first save set, B the second, and so forth

3.6.3.2 Installing a Product

To install the product on your system, enter a command in the following format:

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

Example

For the product NEWAID010, enter this command:

```
$ @SYS$UPDATE:VMSINSTAL NEWAID010 USER1:[PRODUCTS]
```

VMSINSTAL.COM installs the NEWAID product on your system disk.

3.6.3.3 Specifying Backup Qualifiers

When you use the Get Save Set option you can specify three BACKUP command qualifiers: /VERIFY, /LOG, and /CONFIRM. The qualifiers must be enclosed in quotation marks because they are passed as a parameter of the Get Save Set option (G). VMSINSTAL passes the parameter to the BACKUP command within VMSINSTAL. The following example includes the Get Save Set option and BACKUP qualifiers:

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

Following are explanations of the qualifiers shown in the example:

Qualifier	Explanation
/VERIFY	Compares the contents of the output specifier with the contents of the input specifier after a save, restore, or copy operation completes. If a file does not compare successfully, an error message reporting this fact is displayed.
/LOG	Causes the file specification of each file processed to be displayed at the terminal during the operation. The default is /NOLOG.
/CONFIRM	Displays a prompt on your terminal before each file is processed. To process the file, you must reply by entering Y or YES and pressing the Return key. The system interprets any other response, including simply pressing the Return key, as NO.

See the *OpenVMS System Management Utilities Reference Manual* for more information about the BACKUP command and its qualifiers.

3.6.4 Using the File Log Option (L)

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

3.6.5 Using the Release Notes Option (N)

Use the Release Notes option to display or print the online release notes file supplied by a layered product and by the update procedure.

Note

Not all layered products provide online release notes.

The release notes file receives the file name *facvvu.release_notes*, where *facvvu* represents the product name code, version, and update numbers (see Table 3–3); for example, NEWAID010.RELEASE_NOTES.

How to Perform This Task

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

```
Release notes included with this kit are always copied to SYS$HELP.

Additional Release Notes Options:
    1. Display release notes
    2. Print release notes
    3. Both 1 and 2
    4. None of the above.

❶ *Select option [2]:

❷ *Queue name [SYS$PRINT]:

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

Following are explanations that correspond to the numbered entries in the example.

❶ This prompt allows you to choose options 1 through 4.

❷ This prompt is displayed only if you select option 2 or option 3. If you enter the name of a print queue, the system displays a message saying that the release notes have been queued successfully to the printer. If you do not specify a print queue, the release notes are sent to SYS$PRINT by default.

❸ This prompt allows you either to continue or to end the installation. The default is to end the installation.

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

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

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

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

3.6.6 Using the Alternate Root Option (R)

You can use the Alternate Root option to install the product on a system disk other than that of the running system. You might use this option to test a layered product without disturbing the running system.

The operating system in the alternate root must be complete and have the same version or update level as the running system. All files and software products that the product installation refers to must be present in the alternate root.

Notes

Not all optional software products allow you to install a product on an alternate root. Consult the documentation of the specific product to determine whether you can install the product on an alternate root.

Also, Digital does not support the installation of a product on an alternate root on the existing system disk.

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

3.7 Using the POLYCENTER Software Installation Utility

The POLYCENTER Software Installation utility can help you install, remove, and manage layered software products on Alpha or VAX systems. It can also save information about software products such as system requirements, installation options, and your answers to questions that the product asked during the installation procedure.

3.7.1 Interfaces to Use

You can perform POLYCENTER Software Installation utility operations from either the DCL interface or from the DECwindows Motif interface.

- The DCL interface is used in most of the examples throughout the description of the POLYCENTER Software Installation utility.

  Use the following command format to invoke each operation:

  ```
  $ PRODUCT subcommand parameter[/qualifier1,...]
  ```

  For example, to install COBOL Version 5.0 and the latest version of FORTRAN, enter the following command:

  ```
  $ PRODUCT INSTALL COBOL/VERSION=5.0,FORTRAN Return

  The following products have been selected:
  DEC AXPVMS COBOL V5.0
  DEC AXPVMS FORTRAN V6.2

  Do you want to continue [YES] Return
  ```

  Section 3.8 describes installation.

  You can enter PRODUCT commands at the DCL prompt ($) or in a DCL command procedure. See the *OpenVMS System Management Utilities Reference Manual* for subcommand syntax information.

  To run the POLYCENTER Software Installation utility as a batch job, see Section 3.8.6.

- The Motif interface, which is only available if you have a workstation or an X terminal running DECwindows Motif for OpenVMS, is described in Section 3.11.

  Use the following command format to open the POLYCENTER Software Installation utility's main window:

  ```
  $ PRODUCT
  ```

  You can add the PRODUCT command to the list of applications in the Session Manager Applications menu. To do this, select the Options item in the Session Manager menu bar, and then select the Menus option. Fill in the application name, ''PRODUCT'', and the DCL Command, also ''PRODUCT''. Click on the arrow button that appears, then click on OK or Apply.

3.7.2 Operations to Perform

Table 3–5 lists operations the POLYCENTER Software Installation utility can perform and how to initiate them from each interface.

Table 3–5 Operations and How to Initiate Them from Each Interface

Operations	DCL	Motif
Install a product	PRODUCT INSTALL	Installation[1]
Extract release notes to a file	PRODUCT EXTRACT RELEASE_NOTES	(Not available from Motif interface)
Display the name of product kits found in a specified directory	PRODUCT FIND	Implicit for Copy[1] and Installation[1]
Remove an installed product	PRODUCT REMOVE	Remove[1]
Create a product configuration file (PCF)	PRODUCT CONFIGURE	Reconfiguration[1] ,[2]
Modify the configuration choices for an installed product	PRODUCT RECONFIGURE	Reconfiguration[1]
Register a product that was previously installed with another tool	PRODUCT REGISTER PRODUCT	Registration[1]
Register a change in the volume label for a device that contains installed products	PRODUCT REGISTER VOLUME	Registration[1]
Copy product kits to a new location	PRODUCT COPY	Copy[1]
Display information about managed objects (for example, files or directories) created during a software product installation	PRODUCT SHOW OBJECT	(Not available from Motif interface)
Display a log of operations performed on software products	PRODUCT SHOW HISTORY	Show History[1]
Display information about installed products	PRODUCT SHOW PRODUCT	Show Product[1]

[1]This is an option on the pull-down Mode menu, which is available from most windows in the interface.

[2]This operation can be performed from the Installation option on the Mode menu in the interface.

3.7.3 Product Files and Databases

The following files are used by the POLYCENTER Software Installation utility:

- The **product description file** (PDF), is provided by the software manufacturer. It contains all the information the POLYCENTER Software Installation utility needs for installing either a software product or a set of software products. The PDF includes a list of configuration choices the product offers, default choices, and product requirements (such as minimum hardware configurations and system parameter values).
- A **product text file** (PTF), is optionally supplied by the software manufacturer. It provides information about the product. The information includes product name, producer, configuration choice descriptions, and message text used during product installation.
- A **product configuration file** (PCF), is optional. It may be supplied by software manufacturer, or you can create it by using the CONFIGURE operation. A PCF contains responses to some or all of the installation questions for a product. It can provide default or required choices, which may differ from the default choices provided in the PDF.
- The **product database** (PDB), is created automatically by the POLYCENTER Software Installation utility. When products are installed, the files and other objects that make up the product, such as directories and accounts, are recorded in the PDB. The configuration choices made during installation are also recorded.

  You can access the PDB to show what products are installed and the dependencies between them. You can also list the files and other objects that make up each product, or the history of installation and upgrade activity.

3.7.4 Format of Software Product Kits

Software products compliant with the POLYCENTER Software Installation utility are distributed in one of two formats:

- Sequential copy format. In this form, the PDF, the PTF, and all files that comprise the product are packaged in a single container file. This container file can be placed either on a random-access device, such as a compact disc, or on a sequential access device, such as a magnetic tape. Most layered products are distributed in sequential copy format.
- Reference copy format. In this form, the product description file and all files that comprise the product are placed in a directory tree on a random-access device. OpenVMS is distributed in reference copy format on CD–ROM.

3.7.5 Software Product Name Conventions

A software product kit packaged in sequential copy format has a container file named in the following format:

```
producer-base-product-version-kit_type.PCSI
```

A software product kit packaged in reference copy format has a product description file in the root directory named in the following format:

```
producer-base-product-version-kit_type.PCSI$DESCRIPTION
```

Each subfield is separated by a hyphen and is defined as follows:

- *producer* is the legal owner of the software product.

- *base* is the base system that specifies the hardware and software platform on which the product runs (for example, AXPVMS or VAXVMS).
- *product* is the name of the software product.
- *version* identifies the version according the format *tmmnn-ue*, described in Section 3.7.5.1.
- *kit_type* identifies a kit type specified as a value from 1 through 7, as shown in Table 3–6.

Table 3–6 PDF Kit Types and Values

Value	Type	Description
1	Full	Application software.
2	Operating system	Operating system software.
3	Partial	An update for the operating system or for a product that is currently installed.
4	Patch	A correction to existing software.
5	Platform	A software system that has more than one product.
6	Transition	A kit that is intended to register a product that was not installed by the POLYCENTER Software Installation utility. Transition kits include only a product definition file and (optionally) a product text file.
7	Mandatory update	A required update, which produces functionally updated software that has the same version number.

3.7.5.1 Version Identification Format

The version of the software product kit is in the format *tmmnn-ue*. This format is described in Table 3–7.

Table 3–7 Format of *tmmnn-ue* Version Identification

t	The type of version (a single uppercase alphabetic character).
mm	The major version number (decimal integer 01 through 99).
nn	The minor version number (decimal integer 00 through 99).
-	The hyphen is required in all cases. When both update level (u) and maintenance edit level (e) are omitted, the version will end with a hyphen and the file name will have a double hyphen (- -) preceding the kit type.
u	The update level (decimal integer 1 through 999). The level is optional.
e	The maintenance edit level (one or more alphanumeric characters beginning with an alphabetic character). This level is optional.

Reading a Displayed Version

The *tmmnn-ue* format used in file names is very similar to the format used to display versions or to enter versions using the /VERSION qualifier. However, when versions are displayed, leading zeros are omitted in *mm* and *nn*, and if neither *u* nor *e* is present, the hyphen (-) is omitted. When you read a displayed version, keep the following guidelines in mind:

- If a hyphen is present and the first character after the hyphen is a digit, then the leading digits after the hyphen are the update level. If nondigit characters

are present, the maintenance edit level consists of the first nondigit character and all following characters. If nondigit characters are not present, the maintenance edit level is blank.

- If a hyphen is present and the first character after the hyphen is a nondigit character, the update level is zero (0) and the maintenance edit level consists of all of the characters after the hyphen.
- If no hyphen is present, the update level is zero and the maintenance edit level is blank.

3.7.5.2 Looking at Software Product Name Examples

The following examples show how the format is used for a sequential copy format kit and two reference copy format kits:

- A sequential copy format kit for DEC Softwindows for OpenVMS VAX and that requires a double hyphen has the following format:

  ```
  DEC-VAXVMS-SOFTWIN-V0101--1.PCSI
  ```

  This format shows that the *producer* is DEC (Digital), the *base* is VAXVMS (OpenVMS VAX), the *product* is SOFTWIN, and the *version* is V1.1. The type of version is V, the major and minor version numbers are each 1. There are no update or maintenance edit levels. The *kit_type* is 1 (full).

- A product description file in a reference copy format kit for OpenVMS Alpha has the following format:

  ```
  DEC-AXPVMS-VMS-V0601-1H2-2.PCSI$DESCRIPTION
  ```

  This format shows that the *producer* is DEC (Digital), the *base* is AXPVMS (OpenVMS Alpha), the *product* is OpenVMS, and the *version* is V6.1-1H2. The type of version is V, the major version number is 6, the minor version number 1, the update level is 1, and the maintenance edit level is H2. The *kit_type* is 2 (operating system).

- A product description file in a reference copy format kit for a field test version of OpenVMS Alpha has the following format:

  ```
  DEC-AXPVMS-VMS-T0602-FT2-2.PCSI$DESCRIPTION
  ```

  This format shows that the *producer* is DEC (Digital), the *base* is AXPVMS (OpenVMS Alpha), the *product* is OpenVMS, and the *version* is V7.0-FT2. The type of version is T, the major version number is 7, and the minor version number is 0. There is no update level because the first character after the hyphen is not a digit. The maintenance edit level is FT2 and the *kit_type* is 2 (operating system).

3.7.6 Creating a Product Configuration File (PCF)

You can create a PCF before or during an installation. You can also create more than one PCF for each product, thereby helping you to customize software installations for unique hardware situations or for different usage patterns within a group.

If a PCF is present and it contains a response for a configuration choice, the default for that choice comes from the PCF. The PCF specifies whether the choice can be changed or whether it is required.

If a PCF is not present or does not contain a response for a configuration choice, the default choice comes from one of two places:

- If the product database (PDB) contains an entry for the choice that was made in a previous installation, then the PDB entry becomes the default configuration choice. It can be changed.
- If the PDB does not contain an entry for the choice, either because the product was not previously installed or because this is a new choice, then the default configuration choice comes from the PDF. It can be changed.

3.7.6.1 Configuration Options

Some of the options available for customizing the PCF are:

- Saving your answer. You can specify that your response to a question (rather than the current default value) be stored in the PCF.
- Not saving your answer. When creating a PCF during an installation, you can answer a question without recording your answer in the PCF. This is useful for responding to questions that are specific to a single system or installation.
- Deferring a question so that it is asked again during a future installation. For example, you might want an installer to verify that a particular response is still valid for the systems on which each installation is being performed.
- Preventing a question from being asked again. If you do not defer a question when you create a PCF, the response recorded in the PCF is used during future installations. The installer is not prompted for the information. This reduces the length and complexity of the actual installation procedure.

3.7.6.2 Configuration Commands

You can use either DCL or the DECwindows Motif interface to define a PCF. Whichever you use, the POLYCENTER Software Installation utility creates a PCF in your current default directory. The default PCF is named DEFAULT.PCSI$CONFIGURATION. If you want to override the default file name or directory, use the /CONFIGURATION=OUTPUT qualifier. See the *OpenVMS System Management Utilities Reference Manual*.

To define a PCF from DCL, invoke the PRODUCT CONFIGURE command. For example:

```
$ PRODUCT CONFIGURE CHESSMASTER
```

To create a PCF using the DECwindows Motif interface, choose either Installation or Reconfiguration from the Mode menu. Select the Source/Work qualifier and enter a valid source location for the Product Definition File (PDF). Select one or more products. Select Write Configuration from the Qualifiers menu. (You may also wish to select Read Configuration.) Enter a file name for Read Configuration, Write Configuration, or both. Expand the selected product or products to see the Write/NoWrite and Defer/NoDefer icons.

Make configuration choices by selecting or canceling options and setting the Write and Defer options. The Write icon is a pen. If the pen is down, Write is set; if the pen us up, NoWrite is set. The Defer icon is an arrowhead. If the arrowhead is highlighted, Defer is in effect; if the arrowhead is not highlighted, NoDefer is in effect.

When the configuration choices are correct, click on the Apply button for Write Configuration to write the PCF. If you do not want to perform the actual installation or reconfiguration, click on the Reset button instead of the Install or Reconfigure button.

3.7.6.3 Recording Configuration Choices

After defining the PCF, the POLYCENTER Software Installation utility prompts you with questions about the product. Determine how and whether your responses are recorded in the PCF by responding to the questions and using two predefined function keys. The following table shows how your responses configure the PCF:

Key	Action by the POLYCENTER Software Installation utility
Return	Accepts the default or explicitly entered choice for the current operation and for entry into the PCF, and then moves to the next choice. If the Defer option is in effect, this entry can be changed when the PCF is used for future installations or upgrades. If the Defer option is not in effect, this entry cannot be changed when the PCF is used for future installations or upgrades. If the Write option is in effect, this entry, including the Defer option, is written into the PCF and used when the PCF is used for future installations or upgrades. If the Write option is not in effect, this entry, including the Defer option, is not written into the PCF and is not used when the PCF is used for future installations or upgrades. In this case, the default for the future installation or upgrade will come from the PDF or PDB.
F17	Toggles the Defer option. By default, the Defer option is not in effect.
F18	Toggles the Write option. By default, the Write option is in effect.

Press the Return key after each response.

Example 3–1 shows how to use keys F17 and F18 in the PCF. Note that this is an example only and does not necessarily represent an actual PCF for a product. A description of the callouts follows Example 3–1.

Example 3–1 Sample Procedure for Creating a PCF

```
$ product configure vms /source=sys$sysdevice:[vms$common] -
_$ /configuration=(output=mypcf) «Return»

$ PRODUCT CONFIGURE VMS/SOURCE=SYS$SYSDEVICE:[VMS$COMMON] -
_$ /CONFIGURATION=(OUTPUT=MYPCF)

The following product has been selected:
DEC AXPVMS VMS V7.0     [Available]

Do you want to continue [YES] Return

*** DEC AXPVMS VMS V7.0: OpenVMS Operating System

Copyright © 1995 Digital Equipment Corporation

    Digital Equipment Corporation

    Do you want all the default values for this
product [YES] N Return ❶

    Accounting Log Report Generator Utility [YES] F17 ❷
%PCSI-I-DEFER, that item has been deferred. Please set the default value.

    Accounting Log Report Generator Utility [YES] F17 ❸
%PCSI-I-UNDEFER, that item is no longer deferred. Please set the default value.

    Accounting Log Report Generator Utility [YES] F17 ❸
%PCSI-I-DEFER, that item has been deferred. Please set the default value.

    Accounting Log Report Generator Utility [YES] Return ❹

    Access Control List Utilities [YES] F18 ❺
%PCSIUI-I-UNWRITE, that item will not be written to configuration file.
Please set the value.

    Access Control List Utilities [YES] Return ❻

    Print and Batch Queue Utilities [YES] NO ❼

    DECdtm Distributed Transaction Manager [YES] Return ❽
.
.
.
    Do you want to view the values? [NO] Return

    Are you satisfied with your answers [YES] Return
%PCSI-I-WRICON, writing configuration file
    SYS$SYSDEVICE:[VMS$COMMON]MYPCF.PCSI$CONFIGURATION;1 ❾
$
```

The callouts in Example 3–1 mark the following actions:

❶ Chooses to select values for individual options instead of accepting default values for all of the options.

❷ Requests (by using the defer key, F17) that the installer be given the choice of whether or not to install the optional example files.

❸ Toggle the the defer option (twice).

❹ Records the default response (Yes) in the PCF. Because the defer option was in effect, when the PCF is used during a future installation, the installer can select the Accounting Log Report Generator utility by default, or can choose not to to select it.

❺ Requests the nowrite option, key F18, so that this choice will not be written to the PCF.

❻ Chooses not to install the Access Control List utilities. Because the nowrite option is in effect, this choice is not written to the PCF.

❼ Chooses not to install the Print and Batch Queue utilities. Because the nowrite option is not in effect (by default) and the defer option is in effect (by default), this choice is written to the PCF and the question is not asked again when the PCF is used during a future installation.

❽ Accepts the default to install the DECdtm Distributed Transaction manager. Because the nowrite option is not in effect (by default) and the defer option is in effect (by default), this choice is written to the PCF and the question is not asked again

❾ Displays the name of the PCF that has been created, DEFAULT.PCSI$CONFIGURATION. The POLYCENTER Software Installation utility displays this message only when you enable message logging by including /LOG in the PRODUCT CONFIGURE command line.

To review your configuration responses, specify the name of the PCF on the PRODUCT CONFIGURE command line and then scan your answers. Press the Return key to leave your answers unchanged.

When you use a single DCL command or Motif operation to install or configure more than one product and write the responses to a PCF, the information for all of the products installed or configured is contained in a single PCF. Use separate operations to install or configure a set of products when you want to keep each product's configuration values in its own PCF.

3.7.6.4 Modifying an Existing PCF

You can modify an existing file by using DCL or Motif. In DCL, specify the name of the PCF to be modified and the name of the PCF to be created. Include both the INPUT and the OUTPUT keywords with the /CONFIGURATION qualifier on the PRODUCT CONFIGURE command line. For example, read the default values in the file PRODUCTA_REV1.DAT, make changes to the file, and save the changes to PRODUCTA_REV2.DAT, the output file:

```
$ PRODUCT CONFIGURE -
_$ /CONFIGURATION=(INPUT=PRODUCTA_REV1.DAT,OUTPUT=PRODUCTA_REV2.DAT) -
_$ PRODUCTA
```

In DECwindows Motif, select both Read Configuration and Write Configuration as described in Section 3.7.6.2.

3.7.7 Using a Product Database

The POLYCENTER Software Installation utility automatically stores information about product installation, configuration choices, and object, such as files and directories, that make up the product in the product database. The product database is useful for recalling information about products installed on your system and for detecting and tracking product dependencies.

3.7.7.1 Adding Information to the Database

Although the POLYCENTER Software Installation utility stores product information for you automatically, you can also add your own information. When you perform a task, you can include a remark—a comment to be recorded in the product database—along with the other information about the task being performed. To add a remark to the product database, use the /REMARK qualifier with either the PRODUCT CONFIGURE or PRODUCT REGISTER PRODUCT

command. See the *OpenVMS System Management Utilities Reference Manual* for information about these commands and the /REMARK qualifier.

3.7.7.2 Registering a Noncompliant Product

To register a product that was installed with a tool other than the POLYCENTER Software Installation utility, enter the PRODUCT REGISTER PRODUCT command. This command records information provided by a PDF. For example:

```
$ PRODUCT REGISTER PRODUCT TOOLCHEST
```

If you do not have a PDF for a product you want to register, enter the following command:

```
$ @SYS$UPDATE:PCSI$REGISTER_PRODUCT.COM
```

This procedure prompts you for the product name, version, and producer. For example, the ''producer'' for Digital products is ''DEC''. The procedure uses this information to create a temporary, minimal PDF. It then executes the PRODUCT REGISTER PRODUCT command to register the product, and deletes the temporary PDF.

Because PCSI$REGISTER_PRODUCT.COM creates only a minimal PDF, it cannot register with the POLYCENTER Software Installation utility database all the information about the product. For this reason, if a PDF for the product is available, use it.

Although a ''transition'' PDF is intended specifically for the PRODUCT REGISTER PRODUCT command, ''full'' and ''operating system'' PDFs can also be registered.

3.7.7.3 Detecting and Tracking Software Dependencies

Some software products depend on other software products to work correctly. For example, a product might work only when a specific version of another product is installed on the system. The POLYCENTER Software Installation utility detects and tracks the dependencies of the products that you install. The POLYCENTER Software Installation utility also attempts to satisfy the requirements of multiple products.

In some instances, the POLYCENTER Software Installation utility is unable to resolve product dependency issues. In such instances, the POLYCENTER Software Installation utility provides feedback on the nature of the conflict and asks you to decide how to proceed.

3.8 Installing with the POLYCENTER Software Installation Utility

The basic steps for installing a software product are:

1. Perform the preliminary steps.
2. Review the product's release notes and installation information.
3. Start the installation. Use either the DCL interface or DECwindows Motif interface. Section 3.7.1 describes the interfaces.
4. Respond to installation questions about product options. The product is not installed on your system until you confirm your selections.
5. Confirm your selections to install the product.

3.8.1 Performing Preliminary Steps

Before installing software, follow these steps:

1. Back up your system disk.
2. Assign privileges.
3. Optionally identify source and destination locations.
4. Install prerequisite software.
5. Identify postinstallation procedures.

3.8.1.1 Assigning Privileges

Make sure you have the following privileges:

BYPASS	OPER
CMKRNL	SYSPRV
LOG_IO	VOLPRO

For many operations, you must specify a location where the software kit resides and a location where you want to install the software. Two methods are available for identifying these locations:

- Defining logical names
- Specifying /SOURCE and /DESTINATION qualifiers on the command line, or in DECwindows Motif, selecting a source and destination

You can also define logical names, and then override them by using the /SOURCE and /DESTINATION qualifiers.

Note

If you do not deassign logical names after they are used, they can cause unexpected results in future operations of the POLYCENTER Software Installation utility. Digital recommends that you use the /SOURCE and /DESTINATION qualifiers.

Logical name PCSI$SOURCE, defines the location of the software kits you want to install. Logical name PCSI$DESTINATION defines the location where you want to install the software. For example, if the software is located in DISK1:[KITS] and you want to install it in DISK2:[APPLICATIONS], include the following lines in the system manager's login command file:

```
$ DEFINE PCSI$SOURCE DISK1:[KITS]
$ DEFINE PCSI$DESTINATION DISK2:[APPLICATIONS]
```

You can override the logical name definitions in either the DCL or DECwindows Motif interfaces. In DCL, use /SOURCE and /DESTINATION qualifiers to specify different sources and destinations. In DECwindows Motif, use the Customize Qualifiers item from the Customize menu to select different sources and destinations. See Section 3.11.5.

If you do not define PCSI$DESTINATION, the utility installs the software product in SYS$COMMON:[VMS$COMMON].

3.8.1.2 Installing Prerequisite Software

Install any prerequisite software or perform any prerequisite tasks. This information should be in the software product's installation instructions or release notes.

3.8.1.3 Identifying Postinstallation Procedures

Note any postinstallation procedures. This information should also be in the software product's installation instructions or release notes.

3.8.2 Extracting a Product's Release Notes

To read a product's release notes, extract a copy of them to a file. For example, use the following command to copy the CMS product release notes to a text file:

```
$  PRODUCT EXTRACT RELEASE_NOTES CMS/file=cms_relnotes.txt
```

If you do not specify a file name, the release notes are written to a file named DEFAULT.PCSI$RELEASE_NOTES in the current directory. It is not necessary to install a software product before you use the POLYCENTER Software Installation utility to extract its release notes.

Extracting release notes is available only through the DCL interface.

3.8.3 Installing a Product

To start an installation, enter the PRODUCT INSTALL command. For example:

```
$  PRODUCT INSTALL CMS
```

To install more than one product at a time, enter a list of product names separated by commas. You can use asterisk (*) wildcard characters in the product names. For example:

```
$  PRODUCT INSTALL CMS/VERSION=3.4,LSE,COB*/VERSION=5.0
```

Table 3–8 lists some of the features you can control with command qualifiers. A complete list is in the *OpenVMS System Management Utilities Reference Manual* and in online help.

Table 3–8 Features You Can Request During an Installation

Feature	Qualifier
Supply answers from a PCF	/CONFIGURATION=INPUT=*file-spec*
Create a new PCF	/CONFIGURATION=OUTPUT=*file-spec*
Specify where to install the files	/DESTINATION=*location*
Display full descriptions of all product installation options and information	/HELP
Display log messages on your terminal	/LOG
Save previous versions of product files during an upgrade	/KEEP
Include a remark in the product database	/REMARK

(continued on next page)

Table 3–8 (Cont.) Features You Can Request During an Installation

Feature	Qualifier
Specify where the distribution kit is located	/SOURCE
Specify configuration variables	/CONFIGURATION=*keyword*
Specify a work area for temporary files	/WORK=*device*

3.8.3.1 Using an Existing PCF

Section 3.7.6 describes how to create a PCF before installing a product. To use this existing PCF during the installation, include the /CONFIGURATION=INPUT qualifier on the PRODUCT INSTALL command line. For example, to install CMS and use configuration choices recorded in the PCF named DEC-VAXVMS-CMS.PCSI$CONFIGURATION:

```
$ PRODUCT INSTALL/CONFIGURATION=INPUT=DEC-VAXVMS-CMS.PCSI$CONFIGURATION -
_$ CMS/VERSION=3.4
```

3.8.3.2 Creating a New PCF During the Installation

If you did not create a PCF before the installation, you can create one during the installation. Include the /CONFIGURATION=OUTPUT=*filename* qualifier on the PRODUCT INSTALL command line. For example:

```
$ PRODUCT INSTALL/CONFIGURATION=OUTPUT=CMSV3.DAT CMS/VERSION=3.0
```

As you respond to questions about the options for CMS Version 3.0, your responses are recorded in the PCF named CMSV3.DAT in your current default directory.

For more information about product configuration files, see Section 3.7.3.

3.8.4 Responding to Installation Questions

During an installation, you can request a full description of product options or an explanation to any single question. You can also accept the default value to any single question or to an entire subset of questions.

3.8.4.1 Requesting an Explanation to Questions

To request a full description of all product options and information, include the /HELP qualifier with PRODUCT INSTALL command. To request help about an individual question, press the Help key or PF2 in response to the question. The POLYCENTER Software Installation utility displays a description (if one is available) and a summary of disk and memory requirements for the option.

The following example uses the Help key:

```
$ PRODUCT INSTALL UCX
   .
   .
   .
Optional example files may be installed... [YES] Help

The example files include client server programming examples.

Block Size -      Total:    507  Optional:      0  Required:    507
Global Pages -    Total:      0  Optional:      0  Required:    0
Global Sections - Total:      0  Optional:      0  Required:    0
```

```
Optional example files may be installed... [YES] Return
   .
   .
   .
```

The amount of information varies; some products provide more information than others, and some products provide no information.

3.8.4.2 Accepting Default Answers

Default answers come from one of three places:

- A Product Configuration File (PCF), if one is supplied
- The Product Database (PDB), for upgrades of previously installed products
- The Product Description File (PDF)

If you specify an input PCF and it contains an answer for an option, the default answer from the PCF is used. Depending on the entry in the PCF, the default answer may or may not be allowed to change,

If no input PCF exists, or if the input PCF does NOT contain an answer for an option, the default answer comes from either the PDB or the PDF. If the PDB is present and contains the option, then the default answer comes from the PDB. If the PDB is not present (a new installation) or does not contain the option (a new option), then the default comes from the PDF. Default answers that come from either the PDB or PDF may be changed.

To answer an option, either press the Return key to accept the default answer or supply your own answer and then press the Return key.

Some products contain a subset of questions or options. During an installation procedure, you can choose to accept the default values for an entire subset or you can select to answer each option in the subset.

When you select an option that has suboptions, the POLYCENTER Software Installation utility will ask:

```
Do you want all the defaults for the subitems? [YES]
```

If you answer YES, you will not be asked about the subitems. Instead, the utility will use the defaults for the subitems. If you answer NO, the utility asks you about each subitem.

3.8.5 Confirming Your Answers

After you respond to questions about product options, the POLYCENTER Software Installation utility can display a summary of your answers. For example:

```
Do you want to view your answers [YES] Return
DEC TCP/IP Services for OpenVMS
    Optional example files may be installed...: NO
    Optional NFS files may be installed...: NO
    Optional applications may be installed...: YES
```

The POLYCENTER Software Installation utility then asks:

```
    Are you satisfied with your answers [YES] Return
```

If you are not, answer No to this question. You can then either enter your answers again or exit the installation procedure:

```
Do you want to re-enter your answers [YES] NO Return
%PCSI-I-USERABORT, operation aborted by user
```

By answering No to this question, you can end the installation procedure. The product is not installed; your system remains unchanged.

3.8.6 Performing the Installation as a Batch Job

To run the POLYCENTER Software Installation utility as a batch job, include PRODUCT commands in a command procedure file and then submit the file to a batch queue. In the command procedure, include the /CONFIGURATION qualifier to specify an existing PCF so the POLYCENTER Software Installation utility can respond to questions about product options and configuration choices. If you do not specify /CONFIGURATION, the defaults are used.

Example 3–2 shows how a product might be installed using a command procedure. The example sets and restores VERIFY, and times the installation.

Example 3–2 Sample Command Procedure for Installing a Product

```
$ SAVE_PROC_VERIFY = F$ENVIRONMENT("VERIFY_PROCEDURE")
$ SAVE_IMAGE_VERIFY = F$ENVIRONMENT("VERIFY_IMAGE")
$ SET VERIFY
$ ON ERROR THEN GOTO ERROR_EXIT
$ START_TIME = F$TIME()
$ WRITE SYS$OUTPUT "START TIME -- ''START_TIME'"
$ PRODUCT INSTALL CHESSMASTER -
    /CONFIGURATION=PRODUCER -
    /HELP -
    /LOG
$ERROR_EXIT:
$ END_TIME = F$TIME()
$ TEMP = F$VERIFY(SAVE_PROC_VERIFY,SAVE_IMAGE_VERIFY)
$ WRITE SYS$OUTPUT "  --------------------------------"
$ WRITE SYS$OUTPUT "  END TIME -- ''END_TIME'"
$ WRITE SYS$OUTPUT "START TIME -- ''START_TIME'"
$ WRITE SYS$OUTPUT "  --------------------------------"
$ EXIT
```

3.9 Performing Other Operations on Installed Software Products

You may want to perform other operations on installed software products (for example, reconfiguring choices made during the installation, recording changes in volume label, or copying the software to a new location or to different media). You may also want to convert a kit to a new format or display product information.

3.9.1 Reconfiguring an Installed Product

After you install a product, you might decide to change the configuration choices made during the installation. This is called **reconfiguration**. You choose the options you want; the POLYCENTER Software Installation utility makes all the necessary changes.

To change the configuration choices for an installed product, use the PRODUCT RECONFIGURE command. The product kit must be present and specified by SOURCE.

3.9.2 Recording a Change in Volume Label in the Product Database

To record a changed volume label in the product database, enter the PRODUCT REGISTER VOLUME command followed by the current volume label and the new volume label.

This command replaces all occurrences of the old volume label with the new volume label. (The POLYCENTER Software Installation utility reads the new label from the disk.)

Note that the PRODUCT REGISTER VOLUME command changes the information in the product database only; it does not change the label on the volume. To rename a volume, use the DCL command SET VOLUME. Then, use the PRODUCT REGISTER VOLUME command to record the new name.

You can also use the PRODUCT REGISTER VOLUME command to record a change in the physical or logical device name.

3.9.3 Copying a Software Kit to a New Location

You can use the POLYCENTER Software Installation utility to copy product distribution kits from one location to another. If you transfer a kit from tape to disk, you can change the format from a sequential copy to a **reference copy**.

To copy a software kit from one location to another, use the PRODUCT COPY command. Specify the current location with the /SOURCE qualifier and the new location with the /DESTINATION qualifier. For example:

```
$ PRODUCT COPY/SOURCE=WORK_DISK:[KITS]/DESTINATION=LOCAL_DISK:[KIT_INSTALL] CMS
```

3.9.4 Converting a Software Kit from One Format to Another

You can convert a software kit from reference format (on disk or CD–ROM) to sequential format or from sequential format to reference format by using the PRODUCT COPY command with the /FORMAT qualifier.

For example, to convert CMS from sequential format to reference format, enter the following command:

```
$ PRODUCT COPY/FORMAT=REFERENCE/SOURCE=MUA1:/DESTINATION=LOCAL_DISK:[KIT_INSTALL] CMS
```

The *POLYCENTER Software Installation Utility Developer's Guide* contains more information on how to convert software kits.

3.9.5 Retrieving Product Information

All of the information stored by the product database (PDB) can be accessed by using the SHOW OBJECT, SHOW PRODUCT, and SHOW HISTORY commands. This section describes how to use these commands to retrieve information from the PDB.

For information on how to display information about installed products with the DECwindows Motif interface to the POLYCENTER Software Installation utility, choose On Window from the Help menu.

3.9.5.1 Displaying Information About Objects

To display information about the managed objects (for example, files, accounts, and directories) associated with the products installed on your system, use the SHOW OBJECT command. Table 3–9 lists questions that can be answered with the SHOW OBJECT command.

Table 3–9 SHOW OBJECT Command: Displaying Managed Object Information

Question	Command
What files or other objects did this product create?	PRODUCT SHOW OBJECT * /PRODUCT=*product-name*
What product created this file or other object?	PRODUCT SHOW OBJECT *object-name*/FULL

3.9.5.2 Displaying Information About the Products

You can obtain information about products installed on your system with the SHOW PRODUCT and SHOW HISTORY commands. Table 3–10 lists some of the questions that can be answered with these commands.

Table 3–10 SHOW PRODUCT and SHOW HISTORY Commands: Displaying Product Information

Question	Command
Which products have been installed?	PRODUCT SHOW HISTORY */OPERATION=INSTALL PRODUCT SHOW PRODUCT *
Product interdependencies: Is product *A* referenced by product *B*?	PRODUCT SHOW PRODUCT *A*/FULL PRODUCT SHOW PRODUCT */REFERENCED_BY=*B*
Which user installed a product?	PRODUCT SHOW HISTORY *product-name* /FULL
Which products were installed before March 31, 1995?	PRODUCT SHOW HISTORY */BEFORE=31-MAR-1995
Were any software patches applied to a product?	PRODUCT SHOW PRODUCT *product-name*/FULL

3.10 Removing Installed Software Products and Kits

When you use the POLYCENTER Software Installation utility to remove an installed product, all of the files, accounts, and other objects that were created for the product when it was installed are removed from your system and from the product database.

To remove an installed product, enter the PRODUCT REMOVE command. For example:

```
$ PRODUCT REMOVE CMS
```

3.11 Using the DECwindows Motif Interface

Using the Motif interface requires a workstation or X terminal running DECwindows Motif for OpenVMS. For general information on how to use DECwindows Motif, see *Using DECwindows Motif for OpenVMS*.

Starting the Interface

To begin an installation or to perform another software management operation, enter the PRODUCT command at the DCL ($) prompt:

```
$ PRODUCT Return
```

This command opens the main window in the POLYCENTER Software Installation utility. See Figure 3–1.

General Steps for Using

The general steps for using the DECwindows Motif interface with the POLYCENTER Software Installation utility are:

1. Select the operation to perform from the Mode menu.
2. Select one or more products from the displayed list of products.
3. Respond to questions about product options, expanding lists of suboptions as needed.
4. Perform the operation.

Getting Help

Detailed information about how to install and manage software products is available on line; all interactive windows offer a Help menu from which you can choose topics.

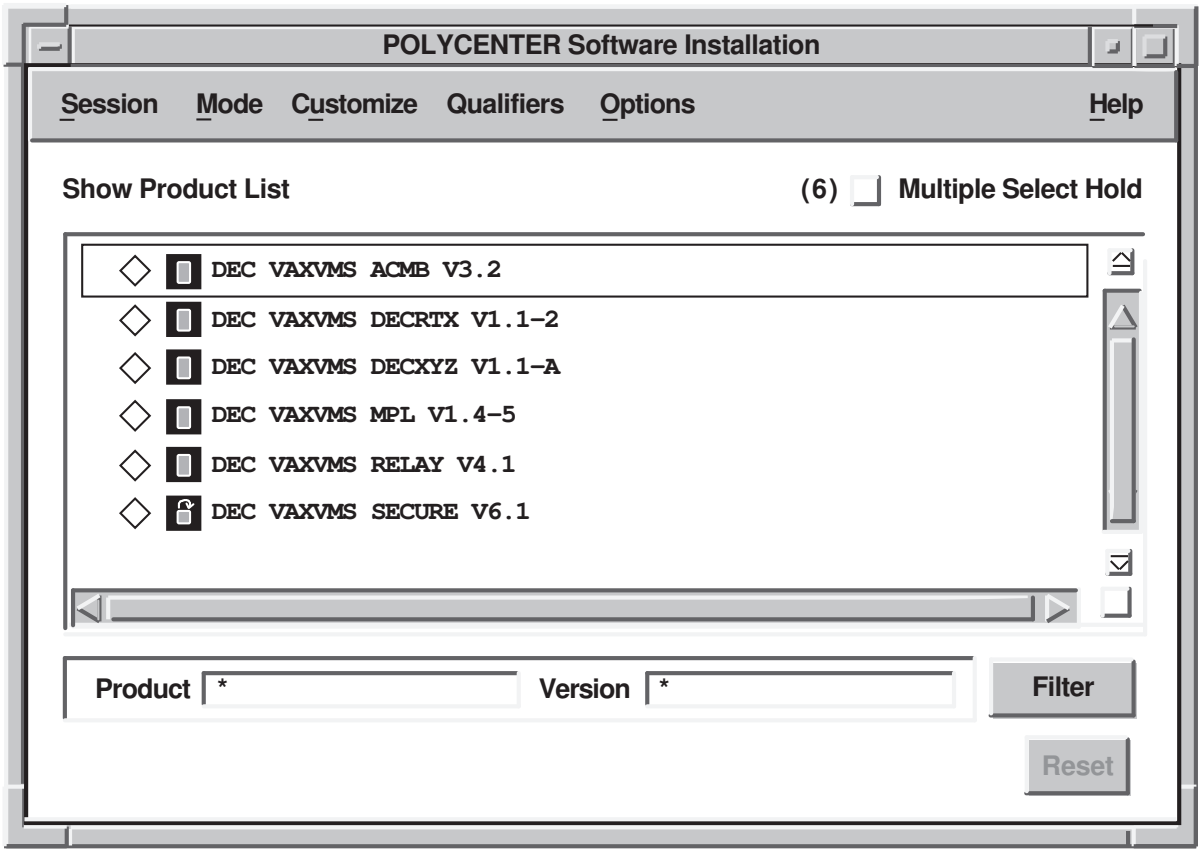
Exiting from the POLYCENTER Software Installation Utility

To exit from the POLYCENTER Software Installation utility, choose End Session from the Session menu.

3.11.1 The POLYCENTER Software Installation Utility Main Window

Invoking the PRODUCT command opens the main window, which displays the Show Product List. See Figure 3–1. The Show Product List includes all products known to the utility's database. If no products are found (for example, when you use the POLYCENTER Software Installation utility on your system for the first time), the Show Product List is empty.

Figure 3–1 POLYCENTER Software Installation Utility Main Window



ZK–6650A–GE

Each product is listed by producer name, base system, product name, and version number. Producer names can be abbreviated. In Figure 3–1, the producer is Digital and is abbreviated as DEC; the base system is the OpenVMS VAX operating system and is abbreviated as VAXVMS.

Note

Sample software product names used in this manual are intended only as examples. Software product names shown do not necessarily exist as software products with which you can use the POLYCENTER Software Installation utility.

The Product and Version fields display an asterisk (*) wildcard character, indicating that all versions of all products are displayed in the Show Product List box. To display only specific products or versions, type the product name and version number in the respective fields (wildcard characters are accepted), and click on the Filter button.

3.11.2 Specifying Operations to Perform

The Mode menu lists operations that the POLYCENTER Software Installation utility can perform. Table 3–11 lists the operations on the Mode menu:

Table 3–11 Operations Listed on the Mode Menu

Operation	Description
Installation	Installs one or more products.
Reconfiguration	Configures or reconfigures a product.
Remove	Removes one or more products that were previously installed by the POLYCENTER Software Installation utility.
Show Product	Displays a list of products that were installed by the POLYCENTER Software Installation utility. This list is also displayed when you first invoke a Motif session with the POLYCENTER Software Installation utility.
Show History	Displays a history of operations for one or more products.
Copy	Copies a software kit from one location or format to another.
Registration	Registers one or more products in the product database.

How to Perform This Task

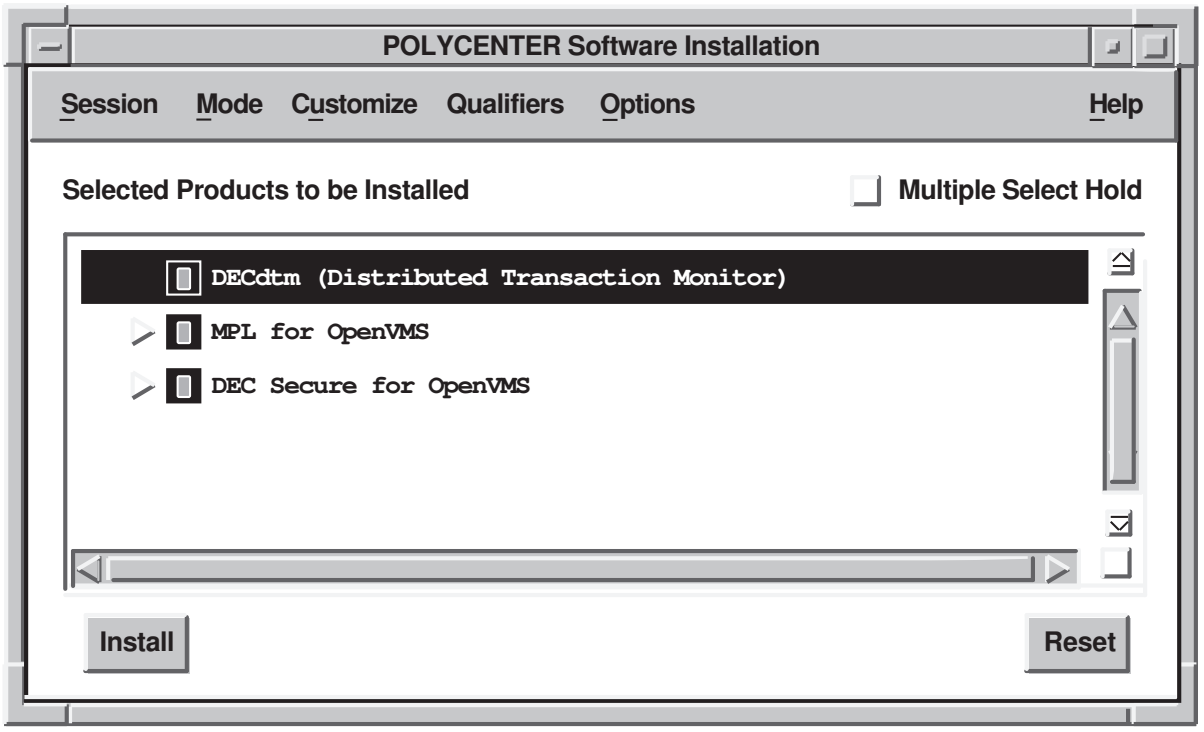
1. Click the Mode menu to display the pull-down list of operations.
2. Specify an operation from the pull-down list. The button in the lower left corner of the window displays the operation you select. This button is called the operation button. The text on this button changes, depending on the operation you select. For example, if you select the Installation operation from the Mode menu, the text on the button reads Install.

After you specify an operation, select products from the list of products.

3.11.3 Selecting Products and Product Options

You can select one or more products for any operation. For example, you can install many products at the same time. You can also select and display information about a product, such as what options are available and which options you want to include in the installation. A product has options if a right-pointing triangle precedes the product name. See Figure 3–2.

Figure 3–2 Selected Products Window



ZK–6648A–GE

Product Selection

To select a product, double click on the product name in the list of products.

To select more than one product, make sure the Multiple Select Hold button in the upper right corner is filled. If it is not filled, click it to fill it. Then, click on each product for which you want the operation to be performed. After selecting the products you want, clear the Multiple Select Hold button by clicking it. This removes nonselected products from the display so that only those you selected are listed.

Option Selection

To select and display information about options for a product, double click the right-pointing triangle. The option names are listed. To see a full description of the option, if one is provided, double click the option name. Options are described in Section 3.7.6.

Icon Conventions

The POLYCENTER Software Installation utility uses icons to indicate whether and how a product option has been selected.

- The checkmark icon indicates whether an option has been selected (a check mark appears in the box) or omitted (the box is blank).
- The diamond icon indicates which option has been selected when several mutually exclusive options are offered. It has three states:
  - An unfilled diamond means the option has not been selected.
  - An unfilled diamond within an unfilled diamond means the option has been selected as the default value.

  - A completely filled diamond within a diamond means the option has been selected explicitly by the user (in other words, the user either has selected a value other than the default or has explicitly selected the default value).

- A question mark (?) in a box is a request for confirmation. After you read the information next to the icon, click on the icon. The icon changes to a check mark.

3.11.4 Performing an Operation

The window's lower left button is called the operation button. The text on the button states the type of operation you chose from the Mode menu. After you select products and product options, click on this button to start the operation. As the operation proceeds, a Work in Progress window displays messages and percentage of completeness.

3.11.5 Customizing the DECwindows Motif Interface

You can tailor the DECwindows Motif interface to suit your own environment. For example, to make installation easier for a less experienced installer, you could reduce the number of fields or options available, or you could make certain features more readily available.

Adding or Removing Menu Items

To add or remove items on any of the menus, except the Customize and Help menus, follow these steps:

1. Pull down the Customize menu.
2. Choose the name of the menu you want to change. A dialog box opens.
3. Make selections in the dialog box.
4. Click on OK to apply your change. The dialog box closes.

Adding Fields and Filters

To add or remove text-entry fields and filters, use the Qualifiers menu. Adding fields and filters is useful for changing variables for an operation. The changes you can make are:

- Add the Producer filter to select products by manufacturer.
- Add the Base System filter to select products by operating system or hardware configuration.
- Remove fields (for example, to prevent the values from being changed or to simplify the window). You can remove one or more of the following:
  - Product/Version filter
  - Producer/Base System filter
- Add the Read Configuration field to specify the name of a product configuration file (PCF) for an installation.
- Add the Write Configuration field to specify the name of the PCF you want to create.
- Add a Remark field to enter a comment in the product database.

You can also use the Customize Qualifiers dialog box to customize fields and filters in the main window. By entering values in the fields provided, you set the default values for those fields.

Controlling Site-Specific Installations

You can use the Customize Qualifiers... dialog box to create a controlled environment for software installers. For example, if you want to discourage others from changing the values you set in the Producer, Source, and Destination fields, do the following:

1. Choose Customize Qualifiers... from the Customize menu.
2. Specify values to set in the Producer, Source, and Destination fields.
3. Disable the fields by toggling both buttons to the left of the fields.

   Disabling the buttons prevents the fields from being listed on the Qualifiers menu or displayed in the main window. The fields are ‘‘hidden’’ and, therefore, are less likely to be changed inadvertently.
4. Click on OK to apply your change. The dialog box closes.
5. Save your settings by choosing Save Settings from the Customize menu.

Saving or Resetting the Changes

By default, the changes you make remain active only during the current session. After you exit from the POLYCENTER Software Installation utility, the settings return to their default values. To preserve a change from one session to the next, choose Save Settings from the Customize menu.

If you want to discard your changes and return the settings to their previous states, use the Reset button.

4

Starting Up and Shutting Down the System

This chapter describes various ways to start up and shut down your system.

To initiate startup of your system, you **boot** it. Many systems have unique booting commands. For detailed booting instructions for your system, see the following manuals:

- On VAX systems, see the upgrade and installation supplement for your VAX computer.
- On Alpha systems, see the *Upgrade and Installation Manual*.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Booting with modified system parameter values	Section 4.2
Booting in an emergency	Section 4.3
Booting with controlled startup	Section 4.4
Solving booting problems	Section 4.5
Writing a new boot block on the system disk	Section 4.6
† Using the Snapshot facility to reboot nonclustered systems	Section 4.7
Performing an orderly shutdown with SHUTDOWN.COM	Section 4.8.1
Performing an orderly shutdown with SYSMAN	Section 4.8.4
Performing an emergency shutdown with the OPCCRASH.EXE program	Section 4.8.5
Performing an emergency shutdown using console commands	Section 4.8.6

†VAX specific

This chapter explains the following concepts:

Concept	Section
Booting and startup processes	Section 4.1.1
Nonstop boot: the most common booting operation	Section 4.1.2.1
Conversational boot: for special booting functions	Section 4.1.2.2
System startup and STARTUP.COM	Section 4.1.3
System shutdown procedures	Section 4.8
The order of shutdown events	Section 4.8.2

4.1 Understanding Booting and System Startup

Booting is the process of loading system software from the system disk into processor memory. When you boot your system, it automatically performs a series of tasks to start up your system. These tasks are collectively known as **system startup**.

You must have installed the operating system before you boot the system for the first time.

Booting procedures vary for different computers. For example, computers with console storage devices use a boot command procedure. You can copy and edit this command procedure to specify the location of the system disk. Other computers have an internal memory device that provides the name of the system disk.

Alpha

On Alpha systems, you cannot boot from a magnetic tape device. ♦

4.1.1 Booting and Startup Processes

Together, the booting and startup processes comprise the following steps:

1. You enter the BOOT command. The boot block points to the **primary bootstrap image**, which is loaded from disk into main memory.

   VAX

   On VAX systems, the primary bootstrap image is VMB.EXE. ♦

   Alpha

   On Alpha systems, the primary bootstrap image is APB.EXE. ♦

   The primary bootstrap image allows access to the system disk by finding the **secondary bootstrap image**, SYS$SYSTEM:SYSBOOT.EXE, and loading it into memory.

2. SYSBOOT.EXE loads the system parameters stored in the default parameter file into memory. (For more information on the default parameter file and loading of system parameters at boot time, see Section 4.2.)

   If you are performing a conversational boot, the procedure stops and displays the SYSBOOT> prompt. (For information on conversational booting, see Section 4.1.2.2.) Otherwise, SYSBOOT.EXE loads the operating system executive into memory and transfers control to the executive.

3. When the executive finishes, it executes the SWAPPER process.

4. The SWAPPER creates the SYSINIT process.

5. SYSINIT creates the STARTUP process.

6. STARTUP executes SYS$SYSTEM:STARTUP.COM (unless you indicated another file using SYSMAN, SYSGEN, or conversational boot). STARTUP.COM executes a series of other startup command procedures, including SYSTARTUP_VMS.COM. (For more information on STARTUP.COM, see Section 4.1.3. For more information on other startup procedures, see Section 5.2.1.)

   The current values of system parameters are written back to the default parameter file.

7. The boot process finishes, and you can log in to the operating system.

4.1.2 Types of Booting Operations

You can perform the following types of booting operations:

Type	Purpose	For More Information
Nonstop boot	To boot without stopping to perform special operations. Use this kind of boot in most cases.	Section 4.1.2.1
Conversational boot	To perform special boot operations—for example, to change system parameters before booting.	Section 4.1.2.2

4.1.2.1 Nonstop Boot: The Most Common Booting Operation

The most common boot operation is a nonstop boot from the system disk. You perform a **nonstop boot** if you do not want to stop to perform special operations—for example, to change system parameter values—before booting.

You would perform a nonstop boot after shutting down a system to perform hardware maintenance. Or, if the system disk drive were broken and you had a copy of the operating system on an alternate system disk, you would perform a nonstop boot from the alternate system disk.

Follow the instructions for a nonstop boot in either of the following manuals:

- On VAX systems, see the upgrade and installation supplement for your VAX computer.
- On Alpha systems, see the *Upgrade and Installation Manual*.

4.1.2.2 Conversational Boot: For Special Booting Functions

A **conversational boot** is used in programming research and development environments where you must alter operating conditions for experimentation, testing, and debugging. You use a conversational boot to perform the following operations:

Operation	For More Information
Boot after showing or modifying individual system parameter values.[1]	Section 4.2.1
Boot using system parameter values from an alternate parameter file.[1]	Section 4.2.2
Boot with default values for system parameters; for example, when modified system parameter values have caused the system to become unbootable.[1]	Section 4.3.1
Boot without running startup or login procedures; for example, when modified startup or login procedures have caused the system to become unbootable.	Section 4.3.2
Boot without the user authorization file; for example, when the user authorization file has been modified so that you cannot log in.	Section 4.3.3
Boot with an alternate site-independent startup procedure.	Section 4.4.1
Boot with a minimum startup.	Section 4.4.3

[1]In most cases, Digital recommends that you use AUTOGEN to modify system parameters. In special cases, however, you can use a conversational boot to modify a parameter value *temporarily*. To change a parameter value *permanently*, you must edit MODPARAMS.DAT and run AUTOGEN. For instructions, see Section 14.5.

Operation	For More Information
Display startup procedure commands while booting.	Section 4.4.4

To boot your system conversationally, follow the instructions for a conversational boot in either of the following manuals:

- On VAX systems, see the upgrade and installation supplement for your VAX computer.
- On Alpha systems, see the *Upgrade and Installation Manual*.

4.1.3 System Startup and STARTUP.COM

Immediately after your system boots, it runs the **site-independent command procedure** SYS$SYSTEM:STARTUP.COM to start up the system and control the sequence of startup events. This section describes STARTUP.COM.

Caution

Do not modify SYS$SYSTEM:STARTUP.COM. This file is deleted and replaced each time you upgrade your system to the next version of the 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.

Although you should not modify STARTUP.COM, sometimes you may want to control site-independent startup when booting your system. For information, see Section 4.4.

STARTUP.COM uses a series of command procedures, executable images, and database files to perform the following startup tasks:

- Define systemwide logical names required for the symbolic debugger, language processors, linker, image activator, and help processor.
- Start processes that control error logging, SMISERVER (the system management server), the job controller, the operator log file, and security auditing.
- Connect devices that are physically attached to the system by invoking the SYCONFIG.COM procedure. Configure devices and load their I/O drivers.
- Install known images to reduce I/O overhead in activating the most commonly run images or to identify images that must have special privileges.

STARTUP.COM executes the following **site-specific** startup command procedures in this order:

1. SYS$MANAGER:SYCONFIG.COM
2. SYS$MANAGER:SYLOGICALS.COM
3. SYS$MANAGER:SYPAGSWPFILES.COM
4. SYS$MANAGER:SYSECURITY.COM
5. SYS$MANAGER:SYSTARTUP_VMS.COM

For information on site-specific startup command procedures, see Section 5.2.

4.1.4 Messages Indicating Booting and Startup Progress

When you successfully boot a system, it displays messages similar to the following:

1. The following message indicates that the system is executing the command procedure SYS$SYSTEM:STARTUP.COM:

   ```
   The OpenVMS system is now executing the system startup procedure.
   ```

   This procedure configures and initializes the system and executes several site-specific command procedures. For more information, see Section 4.1.3.

2. A short time later (up to a few minutes), the system displays a message similar to the following:

   ```
   The OpenVMS system is now executing the site-specific system startup commands.
   ```

   This message indicates that the system is executing SYSTARTUP_VMS.COM. You can modify this file to perform various operations at startup time. For more information, see Section 5.2.7.

3. Finally, the procedure displays informational messages and accounting information. For example:

   ```
   %SET-I-INTSET, login interactive limit=64, current interactive value = 0
   19-APR-1995 15:00:00.00
     SYSTEM       job terminated at 19-APR-1995 15:00:00.00

   Accounting information:
    Buffered I/O count:       133     Peak working set size:        401
    Direct I/O count:          12     Peak pagefile size:          2379
    Page faults:              325     Mounted volumes:                0
    Charged CPU time: 0 00:00:55.23   Elapsed time:       0 00:01:31.24
   ```

   After the system displays this information, you can log in.

Spurious Message During Booting

You may see the following message on some systems during booting. Contrary to the message, the system booted correctly.

```
%INSTALL-I-NONRES, installed image non-resident with other specified
options
-INSTALL-E-NOGHREG, insufficient memory in the code or data granularity hint
region
```

You can ignore the message or eliminate it using AUTOGEN. The message means that the ADD/RESIDENT command of the Install utility (INSTALL) caused image code sections or read-only data sections to be placed in the granularity hints code region.

4.2 Booting with Modified System Parameter Values

Using a conversational boot, you can modify system parameter values as follows:

Task	For More Information
Boot after showing or modifying individual system parameter values	Section 4.2.1
Boot with an alternate system parameter file	Section 4.2.2
Boot with default values for system parameters	Section 4.3.1

Before using a conversational boot to show or modify system parameter values, you must be familiar with the following terms:

Term	Definition
Active values	System parameter values stored in memory and used by the active system.
Current values	System parameter values stored in the default parameter file. When the system boots, it sets active values for system parameters using the current values. †On VAX systems, the default system parameter file is SYS$SYSTEM:VAXVMSSYS.PAR. ‡On Alpha systems, the default system parameter file is SYS$SYSTEM:ALPHAVMSSYS.PAR.
Default values	System parameter values stored in the default list and used by default.

†VAX specific
‡Alpha specific

For more information on system parameters, see Section 14.1.

4.2.1 Booting After Showing or Modifying Individual System Parameter Values

In a conversational boot operation, you can show and modify values for individual parameters.[1] The system modifies the values both in memory and in the system parameter file.

How to Perform This Task

1. Follow the instructions for performing a conversational boot in one of the following manuals:
   - On VAX systems, see the upgrade and installation supplement for your VAX computer.
   - On Alpha systems, see the *Upgrade and Installation Manual*.
2. At the SYSBOOT> prompt, enter SHOW and SET commands to show and change the value of system parameters. For example:

   ```
   SYSBOOT> SET UAFALTERNATE 1
   ```

   For information about SET and SHOW commands, see the *OpenVMS System Management Utilities Reference Manual* (SYSGEN).
3. After you show or set the necessary parameter values, enter the CONTINUE command to continue booting.

Example

```
SYSBOOT> SHOW UAFALTERNATE
Parameter Name            Current    Default     Min.     Max.     Unit
Dynamic
--------------            -------    -------    -------  -------   ---- -------
UAFALTERNATE                    0          0         0         1   Boolean
SYSBOOT> SET UAFALTERNATE 1
SYSBOOT> CONTINUE
```

[1] In most cases, Digital recommends that you use AUTOGEN to modify system parameters. In special cases, however, you can use a conversational boot to modify a parameter value *temporarily*. To change a parameter value *permanently*, you must edit MODPARAMS.DAT and run AUTOGEN. For instructions, see Section 14.5.

4.2.2 Booting with an Alternate System Parameter File

In programming research and development environments where you must alter operating conditions for experimentation, testing, and debugging, you might want to temporarily boot your system using system parameter values stored in a parameter file other than the default parameter file. The conversational boot operation lets you reset active values using a different parameter file.[1]

How to Perform This Task

1. Follow the instructions for performing a conversational boot in one of the following manuals:
   - On VAX systems, see the upgrade and installation supplement for your VAX computer.
   - On Alpha systems, see the *Upgrade and Installation Manual*.
2. At the SYSBOOT> prompt, enter a command in the following format:

   USE file-spec

   For *file-spec*, specify the file name and type of the alternate parameter file. The file must be in SYS$SYSTEM. You cannot specify a device name. For example:

   ```
   SYSBOOT> USE ALTPARAMS.DAT
   ```
3. Enter the CONTINUE command to continue booting.

Example

```
SYSBOOT> USE ALTPARAMS.DAT
SYSBOOT> CONTINUE
```

4.3 Booting in an Emergency

If a system problem prevents your system from booting, you might need to perform an emergency boot operation. Table 4–1 describes these emergency boot operations.

Table 4–1 Emergency Boot Procedures

Operation	Use	For More Information
Booting with default system parameters	When parameter values in the parameter file have been modified so that the system is unbootable	Section 4.3.1
Booting without startup and login procedures	If an error in the startup or login procedures prevent you from logging in	Section 4.3.2
Booting without the user authorization file	If you have forgotten the password and cannot log in to a privileged account	Section 4.3.3

4.3.1 Booting with Default System Parameters

If the current values stored in the parameter file have been incorrectly modified, these incorrect values might cause the system to become unbootable. With a conversational boot operation, you can reset the active values for all system parameters to the default value.[1]

How to Perform This Task

1. Perform a conversational boot by following the instructions in one of the following manuals:
   - On VAX systems, see the upgrade and installation supplement for your VAX computer.
   - On Alpha systems, see the *Upgrade and Installation Manual*.
2. At the SYSBOOT> prompt, enter the following command:

   ```
   SYSBOOT> USE DEFAULT
   ```

   This command specifies that default values should be used for all parameters.
3. Enter the following command to continue booting:

   ```
   SYSBOOT> CONTINUE
   ```
4. When the system finishes booting, determine which changed parameter caused the problem, and reset the parameter value. If you specified the value for the parameter in the AUTOGEN parameter file MODPARAMS.DAT, fix the value in that file and run AUTOGEN. For more information, see Section 14.5.
5. Shut down and reboot the system.

Example

```
SYSBOOT> USE DEFAULT
SYSBOOT> CONTINUE
Username: SYSTEM
Password:

$ EDIT SYS$SYSTEM:MODPARAMS.DAT
   .
   .
   .
[Insert the following line in MODPARAMS.DAT:]
MIN_NPAGEDYN = 2999808
   .
   .
   .
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

4.3.2 Booting Without Startup and Login Procedures

If the system does not complete the startup procedures or does not allow you to log in, bypass the startup and login procedures. The startup and login procedures provided by Digital should always work. However, if you introduce an error when modifying the startup or login procedures, it is possible to accidentally lock yourself out of the system.

[1] In most cases, Digital recommends that you use AUTOGEN to modify system parameters. In special cases, however, you can use a conversational boot to modify a parameter value *temporarily*. To change a parameter value *permanently*, you must edit MODPARAMS.DAT and run AUTOGEN. For instructions, see Section 14.5.

How to Perform This Task

1. Perform a conversational boot operation by following the instructions in one of the following manuals:
   - On VAX systems, see the upgrade and installation supplement for your VAX computer.
   - On Alpha systems, see the *Upgrade and Installation Manual*.
2. Enter the following command at the SYSBOOT> prompt:

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

3. Enter the following command to continue booting:

   ```
   SYSBOOT> CONTINUE
   ```

4. When the system is booted, the operator console displays the DCL command prompt ($). You are logged in.
5. Enter the following DCL command:

   ```
   $ SET NOON
   ```

   This command directs the operating system to ignore any errors that might occur. If you do not enter this command and you invoke an error, the system will log you out.
6. 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.

   Invoke a text editor to correct the file. Note that some system consoles might 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.
7. Invoke SYSMAN and enter the following commands to reset the startup procedure:

   ```
   $ RUN SYS$SYSTEM:SYSMAN
   SYSMAN> PARAMETERS USE CURRENT
   SYSMAN> PARAMETERS SET/STARTUP SYS$SYSTEM:STARTUP.COM
   SYSMAN> PARAMETERS WRITE CURRENT
   SYSMAN> EXIT
   $
   ```

8. Perform a normal startup by entering the following command:

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

Example

```
SYSBOOT> SET/STARTUP OPA0:
SYSBOOT> CONTINUE
$ SET NOON
$ SET DEFAULT SYS$SYSROOT:[SYSEXE]
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAMETERS USE CURRENT
SYSMAN> PARAMETERS SET/STARTUP SYS$SYSTEM:STARTUP.COM
SYSMAN> PARAMETERS WRITE CURRENT
SYSMAN> EXIT
$ @SYS$SYSTEM:STARTUP
```

4.3.3 Booting Without the User Authorization File

Ordinarily, the startup and login procedures provided by Digital always work; however, certain user interventions can cause them to fail. A very simple way to lock yourself out of the system is to set passwords to login accounts and forget them. In such an emergency, you can use the alternate user authorization file rather than the standard user authorization file.

Note

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

Setting the system parameter UAFALTERNATE 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 it cannot find this file, the system assumes that the UAF is corrupt and accepts any user name and any two passwords to log you in to the system from the system console. Logins are prohibited from all other terminal lines.

When you perform this procedure, the system assigns the following values to your user account:

Field	Value
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 set to the name of the device on which you logged in (for example, _OPA0:).

How to Perform This Task

1. Perform a conversational boot by following the instructions in one of the following manuals:
   - On VAX systems, see the upgrade and installation supplement for your VAX computer.
   - On Alpha systems, see the *Upgrade and Installation Manual*.
2. At the SYSBOOT> prompt, enter the following command:

   ```
   SYSBOOT> SET UAFALTERNATE 1
   ```

3. If your system is running DECwindows Motif for OpenVMS systems, you must also disable the windowing system by entering the following command:

   ```
   SYSBOOT> SET WINDOW_SYSTEM 0
   ```

4. Enter the CONTINUE command to continue booting:

```
SYSBOOT> CONTINUE
```

5. When the startup procedure completes, log in on the console terminal by entering any user name and any two passwords in response to the *Username:* and *Password:* prompts.

6. Enter the following command to use the default UAF:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE SYSUAF SYS$SYSTEM:SYSUAF.DAT
```

7. Use the Authorize utility to fix the problem that caused you to be locked out of the system (for example, a forgotten password). Enter HELP MODIFY at the UAF> prompt for information about modifying passwords. For more details, see the *OpenVMS System Management Utilities Reference Manual*.

8. Enter the following commands to invoke SYSMAN and clear the UAFALTERNATE system parameter you set in step 2:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAMETERS USE CURRENT
SYSMAN> PARAMETERS SET UAFALTERNATE 0
```

   In most cases, Digital recommends that you use AUTOGEN to modify system parameters. However, since this parameter is being changed only temporarily, you can use SYSMAN or SYSGEN to change it back.

9. If you disabled the windowing system in step 3, reenable it by entering the following command:

```
SYSMAN> PARAMETERS SET WINDOW_SYSTEM 1
```

10. Enter the following command to save the changed system parameter values:

```
SYSMAN> PARAMETERS WRITE CURRENT
```

11. Shut down and reboot the system.

Example

```
SYSBOOT> SET UAFALTERNATE 1
SYSBOOT> SET WINDOW_SYSTEM 0
SYSBOOT> CONTINUE
Username:  Return
Password:  Return

Password:  Return

$ DEFINE/SYSTEM/EXECUTIVE_MODE SYSUAF SYS$SYSTEM:SYSUAF.DAT
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
AUTHORIZE> MODIFY SYSTEM/PASSWORD=FGLFTUTU
AUTHORIZE> EXIT

$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAMETERS USE CURRENT
SYSMAN> PARAMETERS SET WINDOW_SYSTEM 1
SYSMAN> PARAMETERS SET UAFALTERNATE 0
SYSMAN> PARAMETERS WRITE CURRENT

SYSMAN> EXIT
$ @SYS$SYSTEM:SHUTDOWN
```

4.4 Booting with Controlled Startup

Section 4.1.3 explains the site-independent startup command procedure, SYS$SYSTEM:STARTUP.COM. By default, when your system boots, it automatically executes STARTUP.COM to execute startup events. Under special circumstances, you might want to control site-independent startup when you boot the system. For example, you might want to perform one of the following tasks:

Task	For More Information
Boot with an alternate site-independent startup procedure	Section 4.4.1
Boot with an alternate site-independent startup command procedure by default	Section 4.4.2
Boot with minimum startup	Section 4.4.3
Display startup procedure commands as they execute	Section 4.4.4

Caution

Do not modify STARTUP.COM. The system requires this procedure to correctly start up the system. For information on modifying *site-specific* startup procedures to perform site-specific operations, see Section 5.2.

4.4.1 Booting with an Alternate Site-Independent Startup Procedure

The default system startup procedure is SYS$SYSTEM:STARTUP.COM. Digital recommends you do not modify STARTUP.COM. However, in special environments, you might want the system to perform special startup commands. The conversational boot lets you specify that the system temporarily use an alternate startup procedure.

You can also perform site-specific startup events by adding commands to the *site-specific* startup command procedures. For more information, see Section 5.2.

How to Perform This Task

1. Follow the instructions for performing a conversational boot in one of the following manuals:
   - On VAX systems, see the upgrade and installation supplement for your VAX computer.
   - On Alpha systems, see the *Upgrade and Installation Manual*.
2. Enter the following command to show the current startup file:

   ```
   SYSBOOT> SHOW/STARTUP
   ```

3. Enter a command in the following format to specify the alternate site-independent startup command procedure:

   SET/STARTUP file-spec

   Specify for *file-spec* the entire file specification for the startup file to be used, including the device and directory. For example:

   ```
   SYSBOOT> SET/STARTUP SYS$SYSTEM:XSTARTUP.COM
   ```

If the startup file specified as *file-spec* does not exist, the system displays the following message:

```
Error opening primary input file SYS$INPUT
File not found
```

Check the file name you entered. Make sure you specified it correctly.

4. Enter the following command to verify the change:

```
SYSBOOT> SHOW/STARTUP
```

5. Enter the following command to continue booting:

```
SYSBOOT> CONTINUE
```

To make your alternate site-independent startup procedure the default startup procedure, see Section 4.4.2.

Example

```
SYSBOOT> SHOW/STARTUP
Startup command file = SYS$SYSTEM:STARTUP.COM
SYSBOOT> SET/STARTUP SYS$SYSTEM:XSTARTUP.COM
SYSBOOT> SHOW/STARTUP
Startup command file = SYS$SYSTEM:XSTARTUP.COM
SYSBOOT> CONTINUE
```

4.4.2 Specifying an Alternate Default Startup Command Procedure

The default system startup procedure is SYS$SYSTEM:STARTUP.COM. However, in special environments, you might want the system to perform special startup commands. If you frequently require a startup command procedure other than SYS$SYSTEM:STARTUP.COM, you can specify that the alternate procedure be used by default.

How to Perform This Task

1. Edit the file SYS$SYSTEM:MODPARAMS.DAT. AUTOGEN uses this file to modify parameters.
2. Add a line to MODPARAMS.DAT assigning the name of your alternate procedure to the symbol STARTUP. For example:

```
STARTUP = "SYS$SYSTEM:MY_STARTUP.COM"
```

3. At a convenient time, invoke AUTOGEN. When the system reboots, the procedure specified in step 2 becomes the default startup command procedure.

Example

```
$ EDIT SYS$SYSTEM:MODPARAMS.DAT
   .
   .
   .
[Insert the following line in MODPARAMS.DAT:]
STARTUP = "SYS$SYSTEM:MY_STARTUP.COM"
   .
   .
   .
$ @SYS$SYSTEM:AUTOGEN SAVPARAMS REBOOT
```

4.4.3 Booting with Minimum Startup

In special cases, you might want to boot your system without performing the full sequence of startup events. For example, if a startup event prevents you from logging in, you might want to boot the system without executing the startup, so that you can log in and fix the problem.

When you boot with minimum startup, the system starts only the components that are absolutely required to run the system. These tasks can vary between different releases of the operating system.

How to Perform This Task

1. Follow the instructions for performing a conversational boot in one of the following manuals:
   - On VAX systems, see the upgrade and installation supplement for your VAX computer.
   - On Alpha systems, see the *Upgrade and Installation Manual*.
2. At the SYSBOOT> prompt, enter the following command:

   ```
   SYSBOOT> SET STARTUP_P1 "MIN"
   ```

3. Enter the following command to continue booting:

   ```
   SYSBOOT> CONTINUE
   ```

4. After the system boots, log in and enter the following commands to invoke SYSMAN and clear the STARTUP_P1 parameter you set in step 2:

   ```
   $ RUN SYS$SYSTEM:SYSMAN
   SYSMAN> PARAMETERS USE CURRENT
   SYSMAN> PARAMETERS SET STARTUP_P1 ""
   SYSMAN> PARAMETERS WRITE CURRENT
   ```

Example

```
[perform a conversational boot]
SYSBOOT> SET STARTUP_P1 "MIN"
SYSBOOT> CONTINUE
[system completes booting]
Username: Return
Password: Return

$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAMETERS USE CURRENT
SYSMAN> PARAMETERS SET STARTUP_P1 ""
SYSMAN> PARAMETERS WRITE CURRENT
```

4.4.4 Booting While Displaying Startup Procedure Commands

In some cases—for example, when you are trying to test a startup command procedure, or when troubleshooting startup problems—it is helpful to display the startup commands as they are executed.

How to Perform This Task

1. Follow the instructions for performing a conversational boot in one of the following manuals:
   - On VAX systems, see the upgrade and installation supplement for your VAX computer.
   - On Alpha systems, see the *Upgrade and Installation Manual*.

2. At the SYSBOOT> prompt, enter the following command:

```
SYSBOOT> SET STARTUP_P2 "YES"
```

3. Enter the following command to continue booting:

```
SYSBOOT> CONTINUE
```

4. After the system boots, log in and enter the following commands to invoke SYSMAN and clear the STARTUP_P2 parameter you set in step 2:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAMETERS USE CURRENT
SYSMAN> PARAMETERS SET STARTUP_P2 ""
SYSMAN> PARAMETERS WRITE CURRENT
```

Example

```
[perform a conversational boot]
SYSBOOT> SET STARTUP_P2 "YES"
SYSBOOT> CONTINUE
[system completes booting]
Username: Return
Password: Return

$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAMETERS USE CURRENT
SYSMAN> PARAMETERS SET STARTUP_P2 ""
SYSMAN> PARAMETERS WRITE CURRENT
```

4.4.5 Displaying Startup Procedure Commands with SYSMAN

In addition to performing a conversational boot to display startup procedures, you can use SYSMAN to display startup status with the STARTUP SET OPTIONS command. The advantage of using SYSMAN is that you can obtain verification and logging for multiple nodes at a time.

SYSMAN startup logging redefines STARTUP_P2 to specify:

- The amount of debugging information STARTUP.COM displays
- Whether to keep a log of the startup

The STARTUP SET OPTIONS command provides four options, as shown in Table 4–2.

Table 4–2 Startup Logging Options

Option	Function
/VERIFY=FULL	Displays every line of DCL executed by component startup procedures and by STARTUP.COM.
/VERIFY=PARTIAL	Displays every line of DCL executed by component startup procedures, but does not display DCL executed by STARTUP.COM.
/OUTPUT=FILE /OUTPUT=CONSOLE	Creates SYS$SPECIFIC:[SYSEXE]STARTUP.LOG, which contains all of the output generated by startup procedures. Alternatively, you can display the output on the console.
/CHECKPOINTING	Displays informational messages describing the time and status of each startup phase and component file.

How to Perform This Task

1. At the $ prompt, enter the following command:

   ```
   $ RUN SYS$SYSTEM:SYSMAN
   ```

2. At the SYSMAN> prompt, enter the following command:

   ```
   SYSMAN> STARTUP SET OPTIONS/[qualifier]
   ```

   Qualifiers can be any of the options specified in Table 4–2. These options take effect the next time you boot the system.

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> STARTUP SET OPTIONS/VERIFY=FULL/OUTPUT=FILE/CHECKPOINTING
```

This example requests startup logging with:

Full verification
Output to the STARTUP.LOG file
Checkpointing

To show the current startup options, enter the following command:

```
SYSMAN> STARTUP SHOW OPTIONS
```

For more information, see the *OpenVMS System Management Utilities Reference Manual*.

4.5 Solving Booting Problems

A hardware or software malfunction can prevent the operating system from booting when you enter the BOOT command.

Hardware Problems

A read error on a disk drive or console medium, or a machine check error, might 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 one or both of the following:

1. Consult the hardware manual for your computer.
2. Contact your Digital representative.

Software Problems

If the operating system is loaded into memory but the STARTUP.COM command procedure does not execute, a software malfunction has probably occurred. Suspect this condition if a message similar to following does not appear:

```
The OpenVMS system is now executing the system startup procedure.
```

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

- Try again, by repeating the boot procedure. For instructions, see one of the following manuals:
  - On VAX systems, see the upgrade and installation supplement for your VAX computer.
  - On Alpha systems, see the *Upgrade and Installation Manual*.

  If you have a removable system disk, replace it with a backup copy of the system disk. Try to boot the system again.

- Leave the system disk in the original drive. Restore a backup copy of the system disk. For instructions, see Section 10.17. Try to boot the system again.

4.6 Writing a New Boot Block on the System Disk

Block 0 on a system disk is the **boot block**. It contains the size and location of the primary bootstrap image, which is used to boot the system.

VAX

On VAX systems, the primary bootstrap image is VMB.EXE. ♦

Alpha

On Alpha systems, the primary bootstrap image is APB.EXE. ♦

Certain processors must read the boot block to obtain the location of the primary bootstrap image. Processors that read a boot block include the following:

- VAX–11/750
- VAX 8200, 8250, 8300, and 8350
- VAX 6000–200, 6000–300, 6000–400, 6000–500, and 6000–600
- VAX 7000 and VAX 10000
- All Alpha systems (subject to change for future systems)

To determine if your system reads the boot block, check one of the following manuals:

- On VAX systems, see the upgrade and installation supplement for your VAX computer.
- On Alpha systems, see the *Upgrade and Installation Manual*.

If you suspect that the boot block on the system disk is invalid, you can write a new boot block using the Writeboot utility (WRITEBOOT). The following actions might cause a boot block to become invalid:

- Modifying the primary bootstrap image with the SET FILE/MOVE command or the $MOVEFILE system service.
- Restoring a backup of the system disk created without the /IMAGE qualifier.
- Adding a new version of the primary bootstrap image, for example, during an operating system upgrade. (When the upgrade procedure adds a new version of the primary bootstrap image, it automatically uses WRITEBOOT to write a new boot block.)
- Adding a new version of the primary bootstrap image that is not contiguous on disk. (Use the DIRECTORY/FULL command to determine this.)

You must have LOG_IO privilege to use the Writeboot utility.

How to Perform This Task

VAX

On VAX systems, follow these steps to use the Writeboot utility:

1. To start the Writeboot utility, enter the following command:

   ```
   $ RUN SYS$SYSTEM:WRITEBOOT
   ```

2. The procedure displays the following message:

   ```
   Target system device (and boot file if not VMB.EXE):?
   ```

   On VAX systems, VMB.EXE is the default bootstrap image. Enter a response in the following format:

device:[VMS$COMMON.SYSEXE]VMB.EXE;

Use the device name format described in the upgrade and installation documentation for your processor. If you want to boot using a bootstrap image other than the default, you must specify the full file specification of the image, including device and directory.

3. The procedure displays the following message:

   ```
   Enter VBN of boot file code (default is one):
   ```

   Ordinarily, the boot code is located at virtual block number (VBN) 1 of the bootstrap image. Press Return to accept the default value of 1.

4. The procedure displays the following message:

   ```
   Enter load address of primary bootstrap in HEX (default is 200):
   ```

   The **load address** is the location in memory (specified in hexadecimal notation) to which the system loads the bootstrap image. Ordinarily you copy the bootstrap image to address 200. Press Return to accept the default value of 200.

5. The Writeboot utility writes the information you specified to the boot block (block 0) on the system disk. ♦

Alpha

On Alpha systems, follow these steps to use the Writeboot utility:

1. To start the Writeboot utility, enter the following command:

   ```
   $ RUN SYS$SYSTEM:WRITEBOOT
   ```

   The procedure asks you whether you want to write the VAX portion of the boot block:

   ```
   Update VAX portion of boot block (default is Y):
   ```

2. Enter NO.

3. The utility displays the following prompt:

   ```
   Update Alpha portion of boot block (default is Y):
   ```

   Press Return to accept the default value of Y.

4. The utility prompts you for the Alpha bootstrap image:

   ```
   Enter Alpha boot file:
   ```

   On Alpha systems, APB.EXE is the default bootstrap image. Enter a response in the following format:

   device:[VMS$COMMON.SYSEXE]APB.EXE;

   For *device*, specify the device name of the system disk.

5. The Writeboot utility writes the information you specified to the boot block (block 0) on the system disk. ♦

On VAX and Alpha systems, the Writeboot utility might display one or more of the following error messages:

- ''You lack LOG_IO privilege.''

  This message means you do not have the correct privilege to use the Writeboot utility.

- ''You lack READ and/or WRITE access to TARGET DEVICE. DISMOUNT and reMOUNT it.''

  This message means that access to the target device is limited. Check the WRITE PROTECT button on the disk drive.

- ''Boot file is not contiguous.''

  This message means that the primary bootstrap image, VMB.EXE or APB.EXE, is not contiguous on disk. Do one of the following:

  VAX

  – On VAX systems, enter the following command:

    ```
    $ COPY/CONTIGUOUS device:[VMS$COMMON.SYSEXE]VMB.EXE -
    _$  device:[VMS$COMMON.SYSEXE]
    ```

    This command copies the VMB.EXE file contiguously to the next higher version number. ♦

  Alpha

  – On Alpha systems, enter the following command:

    ```
    $ COPY/CONTIGUOUS device:[VMS$COMMON.SYSEXE]APB.EXE -
    _$  device:[VMS$COMMON.SYSEXE]
    ```

    This command copies the APB.EXE file contiguously to the next higher version number. ♦

  Rewrite the boot block for the new image by running WRITEBOOT again.

  VAX

- On VAX systems, ''VBN must be >= 1.''

  This message means you cannot specify a 0 as the virtual block number (VBN). ♦

Example

VAX

On VAX systems, the following example writes a boot block on a system disk:

```
$ RUN SYS$SYSTEM:WRITEBOOT
Target system device (and boot file if not VMB.EXE):? DUA0:[VMS$COMMON.SYSEXE]VMB.EXE
Enter VBN of boot file code (default is one):Return
Enter load address of primary bootstrap in HEX (default is 200): Return ♦
```

Alpha

On Alpha systems, the following example writes a boot block on a system disk:

```
$ RUN SYS$SYSTEM:WRITEBOOT
Update VAX portion of boot block (default is Y): N
Update Alpha portion of boot block (default is Y): Return
Enter Alpha boot file: DUA0:[VMS$COMMON.SYSEXE]APB.EXE; ♦
```

4.7 Using the Snapshot Facility to Reboot Nonclustered Systems (VAX Only)

VAX

The OpenVMS Snapshot facility (Snapshot) lets you reduce system startup time by booting your system from a saved system image disk file. Snapshot eliminates the need to execute many of the procedures required by the normal startup process, as well as the time needed for DECwindows login.

You can use Snapshot only for VAX workstations that are not in a VMScluster environment. Snapshot is especially well suited for single-user workstations.

Restriction and Workaround

Snapshot is not supported in VAXstation 4000 Model 60 and Model 90 systems with the SPX/g or SPX/gt graphic subsystem (GECDRIVER).

As a workaround, you can do one of the following:

- Do not use the Snapshot feature.
- Use the Snapshot feature and follow these steps:
  1. Disable DECwindows startup.
  2. Take the snapshot before the GECDRIVER is loaded.
  3. Start up DECwindows in the SYS$MANAGER:SNAPSHOT$SYCLEANUP.COM procedure.

4.7.1 Preparing Your Workstation for a System Image Snapshot (VAX Only)

To use Snapshot, you take a system image snapshot that can be used for subsequent rebooting. First, you must prepare your workstation for a system image snapshot.

How to Perform This Task

1. Ensure that account quotas are set to sufficient levels.

   Take the system image snapshot from the SYSTEM account. Make sure the following quotas for the SYSTEM account are met or exceeded:

   - A WSDEF quota of at least 1024 blocks
   - A WSQUO quota of at least 1024 blocks

   To determine the current quotas for the SYSTEM account and to modify the current quotas, see Chapter 6.

2. Modify your site-specific startup procedures to conform to the requirements of the Snapshot facility.

   The Snapshot facility supports a finite number of applications and components on your system. If you want to use Snapshot, you must be sure that the site-specific startup file, SYS$MANAGER:SYSTARTUP_VMS.COM, includes references to only those features that Snapshot supports. Features that are unsupported by Snapshot should be addressed in a separate file, SYS$MANAGER:SNAPSHOT$SYCLEANUP.COM.

   The following functions are supported by the Snapshot facility:

   - Starting batch and print queues without special symbionts. Note, however, that these queues are temporarily stopped when you create a snapshot.
   - Defining logical names.
   - Installing images that are not revised between reboots.
   - Starting up DECnet Phase IV.
   - Defining DCL symbols.
   - Loading licenses.
   - Mounting Files–11B disks. CD drives are supported; however, their power-failure initialization sequence is long, and you can realize a faster system restart by dismounting such disks before taking the system image snapshot.

All other operations are unsupported and might cause the snapshot to fail. Some of the most common unsupported operations are as follows:

- Loading user-written drivers
- Loading user-written system images
- Running layered product images
- Mounting LAD, InfoServer, DFS, and disks that are foreign or not Files–11-B

Before attempting to run the Snapshot command procedure, remove the commands that start up any of these operations (as well as commands that start up any other unsupported operations) from SYS$MANAGER:SYSTARTUP_VMS.COM.

Applications that are not supported by Snapshot (and therefore removed from SYS$MANAGER:SYSTARTUP_VMS.COM) should be placed in the post-startup cleanup file, SYS$MANAGER:SNAPSHOT$SYCLEANUP.COM. This file is executed after a system startup that has been generated from a saved system image. The SYS$MANAGER:SNAPSHOT$SYCLEANUP.COM file has the same format as the standard SYS$MANAGER:SYSTARTUP_ VMS.COM file. A template for the cleanup file is available in the file SYS$MANAGER:SNAPSHOT$SYCLEANUP.TEMPLATE.

3. Modify certain parameters.

   For your workstation, follow each of these steps to prepare for a system image snapshot:

   a. Turn off logging in the Session Manager by creating a file named SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM and restart DECwindows, as follows:

   ```
   $ SET DEFAULT SYS$MANAGER
   $ CREATE DECW$PRIVATE_APPS_SETUP.COM
   $ DECW$SESSIONLOG == "FALSE"
   Ctrl/Z
   $ @DECW$STARTUP RESTART
   Restarting the DECwindows Software stops everything displaying on your
   workstation.  Do you really want to restart the DECwindows Software? Y
   ```

   b. Turn off broadcast for the console by entering the following command:

   ```
   $ SET TERMINAL/PERMANENT/NOBROADCAST OPA0:
   ```

   c. Set the SYS$DECDTM_INHIBIT logical name by adding the following line to the file SYS$MANAGER:SYLOGICALS.COM:

   ```
   $ DEFINE /SYSTEM /EXEC SYS$DECDTM_INHIBIT TRUE
   ```

   Note

   Snapshot is valid only for workstations that are not in a VMScluster environment. Therefore, the VAXCLUSTER system parameter must be set to 0 for you to use Snapshot. For information about setting system parameters, see Chapter 14.

4. Eliminate open writable files.

To take a successful snapshot, you must not have any open writable files. You can take a saved image snapshot so that the rebooted system will display either the DECwindows LOGIN screen or the pause screen of an active session.

If you want to take a saved image snapshot with an active paused screen, be sure that the following DECwindows applications are not active in your session because they might have open files:

- Message Window—Avoid specifying *Message Window* for automatic startup
- LinkWorks—Avoid specifying LinkWorks Setup for automatic startup or other applications, such as the Bookreader
- Cardfiler
- DECwindows Mail
- Calendar

Some of these applications can be started after the snapshot reboot by adding them to the SYS$MANAGER:SNAPSHOT$SYCLEANUP.COM file.

4.7.2 Taking the Saved Image Snapshot (VAX Only)

After you have prepared your system as described in Section 4.7.1, you can run the Snapshot command procedure, which takes a system image snapshot and then shuts down and reboots your system.

How to Perform This Task

1. From a DECterm window, set host to 0 and log in to the SYSTEM account, as shown in the following sequence:

```
$ SET HOST 0
Username: SYSTEM
Password:
$
```

2. Now execute the following DCL commands:

```
$ SET DEFAULT SYS$MANAGER
$ @SNAPSHOT
```

   The Snapshot procedure reminds you of some required preparations and asks if you want to continue; if you have completed the steps in the previous section, answer YES.

3. Snapshot creates a detached process to write the snapshot file 60 seconds later and then logs off. At this time, either pause the workstation or exit from the current session. Note that any keystrokes that you type before pausing or exiting are stored in the type-ahead buffer and executed after the system has rebooted.

Note

You must either pause the workstation or detach or log out from any interactive jobs before the system image snapshot is taken. (Without this restriction, anyone with physical access to your workstation could shut down and then reboot the workstation, and then have access to the currently logged-in account.)

Also note that, on DECwindows Motif for OpenVMS systems, unpausing the screen opens SYSUAF.DAT and VMS$MAIL_PROFILE.DATA for a write operation. If this occurs, you cannot attempt to take another snapshot until you start a new session.

---

4. After successful creation of the snapshot file, the system will enter an autoreboot sequence and display the following messages:

```
%SNAPSHOT-I-OPEN, opening the snapshot file
   .
   .
   .
%SNAPSHOT-I-RESTART, restart to follow
```

   If the following message is displayed, then the system image has been saved and the Snapshot procedure is successful:

```
%SNAPSHOT-S-SUCCESS, snapshot file write is complete
```

   If you see these audit messages, close the console window by pressing Ctrl/F2. The Snapshot procedure is complete when the windows have been refreshed.

   If you do not see these messages, the Snapshot procedure was not successful. It might take you several attempts to get a successful system snapshot. See Section 4.7.3 for some hints about solving problems after an unsuccessful attempt to run the Snapshot procedure.

5. Once you have successfully created a snapshot file, you can continue to use it as long as none of the files that were open at the time of the snapshot are deleted or modified.

   Create a SYS$MANAGER:SNAPSHOT$SYCLEANUP.COM from SYS$MANAGER:SNAPSHOT$SYCLEANUP.TEMPLATE and review this procedure to add any desired site-specific restart operations such as the starting of queues and batch jobs. You can modify this procedure after taking a valid system snapshot, and the saved system image will remain valid.

4.7.3 Problem Solving When Using the Snapshot Facility (VAX Only)

The following sections describe and offer solutions for some of the problems that you might encounter with Snapshot.

Diagnosis of the snapshot problem depends on where the snapshot failed. Examine the SYS$MANAGER:SNAPSHOT$CLEANUP.LOG file and look for warning or failure messages.

In most instances when a system image snapshot fails, Snapshot resets the system parameter that defines the boot style to 0 to indicate a traditional boot procedure. However, in some cases this parameter might not be reset to 0. If your system does not reboot using the traditional boot procedure, make sure that the BOOT_STYLE system parameter is set to 0.

To verify and change this parameter, you must initiate a conversational boot, as shown in the following example:

```
>>>B/1 <disk>
SYSBOOT> SET BOOT_STYLE 0
SYSBOOT> CONTINUE
```

Common Snapshot Errors

This section lists some of the errors that you might encounter when trying to set up a system image snapshot for your system.

- Uncertified Images

  The following message indicates that an image active at the time of the system snapshot has not been previously tested with the Snapshot facility:

  ```
  %SNAPSHOT-W-NORESTART, process <pid> is not restartable and can not
  be suspended
  ```

  Such processes might cache stale data and cause subsequent system problems. Uncertified processes must be stopped prior to creating a snapshot file.

- Open Writable Files

  Processes with open writable files are not allowed because the file can change between reboots. If the snapshot procedure encounters an open writable file, the following error message will be returned:

  ```
  %SNAPSHOT-W-WRITEFILE, process <pid> has an open writable file and
  can not be suspended
  ```

  You can use the DCL command SHOW DEVICES/FILES/NOSYSTEM to display the open files. Frequently, the file name (*LOG, SYSUAF.DAT, VMSMAIL_PROFILE.DATA) will indicate which file is writable. Processes with open writable files must be stopped before a snapshot is possible.

  DECwindows workstations are a special case, because certain mail operations and the act of unpausing a session can cause SYSUAF.DAT and VMSMAIL_ PROFILE.DATA to become permanently open. In such a case, the session must be restarted.

  The DECwindows Motif for OpenVMS systems server occasionally opens its log file temporarily to write error and status messages. If a snapshot attempt occurs at this time, the snapshot will fail with the following error in SYS$MANAGER:SNAPSHOT$CLEANUP.LOG:

  ```
  %SNAPSHOT-I-NOTIFY, process 0000004F DECW$SERVER_0 being notified
  %SNAPSHOT-W-WRITEFILE, process 0000004F has an open writable file and can
  not be suspended
  %SNAPSHOT-F-PROCTIMOUT, process synchronization timeout or abort
  ```

  If this error occurs and the DECwindows Motif for OpenVMS systems server has not been quiescent (that is, windows have been appearing or disappearing), then retry the snapshot save operation. If the error still occurs, then issue the following DCL command to determine which files are open by the server and look for any log files that might be present:

  ```
  $ SHOW DEVICES/FILES/NOSYSTEM SYS$SYSDEVICE:
  ```

  If a log file is present, check to be sure that the exec mode symbol, DECW$SM_CREATE_LOG in the DECW$LOGICAL_NAMES table, is defined to be FALSE.

- Unsupported Configurations

  You can use Snapshot only on workstations that are not in a VMScluster configuration.

- Incomplete Notification

  The Snapshot procedure relies on an AST notification mechanism to signal other processes of an impending shutdown and subsequent restart. Some images, such as DECW$TERMINAL_CREATE (activated by the DCL command CREATE/TERMINAL), remain in a higher mode and prevent deliveries of these ASTs. In such cases, the following status notice might appear in the log after a process synchronization timeout:

  ```
  %SNAPSHOT-I-PROCSTATUS, process <pid> is signaling an abort.
  ```

  Such processes must be stopped before the creation of a snapshot file is possible.

- File Size Problems

  The system snapshot software creates a snapshot file automatically. Occasionally, the allocation for this file is incorrect, and the following error is generated:

  ```
  %SNAPSHOT-F-INSUFBLOCKS, insufficient blocks in the snapshot file
  ```

  A related problem is a device allocation failure (RMS Device Full) during the attempt to create the snapshot file.

  In these cases, the snapshot software must guess at the file size because the file system is shut down at the time that the file is written. The more use the system has before the snapshot is taken, the more fragmented memory becomes, which increases the chances that the approximated file size will be wrong.

  One solution is to manually create or extend the snapshot file. The %SNAPSHOT-I-FILESTATS message gives the needed file size, and the following SYSGEN command can perform the file creation:

  ```
  $ RUN SYS$SYSTEM:SYSGEN
  SYSGEN> CREATE SYS$LOADABLE_IMAGES:SNAPSHOT.SYS/SIZE=blocks
  ```

- DECdtm Transactions

  If applications are executing DECdtm transactions when you run Snapshot, Snapshot fails with the following message:

  ```
  Unable to close DECdtm transaction log.  Snapshot attempt aborted.
  ```

  If you see this message, wait 30 seconds, then run Snapshot again.

  If Snapshot fails again with the same message, follow these steps:

  1. Add the following line to the SYS$MANAGER:SYLOGICALS.COM command procedure:

     ```
     $!
     $ DEFINE/SYSTEM/EXECUTIVE SYS$DECDTM_INHIBIT yes
     $!
     ```

  2. Add the following lines to the SYS$MANAGER:SNAPSHOT$SYCLEANUP.COM command procedure:

     ```
     $!
     $ DEASSIGN/SYSTEM/EXECUTIVE SYS$DECDTM_INHIBIT
     $ @SYS$STARTUP:DECDTM$STARTUP.COM
     $!
     ```

  3. Reboot your workstation.

  4. Run Snapshot again.

5. After you have successfully taken a saved image snapshot, remove the line you added in step 1 from the SYS$MANAGER:SYLOGICALS.COM command procedure.

- Disk Space

  The availability of a faster system startup procedure might tempt you to bypass the normal shutdown procedures. In such a case, disk space can eventually disappear and the system can become unable to restart completely.

  Always perform a normal system shutdown or recover disk space by rebuilding the disks periodically with the DCL command SET VOLUME /REBUILD. Note that the solution to this problem does not require the creation of a system snapshot file.

- No Batch Jobs or Queues on Restart

  Because the queue manager has an open writable file, the snapshot procedure must stop all batch jobs and shut down the queue manager. To restart these queue files, edit the SYS$MANAGER:SNAPSHOT$SYCLEANUP.COM procedure and add the queue commands appropriate for your system.

If the snapshot has hung, attempt to force a crash dump. Refer to your system hardware documentation for instructions on forcing a crash dump.

If the dump does not take place, reboot using a conversational boot and retry the snapshot with the following command line:

```
$ @SNAPSHOT DEBUG TRUST_DUMP
```

The TRUST_DUMP parameter indicates that you will not attempt to extend or delete the system dump file and the snapshot image can always trust the original system dump file mapping information in the boot control block. The DEBUG parameter turns off the snapshot watchdog timeout mechanism in case DELTA or XDELTA is in use. Note that TRUST_DUMP has an effect only if the snapshot file is on the system disk.

After the dump file write attempt is complete, you must boot normally by executing a conversational boot and setting BOOT_STYLE to 0, as shown in the following example:

```
>>>B/1 <disk>
SYSBOOT> SET BOOT_STYLE 0
SYSBOOT> CONTINUE
```

Note that the system parameter BOOT_STYLE is automatically set to 1 by the Snapshot command procedure. When BOOT_STYLE is 1, an attempt to boot the system using a system image snapshot is used; when BOOT_STYLE is 0, the system boot is handled in the traditional manner, with no system image snapshot.

Once the system is completely up, you can then analyze the crash dump using the SHOW SNAPSHOT command in the System Dump Analyzer utility (SDA), as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ ANALYZE/CRASH SYSDUMP.DMP
SDA> SHOW SNAPSHOT
```

When a New System Image Snapshot Is Needed

Once you create a successful snapshot image file, you can boot your system using that system image snapshot until any of the following events take place on your system:

- Memory Configuration changes
- Expired licenses
- Open files that have changed
- Changes in disk configuration

In most cases, if you try to reboot your system after one of these changes and the BOOT_STYLE parameter in SYSGEN is set to 1, you will incur an error. Recovery from these errors usually requires a conversational system boot (changing the BOOT_STYLE parameter to 0, and then continuing with a normal boot) followed by the creation of a new snapshot file. ♦

4.8 Shutting Down the System

The operating system provides the following shutdown procedures:

Procedure	Purpose	For More Information
SHUTDOWN.COM	An orderly shutdown procedure. 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.	Section 4.8.1
OPCCRASH.EXE	An emergency shutdown program. Run the OPCCRASH emergency shutdown program if you are unable to perform an orderly shutdown with SHUTDOWN.COM.	Section 4.8.5
Shutdown using console commands	Emergency shutdown commands. Use these console shutdown commands only if OPCCRASH.EXE fails.	Section 4.8.6

4.8.1 Performing an Orderly Shutdown with SHUTDOWN.COM

Use SYS$SYSTEM:SHUTDOWN.COM to shut down the system in an orderly fashion. See Section 4.8.2 for the order of shutdown events.

Do not modify SHUTDOWN.COM. To perform site-specific operations during shutdown, see Section 4.8.3.

Ordinarily, you shut down the system from the SYSTEM account, which includes all privileges by default. To execute SHUTDOWN.COM, you must have either the SETPRV privilege or *all* of the following privileges:

AUDIT
CMKRNL
EXQUOTA
LOG_IO
NETMBX
OPER
SECURITY
SYSNAM

SYSPRV
TMPMBX
WORLD

You can cancel a shutdown without any side effects by pressing Ctrl/Y before SHUTDOWN.COM displays the following message:

```
%SHUTDOWN-I-SITESHUT, The site-specific shutdown procedure will now be invoked.
```

If you press Ctrl/Y after this display, certain system components might have already been shut down, and you will need to recover manually. For example, you might have to manually restart processes, mount disks, or reboot the system.

How to Perform This Task

1. Log in to the system manager's account (SYSTEM), or any privileged account, and enter the following command:

   ```
   $ @SYS$SYSTEM:SHUTDOWN.COM
   ```

   This command invokes the orderly shutdown procedure. The procedure 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.

2. The system displays the following question:

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

   Enter an integer. If you have defined the system logical name SHUTDOWN$MINIMUM_MINUTES, its integer value is the minimum value that you can enter. For example, 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, SHUTDOWN.COM uses the logical name value.

   Caution

   The default is 0 minutes. If you have not defined the logical name SHUTDOWN$MINIMUM_MINUTES, and you do not enter a value, the system will be shut down immediately after you answer the last question.

3. The system displays the following question:

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

   Enter a one-line reason for shutting down the system. For example, ''Monthly preventive maintenance.''

4. The system displays the following question:

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

   Enter YES or NO (Y or N). Note, however, that you cannot spin down the system disk. Also, many disks, particularly SCSI disks, do not spin down in response to this option.

5. The system displays the following question:

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

If you have entered site-specific commands in SYSHUTDWN.COM, press the Return key to accept the default answer, YES. For more information, see Section 4.8.3.2.

6. The system displays the following question:

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

By default, the system does not automatically reboot. However, if you respond YES, the system attempts to reboot automatically when the shutdown is complete. For example, you would specify YES if you are rebooting the system after modifying values for nondynamic system parameters with SYSMAN or SYSGEN. (When you change nondynamic system parameters, you must reboot the system for the new values to take effect.)

7. The system displays a question similar to the following:

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

If you entered YES in step 6, the default answer to this question is ''[shortly via automatic reboot]''.

Press Return to take the default, or enter the expected reboot time in the format you want users to see. 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.

8. The procedure prompts you to specify one or more shutdown options, as follows (if your system is not a member of a VMScluster environment, the procedure lists only the REBOOT_CHECK and SAVE_FEEDBACK options):

```
Shutdown options (enter as a comma-separated list):
 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
 SAVE_FEEDBACK       Save AUTOGEN feedback information from this boot
 DISABLE_AUTOSTART   Disable autostart queues
Shutdown options [NONE]
```

Specify the options you want to use. Choose from the following:

Option	Description
REMOVE_NODE	Causes other nodes in the cluster to decrease the value of the EXPECTED_VOTES system parameter. (This parameter is automatically increased each time a node *joins* the cluster.) Specifying REMOVE_NODE will not decrease the EXPECTED_VOTES below the quorum value. Use this option if the node you are shutting down will be out of the cluster a considerable period of time.
CLUSTER_SHUTDOWN	Synchronizes the shutdown of a cluster; only when the shutdown of each node has progressed to a certain point will the shutdown be completed. Use this option on each node in the cluster to synchronize the shutdown.

Option	Description
REBOOT_CHECK	Verifies the presence of files necessary to reboot the system after shutdown completes. The procedure checks for the necessary files and notifies you if any are missing. Replace missing files before proceeding.
SAVE_FEEDBACK	Records feedback data collected from the system since it was last booted and creates a new version of the AUTOGEN feedback data file, which AUTOGEN can use the next time it runs. For detailed information on using the AUTOGEN feedback mechanism, see Section 14.4.1.
DISABLE_AUTOSTART	Specifies the time interval between the DISABLE AUTOSTART/QUEUES command and system shutdown. For more information, see Section 13.9.1.13.

Example

```
$ @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]? Return
Do you want to invoke the site-specific shutdown procedure [Yes]? Return
Should an automatic system reboot be performed [No]? Return
When will the system be rebooted [later]? 12:30
Shutdown options (enter as a comma-separated list):
 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
 SAVE_FEEDBACK       Save AUTOGEN feedback information from this boot
 DISABLE_AUTOSTART   Disable autostart queues
Shutdown options [NONE] Return

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-MAY-1995 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 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 3 minutes; back up 12:30. Please log off node AVALON.
MONTHLY PREVENTIVE MAINTENANCE
```

```
%%%%%%%%%%  OPCOM, 16-MAY-1995 12:07:12.30  %%%%%%%%%%
Message from user DECnet on AVALON
DECnet event 2.0, local node state change
From node 2.161 (AVALON), 16-MAY-1995 12:07:22.26
Operator command, Old state = On, New state = Shut
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-MAY-1995 12:08:12:30  %%%%%%%%%%
Message from user JOB_CONTROL on AVALON
-SYSTEM-S-NORMAL, normal successful completion

%%%%%%%%%%  OPCOM, 16-MAY-1995 12:08:42.30  %%%%%%%%%%
Message from user DECNET on AVALON
DECnet shutting down

%SYSTEM-I-STOPQUEUES, The queues on this node will now be stopped.
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

SHUTDOWN message on AVALON, from user SYSTEM at _AVALON$OPA0:   12:10:00.20
AVALON will shut down in 0 minutes; 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-MAY-1995 12:09:42.30  %%%%%%%%%%
Message from user System on AVALON
_AVALON$OPA0:, AVALON shutdown was requested by the operator.

%%%%%%%%%%  OPCOM, 16-MAY-1995 12:10:02.44  %%%%%%%%%%
Logfile was closed by operator _AVALON$OPA0:
Logfile was SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;8

%%%%%%%%%%  OPCOM, 16-MAY-1995 12:10:32.20  %%%%%%%%%%
Operator _AVALON$OPA0: has been disabled, username SYSTEM

        SYSTEM SHUTDOWN COMPLETE
```

VAX

On VAX systems, the following message is also displayed:

```
USE CONSOLE TO HALT SYSTEM
```

Halt the system after you see this message. ♦

4.8.2 Understanding the Order of Shutdown Events

The following events occur as the shutdown proceeds. The procedure displays the corresponding messages on the terminal.

1. At decreasing time intervals, SHUTDOWN.COM broadcasts, to all users on the system, a message requesting users to log out.
2. SHUTDOWN.COM defines the system logical name SHUTDOWN$TIME to be the absolute time of shutdown. For example, if you execute SHUTDOWN.COM, and at 12:00 you specify the value 10 in response to the first question, SHUTDOWN defines the logical name to be 12:10 on that day. To see if a shutdown is in progress or to determine the actual time of shutdown, you can enter the command SHOW LOGICAL SHUTDOWN$TIME. This feature is useful if you miss a shutdown broadcast message.

3. At 6 minutes or less before system shutdown, the terminal from which you invoked SHUTDOWN becomes an operator's console. SHUTDOWN disables all future nonoperator logins and shuts down the DECnet network if it is running. At this point, users logged in to the system with the SET HOST command lose their sessions.
4. One minute before shutdown, SHUTDOWN.COM stops batch and output execution queues and stops the queue manager.
5. At the absolute time of shutdown, SHUTDOWN.COM invokes the site-specific shutdown command procedure SYS$MANAGER:SYSHUTDWN.COM, if you requested it.
6. SHUTDOWN.COM stops all remaining user processes; however, system processes continue. Ancillary control processes (ACPs) might delete themselves when their mounted volumes are finally dismounted.
7. On multiprocessor systems, SHUTDOWN.COM stops the secondary processors.
8. SHUTDOWN.COM removes all installed images.
9. SHUTDOWN.COM dismounts all mounted volumes and, if you requested it, spins down the disks. If you defined SHUTDOWN$VERBOSE, the procedure lists each disk as it is dismounted.

   The procedure does not spin down the system disk, nor does it dismount or spin down the quorum disk (if one exists on your system).
10. SHUTDOWN.COM closes the operator log file.
11. SHUTDOWN.COM runs the program SYS$SYSTEM:OPCCRASH.EXE to shut down the system.
12. If you requested an automatic reboot, the system reboots, provided you set the necessary controls. You requested an automatic reboot if you answered YES to the following question:

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

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

    ```
    SYSTEM SHUTDOWN COMPLETE
    ```

    VAX

    On VAX systems, the following message is also displayed:

    ```
    USE CONSOLE TO HALT SYSTEM
    ```

    Halt the system after you see this message. ♦

4.8.3 Customizing SHUTDOWN.COM to Perform Site-Specific Operations

In addition to choosing shutdown options when you execute SHUTDOWN.COM, you can customize SHUTDOWN.COM to meet the needs of your site in the following ways.

Method	For More Information
Defining logical names	Section 4.8.3.1
Modifying the site-specific shutdown command procedure	Section 4.8.3.2

4.8.3.1 Defining Logical Names

Before executing SHUTDOWN.COM, you can define the following logical names to control the operations of the command procedure:

Logical Name	Description
SHUTDOWN$DECNET_MINUTES	Defines the number of minutes remaining before DECnet is shut down; must be defined with the /SYSTEM qualifier. The default is 6 minutes.
SHUTDOWN$DISABLE_AUTOSTART	Specifies the number of minutes between the time autostart is disabled for queues and the time the system is shut down. For more information, see Section 13.9.1.13.
SHUTDOWN$INFORM_NODES	Specifies a list of VMScluster nodes to be notified when the system is shutting down. This logical name is described in detail in this section.
SHUTDOWN$MINIMUM_MINUTES	Defines the minimum number of minutes you can specify as number of minutes to shutdown. For example, if your users require 30 minutes' notice before a system shutdown, define this logical name to be 30.
SHUTDOWN$QUEUE_MINUTES	Defines the number of minutes remaining before shutdown when queues are shut down; must be defined with the /SYSTEM qualifier. The default is 1 minute.
SHUTDOWN$TIME	Defines the absolute time of shutdown.
SHUTDOWN$VERBOSE	If defined to any string, specifies that the shutdown command procedure is to list each disk as it is dismounted.

If you plan to use an option every time you use SHUTDOWN.COM, define the logical name in the site-specific startup command procedure SYLOGICALS.COM. For more information, see Section 5.2.5.

Specifying a List of Nodes to Be Notified When the System Is Shutting Down

You can define the logical name SHUTDOWN$INFORM_NODES to be a list of VMScluster nodes that are notified when the system is shut down. You must define SHUTDOWN$INFORM_NODES before executing SYS$SYSTEM:SHUTDOWN.COM.

To define SHUTDOWN$INFORM_NODES, enter a command in the following format:

DEFINE SHUTDOWN$INFORM_NODES "node-list"

For *node-list*, specify the list of nodes to be informed. For example:

```
$ DEFINE SHUTDOWN$INFORM_NODES "NODE1,NODE2,NODE3"
```

If you plan to inform the same nodes every time you shut down the system, add the command to the site-specific startup command procedure SYLOGICALS.COM. For more information, see Section 5.2.5.

If you define SHUTDOWN$INFORM_NODES, all member nodes included in the list are notified when you execute SHUTDOWN.COM. Users on the node that is being shut down are always notified, regardless of whether you define SHUTDOWN$INFORM_NODES. If you omit the name of the node that is being

shut down from the list specified in the DEFINE command, SHUTDOWN.COM automatically adds the name to the list.

The information in Table 4–3 indicates which nodes are notified at different phases of the shutdown sequence, depending on whether SHUTDOWN$INFORM_NODES is defined.

Table 4–3 Node Notification During Shutdown

Shutdown Phase	If SHUTDOWN$INFORM_NODES Is Not Defined	If SHUTDOWN$INFORM_NODES Is Defined
First shutdown notification	Notify all terminals on all nodes	Notify all terminals on all listed nodes
Between first shutdown notification and 2 minutes before final shutdown	Notify all terminals logged in to the node that is shutting down	Notify all users logged in on all listed nodes
Between 2 minutes before final shutdown notification until final shutdown	Notify all users logged in on all nodes	Notify all users logged in on all listed nodes
Shutdown canceled	Notify all terminals on all nodes	Notify all terminals on all listed nodes

4.8.3.2 Modifying the Site-Specific Shutdown Command Procedure

You can add site-specific commands to the site-specific shutdown procedure SYS$MANAGER:SYSHUTDWN.COM. An empty SYSHUTDWN.COM file is included in your distribution kit.

SHUTDOWN.COM prompts you to indicate if you want to execute the site-specific procedure SYSHUTDWN.COM:

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

To invoke the site-specific procedure, press the Return key to accept the default answer YES.

4.8.4 Performing an Orderly Shutdown with the System Management Utility (SYSMAN)

The advantage of using SYSMAN for shutdown is that you can shut down a *group* of nodes quickly. SYSMAN enables you to enter all of the shutdown parameters in one command line, rather than responding to the interactive dialog in SHUTDOWN.COM. SYSMAN does not wait for the nodes to shut down before you can use other SYSMAN commands; the interface returns immediately.

How to Perform This Task

1. Enter the following command at the $ prompt:

   ```
   $ RUN SYS$SYSTEM:SYSMAN
   ```

2. At the SYSMAN> prompt, enter the following command:

   ```
   SYSMAN> SHUTDOWN NODE/[qualifier]
   ```

Qualifiers can be any of the following options:

Qualifier	Function
MINUTES_TO_SHUTDOWN	Indicates the number of minutes until shutdown occurs.
REASON	Indicates the reason for the shutdown.
REBOOT_TIME	Indicates the time you expect to reboot the system, such as LATER, 2 P.M., or 14:00. This time is displayed in the shutdown message to users.
[NO]SPIN_DOWN_DISKS	Spins down disks. The default is NO. You cannot spin down the system disk.
[NO]INVOKE_SYSHUTDOWN	Invokes the site-specific shutdown procedure. The default is INVOKE_SYSHUTDOWN.
[NO]AUTOMATIC_REBOOT	Reboots the system automatically when the shutdown is complete. The default is NO.
[NO]REBOOT_CHECK	Checks for basic operating system files and notifies you if any are missing. The default is NO.
[NO]CLUSTER_SHUTDOWN	Shuts down the entire VMScluster. The default is NO.
[NO]REMOVE_NODE	Removes the node from the active cluster quorum; use this when you do not expect the shut-down node to rejoin the cluster for an extended period. The default is NO.
[NO]SAVE_FEEDBACK	Records feedback data from the system since it was last booted and creates a new version of the AUTOGEN feedback data file, which you can use the next time you run AUTOGEN. The default is NO.

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SHUTDOWN NODE/MINUTES_TO_SHUTDOWN=10/REBOOT_TIME="later" -
_SYSMAN> /REASON="DISK CORRUPTION PROBLEMS"/REBOOT_CHECK/SAVE_FEEDBACK
```

If you enter this command example on NODE21, it requests a shutdown on NODE21 with:

- A message to users on all the cluster nodes, specifying:

  ```
  SHUTDOWN message on node NODE21, from user SYSTEM at _NODE21$0PA0:
  12:00:00:20. NODE21 will shut down in 10 minutes; back up later.
  Please log off node NODE21. DISK CORRUPTION PROBLEMS
  ```

- A check for any missing operating system files and notification any are missing
- Creation of a new AUTOGEN feedback data file based on the feedback data collected since the system was last booted

For more information, see the *OpenVMS System Management Utilities Reference Manual*.

4.8.5 Performing an Emergency Shutdown with the OPCCRASH.EXE Program

Ordinarily, you shut down the system using the orderly shutdown procedure SHUTDOWN.COM. After SHUTDOWN.COM performs orderly housekeeping tasks, it invokes the program SYS$SYSTEM:OPCCRASH.EXE to shut down the system. OPCCRASH.EXE performs only the following minimal housekeeping functions:

- Writes the modified page list back to disk. This ensures that all writable section files are updated to their correct state before the system crashes and all in-memory data is lost.
- Unless the logical name OPC$NODUMP is defined, creates a crash dump by writing physical memory to the system dump file. For more information on the system dump file, see Section 15.1.

In an emergency, if you cannot invoke SHUTDOWN.COM, you can run the OPCCRASH.EXE program to shut down your system immediately without performing any of the housekeeping functions that ensure an orderly shutdown.

Note

Run the OPCCRASH.EXE program directly only if SHUTDOWN.COM fails.

How to Perform This Task

To run the OPCCRASH.EXE program directly, you must have the CMKRNL privilege. You can enter the commands from any terminal and any privileged account. Follow these steps:

1. Log in to any privileged account.
2. Enter the following command:

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

3. 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 one of the following manuals:
   - On VAX systems, see the upgrade and installation supplement for your VAX computer.
   - On Alpha systems, see the *Upgrade and Installation Manual*.
4. A message similar to the following is displayed at the console:

   ```
   SYSTEM SHUTDOWN COMPLETE
   ```

VAX

On VAX systems, the following message is also displayed:

```
USE CONSOLE TO HALT SYSTEM
```

Halt the system when you see this message. ♦

Example

The following example runs the OPCCRASH program to force a system crash, and halts the system:

```
$ RUN SYS$SYSTEM:OPCCRASH

         SYSTEM SHUTDOWN COMPLETE

Ctrl/P
>>>HALT

     HALTED AT 8000708A
```

VAX

On VAX systems, the following message is also displayed:

```
USE CONSOLE TO HALT SYSTEM
```

Halt the system when you see this message. ♦

4.8.6 Performing an Emergency Shutdown Using Console Commands

Certain computer consoles have an additional emergency CRASH command. If your computer has the CRASH command, it is located on the console media; you can execute it only from the console prompt on the console terminal. For example:

```
P00>>> CRASH
```

If the CRASH command does not exist on your console, you can shut down the system manually from the console.

Note

Use CRASH commands from the console only if the OPCCRASH.EXE program fails.

VAX

On VAX systems, enter the following commands:

```
P00>>> D PSL 041F0000
P00>>> D PC FFFFFFFF
P00>>> CON ♦
```

Alpha

On Alpha systems, enter the following commands:

```
P00>>> D PS 1F00
P00>>> D PC FFFFFFFF00000000
P00>>> CON ♦
```

See one of the following manuals for a description of the CRASH command or for equivalent commands to use to force an abrupt emergency shutdown:

- On VAX systems, see the upgrade and installation supplement for your VAX computer.
- On Alpha systems, see the *Upgrade and Installation Manual*.

5
Customizing the Operating System

After you install the operating system, you can customize it for site-specific requirements.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Adding and deleting optional files	Section 5.1
Modifying site-specific startup command procedures	Section 5.2
Modifying login command procedures	Section 5.3
Customizing startup databases	Section 5.4
†Registering images that have system version dependencies	Section 5.5
Customizing the Help Message database	Section 5.6
Customizing MAIL	Section 5.7
Setting correct time zone information on your system	Section 5.8
‡ Setting time using the Battery_Backed Watch (BBW)	Section 5.9
Choosing languages, and date and time formats	Section 5.10
Saving your customization	Section 5.11

†VAX specific
‡Alpha specific

This chapter explains the following concepts:

Concept	Section
Site-specific startup command procedures	Section 5.2.1
The order of startup events	Section 5.2.2
Startup databases	Section 5.4.1
The layered product startup database	Section 5.4.2

5.1 Adding and Deleting Optional Files

OpenVMS lets you customize the size of the operating system by deleting or adding optional system files, including support for DECwindows. This is particularly valuable for small systems or systems with limited disk space.

For example, if your system is a MicroVAX II computer with an RD54 system disk and you will not use system programming features such as the Delta/XDelta Debugger (DELTA/XDELTA) or the System Dump Analyzer utility (SDA), you might remove these files from the system disk.

Depending on the system you are using, you can add or delete files in one of the following ways:

VAX

- On VAX systems, you can use the OpenVMS Tailoring utilities, VMSTAILOR and DECW$TAILOR:
  - VMSTAILOR—applies to optional system files
  - DECW$TAILOR—applies to support for DECwindows

  You delete files from and add files to the system disk by identifying classes and subclasses of operating system files that you want to add or delete. You might delete or add an entire class or selected subclasses of files within a class.

  If you delete files, you can add them again at any time using VMSTAILOR or DECW$TAILOR and your operating system distribution media.

  See the *Upgrade and Installation Manual* for more information about VMSTAILOR and DECW$TAILOR.♦

Alpha

- On Alpha systems, you can use the POLYCENTER Software Installation utility to add or delete optional system files. You add or delete files by selecting or deselecting options and suboptions.

  To invoke the POLYCENTER Software Installation utility for this purpose, you can use either of the following methods:
  - Use the DCL command PRODUCT RECONFIGURE.
  - Use the DECwindows Motif interface from the POLYCENTER Software Installation utility and select the Reconfigure option on the Mode menu.

  For more information on using the POLYCENTER Software Installation utility, see Section 3.7.♦

5.2 Modifying Site-Specific Startup Command Procedures

An important part of customizing your system is to create or modify site-specific startup command procedures. Adding commands to these procedures ensures that the commands are executed each time the system reboots.

5.2.1 Understanding Site-Specific Startup Command Procedures

You should understand the following terms:

Term	Definition
Startup command procedure	A command procedure that executes when the system starts up.
Site-independent startup command procedure	A startup command procedure that is required for and provided with all OpenVMS systems, regardless of site-specific requirements. This procedure is named SYS$SYSTEM:STARTUP.COM. *Do not modify this procedure*. When your system boots, it automatically executes STARTUP.COM. For more information, see Section 4.1.3.

Term	Definition
Site-specific startup command procedures	Startup command procedures that you can modify to perform operations specific to your site. Use any text editor to add or modify commands in these procedures. STARTUP.COM executes several site-specific startup command procedures that Digital provides. These procedures are listed in Table 5–1. You can also create your own procedures and execute them from SYSTARTUP_VMS.COM.

Table 5–1 lists and describes the site-specific startup command procedures provided by Digital, in the order in which they execute. These procedures are located in the system directory with the logical name SYS$STARTUP.

Table 5–1 Site-Specific Startup Command Procedures

Order	Command Procedure	Function
1	SYCONFIG.COM	A file to which you add commands for site-specific device configuration. For more information, see Section 5.2.4.
2	SYLOGICALS.COM	A file to which you add commands to define your site-specific system logical names. For more information, see Section 5.2.5.
3	SYPAGSWPFILES.COM	A file to which you add commands to install page and swap files (other than the primary page and swap files in SYS$SYSTEM, which are installed automatically). For more information, see Section 5.2.3.
4	SYSECURITY.COM	A file to which you add commands to define the location of security auditing and security archive files before starting the security auditing server. For more information, see Section 5.2.6.
5	SYSTARTUP_VMS.COM	A general-purpose command procedure to which you add commands to perform miscellaneous operations for setting up your site. For example, you might mount public disks in SYSTARTUP_VMS.COM. For more information, see Section 5.2.7.

Template Files

Your distribution kit provides two versions of each site-specific command procedure in the directory SYS$MANAGER:

- An executable version with the file type .COM (for example, SYS$MANAGER:SYCONFIG.COM). The system executes files with the file type .COM; you can edit .COM files (except for STARTUP.COM) to meet your site-specific needs.
- A backup version with the file type .TEMPLATE (for example, SYS$MANAGER:SYCONFIG.TEMPLATE).

Note

Do not modify or delete the Digital-supplied template command files with the .TEMPLATE file type. The VMSKITBLD.COM procedure uses these files to create a new system disk. If you must use the .TEMPLATE

version of the file because your .COM version is damaged, copy the .TEMPLATE file to a file with the .COM file type, and edit the copy.

Rules for Modifying Startup Command Procedures

When modifying site-specific startup command procedures, be sure to follow these rules:

- Conform to the rules of command procedures, as described in the *OpenVMS User's Manual*.
- Keep the files in the SYS$MANAGER directory.
- Keep the file names given to the command procedures.
- Modify only the executable version of the files (with the file type .COM), not the template version (with the file type .TEMPLATE).
- Do not modify the *site-independent* startup command procedure STARTUP.COM.
- Before modifying command procedures, understand the order of startup events. For information, see Section 5.2.2.

Caution

The startup procedures provided by Digital should always work. However, if you introduce an error in the startup or login procedures, it is possible to accidentally lock yourself out of the system. Section 4.3.2 describes a boot procedure you can use in such an emergency.

5.2.2 Understanding the Order of Startup Events

Before modifying the site-specific startup command procedures, you should understand the order of system startup events.

A database file named VMS$PHASES.DAT determines the order of the phases of the startup procedure. It is a sequential list of the phases that STARTUP.COM starts. It includes a series of four basic phases (INITIAL, CONFIGURE, DEVICE, and BASEENVIRON) that start the operating system, followed by a series of phases for layered products.

Caution

Do not modify VMS$PHASES.DAT. To start up correctly, the system requires that the contents of this file remain intact.

On starting up, a system performs tasks in the following order:

1. Defines logical names needed for basic operations, and installs images listed in SYS$MANAGER:VMSIMAGES.DAT.
2. Executes SYCONFIG.COM.
3. Adds any new drivers by executing one of the following commands:

VAX

- On VAX systems, the SYSGEN command AUTOCONFIGURE ALL. This command automatically configures the device driver database, locates all standard devices attached to the system, and loads and connects their device drivers. ♦

Alpha

- On Alpha systems, the SYSMAN command IO AUTOCONFIGURE. This command automatically configures the device driver database, locates all standard devices attached to the system, and loads and connects their device drivers. ♦

If the symbol STARTUP$AUTOCONFIGURE_ALL is defined as 0 or FALSE by SYS$MANAGER:SYCONFIG.COM, this step is not performed.

4. Installs the primary swap file, if the file is present.
5. Starts the CONFIGURE process (swappable). If the system parameter NOAUTOCONFIG is set to 1, the CONFIGURE process is not started. If the symbol STARTUP$AUTOCONFIGURE_ALL is defined as 0 or FALSE by SYS$MANAGER:SYCONFIG.COM, this step is not performed.
6. Executes SYLOGICALS.COM. At this point, all devices have been made available through the AUTOCONFIGURE ALL command (step 3) or will be made available by the CONFIGURE process (started in step 5).
7. If the system is a satellite node in a VAXcluster or VMScluster environment, executes SATELLITE_PAGE.COM to install page and swap files on a local disk. SATELLITE_PAGE.COM is created when you execute the CLUSTER_CONFIG.COM procedure.
8. Executes SYPAGSWPFILES.COM.
9. Performs the following steps in no specified order:
   - Installs required images
   - Starts various operating system processes (OPCOM, CACHE_SERVER, CLUSTER_SERVER, ERRFMT, JOBCTL)
   - Executes SYSECURITY.COM and starts the AUDIT_SERVER process

   VAX

   - On VAX systems, starts the security server, SECURITY_SERVER, which manages the proxy database and intrusion database ♦
   - Starts the License Management Facility (LMF) and loads all appropriate Product Authorization Keys (PAKs) from the LMF database
10. Performs the following steps in no specified order:
    - Enables operator consoles and the operator log files
    - Starts the SMISERVER process

Note

The order of events within system startup might change in future releases of the operating system.

5.2.3 Modifying SYPAGSWPFILES.COM to Install Page and Swap Files

When the system boots, it automatically installs the primary page and swap files, if they are present in the SYS$SYSTEM directory. If the page and swap files are not in SYS$SYSTEM, or if secondary page and swap files are located on a disk other than the system disk, you must install these files each time the system boots. To install these files, add commands to SYPAGSWPFILES.COM.

Before performing this task, you should understand page and swap files and why you might want to move them. For more information, see Section 15.2.

The SYPAGSWPFILES.COM file can also include commands other than INSTALL commands, such as SYSGEN CREATE commands and the DCL commands INITIALIZE and MOUNT, to set up the page and swap files. Note that, at the time STARTUP.COM invokes SYPAGSWPFILES.COM, only the system disk is mounted. Therefore, you might need to add MOUNT commands to SYPAGSWPFILES.COM to mount the disks that hold the page and swap files.

The system must have installed at least one page file before SYPAGSWPFILES.COM exits. Otherwise, STARTUP.COM displays the following error message:

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

Caution

If a system dump file with the name SYSDUMP.DMP does not exist in the SYS$SPECIFIC:[SYSEXE] directory, the primary page file PAGEFILE.SYS must exist in SYS$SPECIFIC:[SYSEXE]PAGEFILE.SYS for writing crash dumps. If neither SYSDUMP.DMP nor PAGEFILE.SYS is located in SYS$SPECIFIC:[SYSEXE], no crash dump file is produced.

You can also use SATELLITE_PAGE.COM to install page and swap files on a satellite node's local disk. SATELLITE_PAGE.COM is created when you run CLUSTER_CONFIG.COM. For more information on installing page and swap files on a satellite node's local disk, see *VMScluster Systems for OpenVMS*.

How to Perform This Task

1. Enter SYSGEN CREATE commands in the following format to create secondary system files in the desired locations:

   CREATE file-spec/SIZE=block-count

   For example:

   ```
   SYSGEN> CREATE DUA2:[PAGE_SWAP]PAGEFILE_1.SYS/SIZE=100000
   SYSGEN> CREATE DUA2:[PAGE_SWAP]SWAPFILE_1.SYS/SIZE=100000
   ```

   The SYSGEN CREATE command creates or extends files that can be used as a page, swap, or dump file. You create these files only once.

   For more information on creating page and swap files, see Section 15.11. For more information on the SYSGEN command CREATE, see the *OpenVMS System Management Utilities Reference Manual* (SYSGEN).

2. Invoke any editor to edit SYS$MANAGER:SYPAGSWPFILES.COM.

3. If necessary, add a MOUNT command to mount the disk or disks that are to hold the secondary page and swap files. Disks other than the system disk are not yet mounted at the time SYPAGSWPFILES.COM is invoked. For information on the MOUNT command, see the *OpenVMS System Management Utilities Reference Manual* (MOUNT).
4. Add the following command to make it easier to invoke SYSGEN:

   ```
   $ SYSGEN := $SYSGEN
   ```

5. Add commands in the following format to SYPAGSWPFILES.COM to install the secondary files each time the system boots.

   For page files, use the following format:

   SYSGEN INSTALL file-spec /PAGEFILE

   For swap files, use the following format:

   SYSGEN INSTALL file-spec /SWAPFILE

   The INSTALL command activates secondary page and swap files. Page and swap files not located in SYS$SYSTEM must be installed each time the system boots.

Example

The following commands in SYPAGSWPFILES.COM install secondary page and swap files on the device DUA10: with the logical name PAGE_SWAP:

```
$ MOUNT/SYSTEM/NOASSIST DUA10: SYS2 PAGE_SWAP
$ SYSGEN := $SYSGEN
$ SYSGEN INSTALL PAGE_SWAP:[SYSTEM]PAGEFILE1.SYS/PAGEFILE
$ if $status then write sys$output "Installed page file PAGEFILE1.SYS"
$ SYSGEN INSTALL PAGE_SWAP:[SYSTEM]SWAPFILE1.SYS/SWAPFILE
$ if $status then write sys$output "Installed swap file swapfile1.sys"
```

5.2.4 Modifying SYCONFIG.COM to Configure Devices

You can add commands to SYCONFIG.COM to perform site-specific device configuration, including connecting nonstandard devices and suppressing autoconfiguration.

5.2.4.1 Connecting Nonstandard Devices

Standard devices are automatically connected and configured by STARTUP.COM each time the system boots. Nonstandard devices (devices not supplied by Digital) are not automatically connected and configured; you must connect and configure these devices manually by entering certain commands. To execute these commands each time the system starts up, add the commands to SYCONFIG.COM.

VAX

On VAX systems, add SYSGEN CONNECT commands. For more information on connecting devices, see Section 7.4. For more information on the SYSGEN CONNECT command, see the *OpenVMS System Management Utilities Reference Manual* (SYSGEN). ♦

Alpha

On Alpha systems, add SYSMAN IO CONNECT commands. For more information on connecting devices, see Section 7.4. For more information on the SYSMAN IO CONNECT command, see the *OpenVMS System Management Utilities Reference Manual* (SYSMAN). ♦

Example

To connect a nonstandard device called the QQ device, add the following commands to SYCONFIG.COM:

```
$ SYSGEN := $SYSGEN
$ SYSGEN CONNECT QQA0
```

5.2.4.2 Suppressing Autoconfiguration of Devices

You might want to suppress autoconfiguration for various reasons, including the following:

- To customize the order in which you configure devices
- To troubleshoot booting problems
- To allow Small Computer System Interface (SCSI) based workstations to use devices on another workstation's SCSI bus without causing conflicts during boot time

You can define a symbol in SYCONFIG.COM to suppress autoconfiguration. For more information, see Section 7.4.3.

5.2.5 Modifying SYLOGICALS.COM to Define Systemwide Logical Names

To define systemwide logical names, add commands to the command procedure SYS$MANAGER:SYLOGICALS.COM.

A systemwide logical name is a logical name that applies to the entire system. It is defined in the system **logical name table** and can be used by any process in a system.

As supplied by Digital, SYLOGICALS.COM contains commands that assign systemwide logical names on a MicroVAX system that is not in a VMScluster environment. If your system is not a standalone MicroVAX system, you can ignore the procedure at the beginning of the template file and add systemwide logical name assignments to the end of the file.

You can add commands to create your own site-specific systemwide logical names. In addition, if you want to change default definitions for the following system logical names, you can include the definitions in SYLOGICALS.COM. Table 5–2 lists some commonly defined logical names.

Table 5–2 Commonly Defined System Logical Names

Logical Name	For More Information
LMF$LICENSE	*OpenVMS License Management Utility Manual*
NETNODE_REMOTE	*DECnet for OpenVMS Networking Manual*
NETPROXY	*OpenVMS Guide to System Security*
†NET$PROXY	*OpenVMS Guide to System Security*
QMAN$MASTER	Section 12.3
RIGHTSLIST	*OpenVMS Guide to System Security*
SYS$ERRORLOG	Section 18.4.1
SYS$MONITOR	Section 18.8.9

†VAX specific

(continued on next page)

Table 5–2 (Cont.) Commonly Defined System Logical Names

Logical Name	For More Information
SYSUAF	*OpenVMS Guide to System Security*
VMSMAIL_PROFILE	*OpenVMS User's Manual*

Digital recommends that you define logical names for system components (for example, public disks and directories) in executive mode, using the /EXECUTIVE_MODE qualifier with the ASSIGN or DEFINE command. This ensures that the logical name is available during system operations such as the activation of privileged mode images (LOGINOUT, MAIL, and so forth).

For detailed information on logical name assignments and the privilege modes (executive, kernel, supervisor, and user), see the *OpenVMS User's Manual*.

How to Perform This Task

1. Invoke any editor to edit the file SYS$MANAGER:SYLOGICALS.COM.
2. Add logical name definitions in the following format to the end of the file, immediately preceding the EXIT command:

   DEFINE/SYSTEM/EXECUTIVE/NOLOG logical-name equivalence-name

   For example:

   ```
   DEFINE/SYSTEM/EXECUTIVE/NOLOG FINANCE_DISK DRAC$DRA2:
   ```

   For more information about the DEFINE command, see the *OpenVMS DCL Dictionary*.
3. Exit the editor to create a new version of the file. The highest version will automatically be invoked by STARTUP.COM each time the system boots.

Example

```
$ DEFINE/SYSTEM/EXECUTIVE/NOLOG FINANCE_DISK DRAC$DRA2:
$ DEFINE/SYSTEM/EXECUTIVE/NOLOG SYSDSK SYS$SYSDEVICE:
```

In this example, any user on the system (and any program running on the system) could use the name FINANCE_DISK (the logical name) in place of DRAC$DRA2: (the physical device name). Similarly, you can refer to the system disk (SYS$SYSDEVICE:) as SYSDSK.

5.2.6 Modifying SYSECURITY.COM to Set Up Security Auditing

SYSECURITY.COM runs prior to starting the security audit server process. You can add commands to this file to mount or define any disks that you want to hold security auditing log files or local security archive files. For more information on security auditing, see Section 18.7.

Ordinarily, the system turns on auditing in VMS$LPBEGIN, just before SYSTARTUP_VMS.COM executes. However, you can change this behavior by redefining the logical name SYS$AUDIT_SERVER_INHIBIT.

To inhibit the automatic startup of auditing, edit the SYS$MANAGER:SYLOGICALS.COM command procedure to add the following line:

```
$ DEFINE/SYSTEM/EXECUTIVE SYS$AUDIT_SERVER_INHIBIT YES
```

Then you can initiate auditing during another phase of system startup, perhaps at the end of SYSTARTUP_VMS.COM, by editing the command file to add the following line:

```
$ SET AUDIT/SERVER=INITIATE
```

For information on editing SYSTARTUP_VMS.COM, see Section 5.2.7.

5.2.7 Modifying SYSTARTUP_VMS.COM to Perform General Operations

To perform any site-specific command not performed by another startup command procedure, you can add or modify commands in the general-purpose, site-specific startup command procedure, SYSTARTUP_VMS.COM.

Digital recommends that you edit this procedure to modify or add commands that perform tasks such as the following:

Task	For More Information
Mounting public disks	Section 5.2.7.1
Setting the characteristics of terminals and printer devices	Section 5.2.7.3
Starting queues and enabling autostart for queues	Section 5.2.7.4
Installing known images	Section 5.2.7.5
‡Installing resident images	Section 5.2.7.6
Setting up the OpenVMS InfoServer Client software	Section 5.2.7.7
Running the System Dump Analyzer	Section 5.2.7.8
Purging the operator's log file	Section 5.2.7.9
Submitting batch jobs that are run at system startup time	Section 5.2.7.10
Creating systemwide announcements	Section 5.2.7.11
Starting up the LAT protocol software	Section 5.2.7.12
Starting the DECnet for OpenVMS network	Section 5.2.7.13
Starting up the DIBOL Message Manager	Section 5.2.7.14
Defining the number of interactive users	Section 5.2.7.15

‡Alpha specific

How to Perform This Task

To modify SYSTARTUP_VMS.COM, perform the following steps:

1. Invoke any editor to edit the file.
2. To prevent the command procedure from exiting if it invokes an error, include the DCL command SET NOON at the beginning of the file. This command disables error checking after the execution of each command in the procedure. For more information on error checking, see the *OpenVMS User's Manual*.
3. Add commands to perform site-specific operations. Sections 5.2.7.1 to 5.2.7.15 describe operations that are typically performed by this command procedure.
4. Exit the editor to create a new version of the file. The highest version will automatically be invoked by STARTUP.COM each time the system boots.

5.2.7.1 Mounting Public Disks

A **public volume** is a disk that any process on the system can access. To make disks available for public use, you must do the following:

- Physically load and spin up the disk.
- If the disk is new, initialize it.
- Mount the disk for systemwide access using the DCL command MOUNT. (Using the MOUNT command for the system disk is not necessary— the system disk is already mounted when the system starts.)

For more information on public volumes, see Section 8.1.3 and Section 8.5. For more information on the MOUNT command, see the *OpenVMS DCL Dictionary*.

How to Perform This Task

Add MOUNT commands in the following format to the command procedure:

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

where:

- *ddcu* is the physical device name (including a colon immediately after the device name). For information on physical device names, see Section 7.1.
- *volume-label* is an alphanumeric identification that you assign with the INITIALIZE command.
- *logical_name* is the logical name that you want to assign to the device. 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 /SYSTEM qualifier makes the disk available for systemwide access.

Note that, by default, the system creates the following logical name when you use the MOUNT command:

```
DISK$volume_label
```

In many cases, using the default logical name will meet your needs.

When mounting disks in a startup command procedure, do not specify the /CLUSTER qualifier, even in a VAXcluster or VMScluster environment. Each node executes its own startup command procedure, so each node mounts disks for itself.

Note

When SYSTARTUP_VMS.COM executes (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_VMS.COM, specify /ASSIST with each MOUNT command.

For a DSA disk, you must also insert a WAIT statement in the command procedure prior to the first MOUNT statement. The wait time is controller-dependent. If you omit a WAIT statement, the MOUNT request might fail with a ‘‘no such device’’ status. See the *OpenVMS I/O User’s Reference Manual* for more information.

For more information on public volumes, see Section 8.1.3 and Section 8.5. For more information on the MOUNT command, see the *OpenVMS DCL Dictionary*.

5.2.7.2 Mounting Disks That Must Be Available Early in Startup

If you have any disks that must be mounted early in startup, you can add MOUNT commands to SYCONFIG.COM. For example, your site might require that certain files be available before SYSTARTUP_VMS.COM executes. For more information about SYCONFIG.COM, see Section 5.2.4.

5.2.7.3 Setting Terminal and Printer Characteristics

To establish the device characteristics of the terminals and printers on the system, use a series of SET commands in your startup command procedure. For more information on the commands you use to set up devices, see Section 7.5.1 and Section 7.6.1.

If your configuration is simple, you can add the commands to SYSTARTUP_VMS.COM. If your configuration requires a large number of commands, create a separate command procedure (for example, DEVICE_SETUP.COM) and execute it from SYSTARTUP_VMS.COM. When the device setup command procedure finishes executing, control returns to SYSTARTUP_VMS.COM.

5.2.7.4 Starting Queues and Enabling Autostart for Queues

You should add commands to SYSTARTUP_VMS.COM to perform the following tasks:

- Enable autostart for queues
- Start nonautostart execution queues

If your configuration is simple, you can add these commands to SYSTARTUP_VMS.COM. On systems with a large number of queues, you might want to include the commands in a separate file named, for example, STARTQ.COM, and include a command in SYSTARTUP_VMS.COM to invoke the queue startup command procedure. The autostart feature simplifies queue startup, and allows you to start queues with fewer commands. Digital recommends you use autostart queues whenever possible to simplify queue startup. For more information on autostart queues, see Section 13.4.

For more specific information on starting queues and enabling autostart for queues in system startup, see Section 13.7.3.

5.2.7.5 Installing Known Images

Section 16.9 explains how 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. You use the Install utility (INSTALL) to install known images. You must reinstall these images each time the system boots.

STARTUP.COM includes a series of INSTALL commands that install certain system programs as known images. You should include any site-specific INSTALL commands in SYSTARTUP_VMS.COM to install images each time the system boots.

Before performing this task, you should understand installed images. For information, see Section 16.9.

Example

The following example shows a command sequence you might include in SYSTARTUP_VMS.COM for installing additional known images:

```
$ INSTALL
  ADD/OPEN/SHARED/HEADER_RESIDENT BASIC
  ADD/OPEN/SHARED/HEADER_RESIDENT FORTRAN
  EXIT
```

5.2.7.6 Installing Resident Images (Alpha Only)

Alpha

Section 16.9.6 explains how you can use the Install utility (INSTALL) to install resident images on Alpha systems. Images must be installed each time the system boots. You can add commands to SYSTARTUP_VMS.COM to automatically perform this task each time the system boots. ♦

5.2.7.7 Setting Up the OpenVMS InfoServer Client Software

Chapter 23 explains the InfoServer system and its uses. If you use the InfoServer system, you will probably perform some setup tasks in SYSTARTUP_VMS.COM. For example, you can add commands to SYSTARTUP_VMS.COM to:

- Start the InfoServer Client for OpenVMS software
- Make remote compact discs available on your system each time the system boots

5.2.7.8 Running the System Dump Analyzer

Section 15.11 explains how to run the System Dump Analyzer utility (SDA) each time the system boots to analyze the system crash dump in case the system failed the last time it was running. You can do this by adding command lines to SYSTARTUP_VMS.COM.

For details, see Section 15.11 and the *System Dump Analyzer Utility Manual*.

Caution

If you use the page file for the crash dump file, you must enter the SDA command COPY when the system reboots, to copy the dump from the page file to another file suitable for analysis. For more information, see Section 15.12.

If you fail to perform the copy operation, pages used to save the crash dump information are not released for paging, and your system might hang because it has insufficient paging space. For more information, see Section 15.12.

Example

The following commands, executed in SYSTARTUP_VMS.COM, invoke SDA, save and analyze the crash dump, and print a listing file:

```
$ ANALYZE/CRASH_DUMP SYS$SYSTEM:SYSDUMP.DMP
    COPY SYS$SYSTEM:SAVEDUMP.DMP        ! Save dump file
    SET OUTPUT DISK1:SYSDUMP.LIS        ! Create listing file
    READ/EXECUTIVE                      ! Read in symbols for kernel
    SHOW CRASH                          ! Display crash information
    SHOW STACK                          ! Show current stack
    SHOW SUMMARY                        ! List all active processes
    SHOW PROCESS/PCB/PHD/REGISTERS      ! Display current process
    EXIT
```

5.2.7.9 Purging the Operator Log File

Section 18.6.1 explains the operator log file. Each time you reboot the system, you create a new version of OPERATOR.LOG. You should devise a plan for regular maintenance of the versions of this file. Add the following command to SYSTARTUP_VMS.COM to purge all but the last two versions of the operator log file:

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

5.2.7.10 Submitting Batch Jobs to Run at Startup Time

Your site might have batch jobs that you want to submit at system startup time. To submit such batch jobs, add SUBMIT commands in the following format to SYSTARTUP_VMS.COM:

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

For example:

```
$ SUBMIT/PRIORITY=255 SYS$MANAGER:SYSDISK_REBUILD
```

This example submits a batch job to run a command procedure each time the system boots. The job is submitted at a high priority to make sure the job is scheduled before any batch jobs users might submit. If possible, submit startup batch jobs at high priority in this way before you start the batch queue.

See Section 13.9.6.4 for information about scheduling of jobs. See the *OpenVMS DCL Dictionary* for information about the SUBMIT command.

5.2.7.11 Creating Systemwide Announcements

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

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

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

Defining SYS$ANNOUNCE

You can define SYS$ANNOUNCE to print an announcement at the beginning of the login procedure for each user. The text prints immediately after a successful dial-in, Ctrl/Y, or carriage return is received. It also prints on LAT terminals when a user connects to a service using the CONNECT command. 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 SYSTARTUP_VMS.COM:

```
$ DEFINE/SYSTEM SYS$ANNOUNCE "SIRIUS VMSCLUSTER 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, the system does not display an announcement.

Caution

Sites requiring moderate or high security should restrict the amount of information displayed in system announcements.

Defining SYS$WELCOME

You can define SYS$WELCOME to display a welcome message whenever a user logs in. The text prints immediately after the user enters the correct password. 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 displays the entire file as a welcoming announcement.

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

```
$ DEFINE/SYSTEM SYS$WELCOME "Welcome to Node RANDOM"
```

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

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

If you do not explicitly define SYS$WELCOME, the terminal displays a standard welcome message similar to the following:

```
Welcome to OpenVMS Version n.n
```

You can add the DECnet for OpenVMS node name to this message by including a translation of the logical name SYS$NODE. When DECnet for OpenVMS starts, it creates the logical name assignment for SYS$NODE.

SYSTARTUP_VMS.COM, supplied as a template with your distribution kit, includes additional command examples for SYS$ANNOUNCE and SYS$WELCOME.

5.2.7.12 Starting Up and Customizing the LAT Protocol Software

Section 24.1 explains the LAT protocol and its uses. To set up your node as a LAT service node and start the LAT protocol software on your system each time the system boots, add the following line to SYSTARTUP_VMS.COM:

```
$ @SYS$STARTUP:LAT$STARTUP.COM
```

When the procedure executes this command, it invokes LAT$STARTUP.COM, which in turn invokes the LAT$CONFIG and LAT$SYSTARTUP command procedures. For more information, see Section 24.4.

5.2.7.13 Starting the DECnet for OpenVMS Network

Before starting the network, you must register your DECnet for OpenVMS license and configure your network. See Section 21.3 for information on setting up the DECnet for OpenVMS network.

If your system participates in a DECnet for OpenVMS network, you might need to start the DECnet software each time your system boots:

- If you are running DECnet for OpenVMS on your system, edit SYSTARTUP_VMS.COM to delete the exclamation point (!) at the beginning of the following command line:

```
$ @SYS$MANAGER:STARTNET.COM
```

- If you are running DECnet/OSI on your system, DECnet is started automatically each time you boot your system.

5.2.7.14 Starting the DIBOL Message Manager

Each node that will execute DIBOL programs must contain a line in SYS$STARTUP:SYSTARTUP_VMS.COM that executes the command procedure SYS$STARTUP:DBLSTRTUP.COM. This command procedure starts the DIBOL Message Manager, used by DIBOL programs as an intermediary in passing messages.

Example

SYSTARTUP_VMS.COM should contain a line as follows:

```
$ @SYS$STARTUP:DBLSTRTUP.COM
```

5.2.7.15 Defining the Number of Interactive Users

By default, when the system starts up, it limits to 64 the number of interactive users allowed to log in.

To change the default value for the number of interactive users that you permit to log in to your system at one time, define the symbol STARTUP$INTERACTIVE_LOGINS to be the maximum number of users in SYSTARTUP_VMS.COM as follows:

STARTUP$INTERACTIVE_LOGINS = = n

For *n*, specify the maximum number of interactive users that can log in at one time.

Note

The number of interactive users is determined by the value authorized by your VAX or Alpha computer license. Therefore, setting the number higher than the license limit has no effect.

The maximum number of interactive users influences the service rating that the LAT software assigns to a service node. The LAT software 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 service node is more able to provide services. For information on LAT software, see Section 24.1.

Example

```
$ STARTUP$INTERACTIVE_LOGINS == 200
```

5.3 Modifying Login Command Procedures to Customize User Environments

In addition to modifying site-specific startup command procedures, you can add commands to **login command procedures** to perform operations each time a user logs in.

Note that although the examples in this section are for DCL (.COM) command procedures, you can substitute other extensions to denote other interfaces such as POSIX.

Command Procedure	Description
SYS$MANAGER:SYLOGIN.COM	A file to which you can add common commands to execute whenever any user logs in. If SYS$MANAGER:SYLOGIN.COM exists and the logical name SYS$SYLOGIN points to this file, SYLOGIN.COM automatically executes when any user logs in.
SYS$LOGIN:LOGIN.COM	A file to which you or users can add commands that are to be executed only when individual users log in to their accounts. If a file named LOGIN.COM exists in a user's SYS$LOGIN directory, it automatically executes when the user logs in.

Caution

If you introduce an error in login procedures, it is possible to accidentally lock yourself out of the system. Section 4.3.2 describes a boot procedure you can use in such an emergency.

SYLOGIN.COM Procedure

As system manager, you create and maintain SYLOGIN.COM. This file is supplied on your distribution kit as a template, and contains commands that you can modify and add to as the needs of your site dictate.

The template for SYSTARTUP_VMS.COM 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
```

Note

If SYLOGIN.COM exits with an error, the user's login command procedure is not executed. If this occurs while the user is logging in to a captive account, the process is terminated because the system environment might not have been set up properly.

If you expect SYLOGIN.COM to cause an error, you must use either SET NOON or ON ERROR commands, and explicitly exit the command procedure with a successful status so that the user's login command procedure can be executed.

LOGIN.COM Procedures

Each user creates and maintains a personal copy of the login command procedure LOGIN.COM. This file must be in the top-level directory for the user's account. As a system manager, helping users to set up a personal copy of LOGIN.COM might be necessary.

Example

See Example 6–1 for a sample SYLOGIN command file. See Example 6–2 and the *OpenVMS User's Manual* for sample LOGIN.COM procedures.

5.4 Customizing Startup Databases with the System Management Utility (SYSMAN)

Startup databases contain information used to start up system software. For example, STARTUP.COM uses information in a startup database named STARTUP$STARTUP_VMS to start the OpenVMS operating system. It uses information in a startup database named STARTUP$STARTUP_LAYERED to start layered products. For more information about startup databases, see Section 5.4.1.

You can use the STARTUP command of the System Management utility (SYSMAN) to customize startup databases as follows:

- Display information in any startup database.
- Create a site-specific startup database.
- Add, modify, or remove elements in the layered product database or site-specific database. (Digital recommends that you do not modify the OpenVMS startup database.)

The following sections describe these tasks.

Before performing these tasks, it helps to understand SYSMAN. For more information, see Section 2.3.1. You should also understand startup databases, in particular, the layered product startup database. For information, see Section 5.4.1 and Section 5.4.2.

5.4.1 Understanding Startup Databases

Three startup database files are provided with the operating system, in the location defined by the logical name SYS$STARTUP:

File	Description
VMS$PHASES.DAT	Determines the order of the phases of startup in a sequential list. This file includes a series of four basic phases (INITIAL, CONFIGURE, DEVICE, and BASEENVIRON) needed to bring the operating system up to a basic working environment, followed by a series of phases for layered products. STARTUP.COM uses this list of phases for startup. *Do not modify this file.*
VMS$VMS.DAT	Equivalent to the logical name STARTUP$STARTUP_VMS. This file contains information about the files used to start the base operating system environment during system startup. *Do not modify this file.* STARTUP$STARTUP_VMS is provided for your information only. Use SYSMAN to display information in this file. For more information, see Section 5.4.5.
VMS$LAYERED.DAT	Equivalent to the logical name STARTUP$STARTUP_LAYERED. This file contains information about files that start site-specific products and layered products. The system uses the information in this file to start layered products during system startup. Section 5.4.2 provides more information about this file. Use SYSMAN to modify this file so that it contains information about all the layered product startup files you want to execute on your system.

If you have site-specific software that you want to manage separately from your layered products, you can use SYSMAN to create an additional startup database.

5.4.2 Understanding the Layered Product Startup Database

The layered product startup database file (referred to by the logical name STARTUP$STARTUP_LAYERED) lists the files and command procedures that start site-specific products and layered products. It contains the following characteristics of each startup file:

- Name of the component file to be run. The file type must be either .EXE or .COM.
- Phase in which the component file is to run. Each phase describes a minimum environment that exists at that point in the startup process, as follows:

  1. BASEENVIRON—Startup tasks execute here. These must be performed before the site-specific startup command procedure, SYSTARTUP_VMS.COM, executes.
  2. LPBEGIN—SYSTARTUP_VMS.COM, the site-specific startup command procedure, executes here, as do any other files that prepare an environment necessary for layered products.
  3. LPMAIN—The majority of layered products execute here. This phase is the default.
  4. LPBETA—Layered products that have a dependency on previously installed products execute here.
  5. END—Products that are dependent on layered products execute here.

  Each phase must meet the prerequisites of the following phase; therefore, the order 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 SYSMAN, be sure that all requisite components occur in a previous phase.
- Mode (or method) by which the component file is to run. Choose one of the following modes:
  - DIRECT (the default, where the command procedure or image is executed immediately)
  - BATCH (valid only for command procedures)
  - SPAWN
- Node restrictions for the component. This is either the node or nodes on which the component file should be run, or the node or nodes on which the component file should *not* be run.
- Parameters passed to the component file for execution. You can pass up to eight parameters, using the following format:

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

  You can omit the parentheses if you pass only a single parameter.

5.4.3 Specifying the Current Startup Database

With SYSMAN, the current database is the one that will be the target for the SYSMAN commands.

You can display or modify STARTUP$STARTUP_LAYERED or database files that you create. You can display STARTUP$STARTUP_VMS, but you should *not* modify it.

By default, the layered product database is the current database. To perform commands on another database, specify it as the current database by entering the SET DATABASE command in the following format:

STARTUP SET DATABASE database

For *database*, specify the name of the database.

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> STARTUP SET DATABASE STARTUP$STARTUP_LOCAL
%SYSMAN-I-NEWCOMPFIL, current component file is now STARTUP$STARTUP_LOCAL
```

5.4.4 Showing the Name of the Target Startup Database

To display which database is the target database, enter the STARTUP SHOW DATABASE command as follows:

```
SYSMAN> STARTUP SHOW DATABASE
```

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> STARTUP SHOW DATABASE
```

5.4.5 Showing the Contents of a Startup Database

To display the contents of the current database, enter the STARTUP SHOW FILE command. For example:

```
SYSMAN> STARTUP SHOW FILE/FULL
```

You can specify various qualifiers for this command to control the amount of information displayed. For more information, see the *OpenVMS System Management Utilities Reference Manual*.

Example

```
$  RUN SYS$SYSTEM:SYSMAN
SYSMAN> STARTUP SHOW FILE/FULL
```

5.4.6 Adding Startup Files to a Startup Database

To add a file to the layered product startup database, use the STARTUP ADD command. The /MODE qualifier specifies the mode of execution for the file. The /PHASE qualifier specifies the phase within system startup when the file is to be executed. For information on the layered product startup phases, see Section 5.4.2.

Do not use this command to modify STARTUP$STARTUP_VMS; this command procedure starts the operating system. The STARTUP MODIFY command requires read and write access to the startup database.

When adding layered product startup files using SYSMAN, be sure that all requisite components occur in a previous phase.

Enter the STARTUP ADD command with appropriate qualifiers. For example:

```
SYSMAN> STARTUP ADD FILE/MODE=DIRECT/PHASE=LPMAIN FOR$LPMAIN_043_STARTUP.COM
```

For information on the valid qualifiers, see the SYSMAN section of the *System Management Utilities Reference Manual*.

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> STARTUP SHOW DATABASE
%SYSMAN-I-DATANAME, STARTUP database is STARTUP$STARTUP_LAYERED

SYSMAN> STARTUP ADD FILE/MODE=DIRECT/PHASE=LPMAIN FOR$LPMAIN_043_STARTUP.COM
```

5.4.7 Changing Information Associated with a Startup File

Once a file is included in the layered product startup database, you can modify the information associated with the file by entering the STARTUP MODIFY command. You can specify any of the following qualifiers to specify the information that is to be changed:

- /MODE
- /NAME=*filespec*
- /PARAMETER=(P1:*arg1*, P2:*arg2*,...)
- /PHASE

Do not use this command to modify STARTUP$STARTUP_VMS. This command requires read and write access to the startup database.

How to Perform This Task

Enter the STARTUP MODIFY command. For example:

```
SYSMAN> STARTUP MODIFY FILE FOR$LPMAIN_043_STARTUP.COM/NODE=ZNODE
```

For information on the qualifiers for this command, see the *OpenVMS System Management Utilities Reference Manual*.

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> STARTUP ADD/MODE=DIRECT/PHASE=LPMAIN FOR$LPMAIN_043_STARTUP.COM
SYSMAN> STARTUP SHOW FILE/NODE
SYSMAN> STARTUP MODIFY FILE FOR$LPMAIN_043_STARTUP.COM/NODE=ZNODE
```

5.4.8 Deleting a Record from a Startup Database

Deleting a record from a startup database prevents a product from starting up. To delete a record, use the STARTUP REMOVE FILE command. This command leaves the startup file intact, but the file is not used in system startup. This command requires read and write access to the startup database. Do not use this command to modify STARTUP$STARTUP_VMS.

To delete a record from a startup database, enter a STARTUP REMOVE FILE command in the following format:

STARTUP REMOVE FILE filename

For *filename*, specify the name of the startup file to be removed.

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> STARTUP SHOW FILE/FULL
SYSMAN> STARTUP REMOVE FILE FOR$LPMAIN_043_STARTUP.COM
SYSMAN> STARTUP SHOW FILE/FULL
SYSMAN> EXIT
```

5.4.9 Preventing a Startup File from Executing

To temporarily prevent a startup file from executing, enter the STARTUP DISABLE command. You can specify the /NODE qualifier to disable the startup file on certain nodes.

This command requires read and write access to the startup database. Do not use this command to modify STARTUP$STARTUP_VMS.

To delete a record from a startup database, enter the STARTUP DISABLE command as follows:

STARTUP DISABLE FILE filename

For *filename*, specify the name of the startup file to be disabled.

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> STARTUP SHOW FILE
SYSMAN> STARTUP DISABLE FILE FOR$LPMAIN_043_STARTUP.COM/NODE=ZURICH
```

5.4.10 Allowing a Previously Disabled Startup File to Execute

If you have disabled a startup file from executing, you can enable it again by using the STARTUP ENABLE command. You can specify the /NODE qualifier to enable the startup file on certain nodes.

This command requires read and write access to the startup database. Do not use this command to modify STARTUP$STARTUP_VMS.

To enable a previously disabled file, enter the STARTUP ENABLE FILE command in the following format:

STARTUP ENABLE FILE filename

For *filename*, specify the name of the file to be enabled.

Example

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> STARTUP ENABLE FILE FOR$LPMAIN_043_STARTUP.COM/NODE=ZURICH
```

5.5 Registering Images That Have System Version Dependencies (VAX Only)

VAX

Applications that run on the OpenVMS operating system sometimes depend on the internal interfaces of a previous version of the operating system. For example, an application might call system routines, reference system data cells, or system data structures. New versions of the operating system might include changes to those interfaces that can break applications depending on those interfaces.

Registering an image records information about the image in a file called the **image registry**. The image activator, INSTALL, and SYSGEN do not check the versions of images that are recorded in the image registry. Image registry allows you to continue to run application images (including main images, shared

libraries, and device drivers) that depend on the internal interfaces of a previous version of the operating system.

The Image Registry facility allows you to register different versions of an image independently. It also allows you to deregister, analyze, and show images in the image registry.

5.5.1 Understanding System Version Dependency and the Image Registry (VAX Only)

Applications that depend on internal interfaces are bound to a particular version of the operating system when the application is linked. Version-dependent images reference both of the following:

- The major version number of the operating system
- A set of component version numbers (version numbers of loadable executive images)

  Images can be linked against SYS.STB in one of two ways:

  - Images linked against SYS.STB with the Linker utility qualifier /SELECTIVE_SEARCH include component version numbers only for those symbols that were referenced at link time. (This is the recommended method.)
  - Images linked against SYS.STB without the /SELECTIVE_SEARCH qualifier include references to all component versions.

When you attempt to run an image, the system checks to determine if the image requires a certain version of the operating system or of system components. If the version of the running system does not match the version requirements of the image, the image fails.

The system also checks version numbers when you attempt to install an image using the Install utility (INSTALL) or connect a device driver using the System Generation utility (SYSGEN).

When you upgrade your system to a new version of the operating system, an image might fail because the new version no longer matches the image's version requirements. However, an image might continue to be compatible with the new operating system version, even if it fails the version check.

Note

Changes in OpenVMS VAX Version 6.0

In OpenVMS VAX Version 6.0, the major version number was not changed; only the version numbers for the following components were increased to reflect changes in these areas:

- FILES_VOLUMES
- MEMORY_MANAGEMENT
- SECURITY

As a result, many version-dependent images built in VMS VAX Version 5.*x* systems (that is, images that do not reference FILES_VOLUMES, MEMORY_MANAGEMENT, or SECURITY) will run on OpenVMS Version 6.0 without registering the images. However, version-dependent images that do reference these components must be registered with the image registry, as explained in this section.

Changes in OpenVMS VAX Version 6.1

In OpenVMS VAX Version 6.1, no version numbers were changed. However, images built on VMS VAX Version 5.*x* systems may need to be registered if they reference FILES_VOLUMES, MEMORY_ MANAGEMENT, or SECURITY.

To continue running a compatible image, you can register the image using the Image Registry facility. You do not need to register images that are linked as part of the installation because they match the current operating system version. However, linking an image during installation does not ensure that system version dependencies do not exist. For information about changes in the current operating system version that might require you to recompile an image or change source code, see the release notes.

Caution

On OpenVMS VAX systems, you must inspect and test an image carefully to avoid system crashes and data corruption. Registering an image does not necessarily make it work; registering only bypasses the version checks.

5.5.2 Using the Image Registry Facility (VAX Only)

To register an image in the image registry, run the command procedure SYS$UPDATE:REGISTER_PRIVILEGED_IMAGE.COM. Enter a command in the following format:

$ @SYS$UPDATE:REGISTER_PRIVILEGED_IMAGE *keyword filename*

where:

keyword — Specifies one or more of the keywords described in Table 5–3, separated by commas.

filename — Specifies the name and location of the image you want to register. The *filename* parameter accepts wildcard characters.

Table 5–3 REGISTER_PRIVILEGED_IMAGE.COM Keywords

Keyword	Action
ANALYZE	Displays version-dependent image names and their subsystem dependencies.
REGISTER	Registers images on the local system.
DEREGISTER	Deletes images from the registry on the local system.
SHOW	Lists the registry content. To display the complete registry content, specify a wildcard (*) for the file name.
CONFIRM	Confirms that each specified image be added to or deleted from the registry (used only with REGISTER and DEREGISTER).
TRACE	Lists each image file for verification purposes (used only with REGISTER and DEREGISTER).
HELP	Describes the supported keywords and provides examples.

If the image does not have a version dependency, the following message is displayed:

```
REGISTER-I-SUMMARY n images examined, n have dependencies
```

In this message, *n* is the number of images examined and the number of images that have dependencies.

Example

The following example adds Rdb/VMS images to the registry:

```
$ @SYS$UPDATE:REGISTER_PRIVILEGED_IMAGE REGISTER SYS$LIBRARY:RDM*
%REGISTER-I-ADDED added RDMPRV to registry
%REGISTER-I-ADDED added RDMSHRP to registry ♦
```

5.6 Customizing the Help Message Database

The Help Message utility (MSGHLP) allows users to quickly access online descriptions of system messages from the DCL prompt. Users with write access to Digital-supplied .MSGHLP$DATA files can customize the Help Message database in more radical ways than general users can. The following sections describe how to perform the following customization tasks:

Task	For More Information
Accessing $STATUS values for uninstalled messages	Section 5.6.1
Creating system-level Help Message database search paths	Section 5.6.2
Deleting Digital-supplied messages	Section 5.6.3
Adding comments to Digital-supplied messages	Section 5.6.4
Changing text in Digital-supplied messages	Section 5.6.5
Adding messages to Digital-supplied database files	Section 5.6.6

Before performing these tasks, you should be familiar with the Help Message utility. For a complete description of Help Message features, basic tasks, and the HELP/MESSAGE command and qualifiers, see the *OpenVMS System Messages: Companion Guide for Help Message Users*. Also see that manual for a description of the files that you must manipulate to customize the Help Message database.

Note

Currently, user-supplied comments or additions to Digital-supplied .MSGHLP$DATA files are not preserved through the next upgrade. However, your own .MSGHLP$DATA files are not affected by future releases.

Note that you can reuse .MSGHLP files to insert your own messages into future Digital-supplied database files. Depending on the data format in future databases, you might also be able to reuse some .MSGHLP files to insert comments.

5.6.1 Accessing $STATUS Values for Uninstalled Messages

Any messages that are not installed as part of the OpenVMS operating system cannot be equated with a value stored in $STATUS until they are recognized by the system. When the Help Message utility attempts to translate the value stored in $STATUS or a value specified with the /STATUS qualifier, the search can fail if the value does not equate to an installed message or a message that was linked into the Help Message utility when it was created by Digital. You can make your system recognize such uninstalled messages. These messages include user-supplied messages, third-party messages, and messages from layered products and certain other OpenVMS facilities.

How to Perform This Task

1. Use the Help Message qualifier /SECTION_FILE=* to include all the OpenVMS supplied message section files that have not already been linked into the main Help Message program, MSGHLP$MAIN.EXE:

   ```
   $ HELP/MESSAGE/SECTION_FILE=*
   ```

   This command generates a user-modifiable object library, SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.OLB. Each module in this library contains a pointer to a message section .EXE file. You can use the /SECTION_FILE qualifier to insert additional modules into this library. (See the following steps.)

   Whenever you use the /SECTION_FILE qualifier to create the object library or add modules to it, file MSGHLP$MESSAGE_SECTIONS.EXE is also automatically created from all the modules in the object library and is placed in your default directory. When you finish modifying the object library, you must copy the final .EXE file to SYS$COMMON:[SYSLIB] (logical name SYS$LIBRARY).

   Thereafter, if Help Message cannot translate a status code and the SYS$LIBRARY:MSGHLP$MESSAGE_SECTION.EXE image exists, Help Message activates it and searches all message section files to which it points. The impact on Help Message search time depends upon the number of files searched.

2. Use the following command as many times as you want to add pointer modules for any other specific message section files to SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.OLB and MSGHLP$MESSAGE_SECTIONS.EXE:

   HELP/MESSAGE/SECTION_FILE=disk:[directory]file-name.EXE

   The default file specification is SYS$MESSAGE:.EXE.

3. Review the contents of the resulting SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.OLB file:

   ```
   $ LIBRARY/LIST MSGHLP$MESSAGE_SECTIONS.OLB
   ```

   The names of the modules in the .OLB file are derived from strings specified in the /SECTION_FILE qualifier.

   Help Message can search a maximum of 42 message section files. Files are searched in this order:

   - The last file activated by a SET MESSAGE command (if any).

- The message section files that Help Message is linked with before it is shipped:
  - SYSMSG.EXE
  - SYSMGTMSG.EXE
  - CLIUTLMSG.EXE
  - PRGMSG.EXE
- Any message section files explicitly linked with MSGHLP$MAIN.EXE or any of the shareable images that it references.
- Any nonduplicate message section files linked with SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.EXE.

  Message section files are searched in the order in which they are listed in the .OLB file. (Order is alphabetical.) If the total count of message section files exceeds 42, message section files listed near the end of the .OLB file might not be searched.

4. If you want, remove any references to message section files in the .OLB file to control which 42 message section files are included in the Help Message search. You might delete a file that you rarely use earlier in the alphabet to ensure that a file later in the alphabet is among the 42 files searched. For example, to delete the NCP (Network Control Program) messages from the .OLB file, use this command:

   ```
   $ LIBRARY/DELETE=NETWRKMSG SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.OLB
   ```

   If you delete a module from the .OLB file, you must execute the HELP /MESSAGE command with the /SECTION_FILE qualifier to generate an updated .EXE file. The qualifier argument can specify either a new file or a file that is already listed in the .OLB file.

5. When your .OLB file reflects the message section files you want Help Message to search, copy the final .EXE file from your account to SYS$LIBRARY:.

Example

The following example demonstrates this sequence of events:

1. Link all OpenVMS supplied message section files.
2. Review the resulting .OLB file.
3. Delete the VVIEFMSG module from the .OLB file.
4. Add the file USERS:[TOOLS]NEW_MSGS.EXE to the list in the .OLB file.
5. Review the revised contents of the .OLB file.
6. Copy the final .EXE file from the local account to SYS$LIBRARY:.

Note that the output from the LIBRARY/LIST commands is omitted from the example.

```
$ HELP/MESSAGE/SECTION_FILE=*
$ LIBRARY/LIST SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.OLB
$ LIBRARY/DELETE=VVIEFMSG SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.OLB
$ HELP/MESSAGE/SECTION_FILE=USERS:[TOOLS]NEW_MSGS.EXE
$ LIBRARY/LIST SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.OLB
$ COPY MSGHLP$MESSAGE_SECTIONS.EXE SYS$LIBRARY:MSGHLP$MESSAGE_SECTIONS.EXE
```

5.6.2 Creating System-Level Database Search Paths

Help Message database files need not reside on the system disk. You can create system logical names to define one or more Help Message search paths to access multiple .MSGHLP$DATA files in different locations.

When Help Message is installed, the OpenVMS messages database file is installed by default at SYS$COMMON:[SYSHLP]MSGHLP$LIBRARY.MSGHLP$DATA. However, this file can optionally be installed on or moved to another disk. The alternate location must be pointed to by logical name MSGHLP$LIBRARY. Use this command to define the logical name:

```
DEFINE/SYSTEM MSGHLP$LIBRARY disk:[directory]MSGHLP$LIBRARY
```

By default, Help Message attempts to look up messages in the default location unless the logical name MSGHLP$LIBRARY is defined. If you do not use the default database location, include the logical name definition command in SYS$MANAGER:SYLOGICALS.COM so that the database is defined each time the system is booted.

Note

If you move MSGHLP$LIBRARY.MSGHLP$DATA to a new location after installation, be sure to set the proper protections on the file and directory so that the database cannot be accidentally deleted or modified. The protections at installation are (RWE, RWE, RE, RE) for the directory and (RWE, RWE, RWE, RE) for the file.

You and other system users can create additional .MSGHLP$DATA files, as described in the *OpenVMS System Messages: Companion Guide for Help Message Users*. None of the .MSGHLP$DATA files need reside on the system disk. You can add new files to a systemwide default database search path defined by MSGHLP$LIBRARY, or you can create specialized search paths to include different configurations of .MSGHLP$DATA files.

A search path definition can include individual file names or can point to one or more directories. If you specify a directory with no file name, Help Message searches all .MSGHLP$DATA files currently found in that directory. Pointing to a directory instead of individual files can minimize your bookkeeping when .MSGHLP$DATA files are added or removed.

To use system resources more efficiently, you can create different search paths for different user groups, depending on which .MSGHLP$DATA files they need to access. You can also set up different directories for different types of messages or for different user groups. For example, you could use three unique logical names to define three different search paths tailored to different user groups:

```
DEFINE/SYSTEM logical-name-1 file-a,file-b,file-c

DEFINE/SYSTEM logical-name-2 file-a,directory-z

DEFINE/SYSTEM logical-name-3 file-x,file-a,directory-y
```

Note

The *first* file you list in a search path is the default database for /INSERT and /DELETE operations that operate on that search path. By default, all other operations access all files in a search path. Specifying a directory

first in a search path risks setting up a default moving target for /INSERT and /DELETE operations if files are added to or deleted from the directory.

Users can select an alternate to the system default database by specifying the /LIBRARY qualifier in the HELP/MESSAGE command. Individual users can also create their own logical name search paths at the process level.

Example

The following example defines a Help Message search path that accesses .MSGHLP$DATA database files in three locations: the Digital-supplied OpenVMS messages at USERS:[TOOLS], the user-supplied file USERS:[NEW_PROJ]OUR_MESSAGES.MSGHLP$DATA, and all .MSGHLP$DATA files in directory TEST:[TRY_ME].

```
$ DEFINE/SYSTEM MSGHLP$LIBRARY USERS:[TOOLS]MSGHLP$LIBRARY,-
_$ USERS:[NEW_PROJ]OUR_MESSAGES.MSGHLP$DATA,TEST:[TRY_ME]
```

5.6.3 Deleting Digital-Supplied Messages from the Database

You can delete Digital-supplied messages from the database to conserve system resources or improve response time.

How to Perform This Task

1. Use the /EXTRACT qualifier to create a .MSGHLP file containing the messages you want to delete from the database. (See the *OpenVMS System Messages: Companion Guide for Help Message Users* for a full description of how to select the contents of the .MSGHLP file.) Some examples follow.

   Use the following syntax to extract all the messages for a specified facility:

   HELP/MESSAGE/FACILITY=facility-name/EXTRACT=filename.MSGHLP

   Use this syntax to extract one or more messages specified by the search string:

   HELP/MESSAGE/EXTRACT=filename.MSGHLP search-string

2. Check the contents of the resulting .MSGHLP file to be sure that it contains only the data that you want to delete from the database. Edit out any messages that you do not want to delete from the database.

3. Use /DELETE to delete the contents of the .MSGHLP file from the database. Include /LIBRARY if the MSGHLP$LIBRARY.MSGHLP$DATA file is not the default database or if it is not the first file in the search path defined by logical name MSGHLP$LIBRARY.

   HELP/MESSAGE/DELETE=filename.MSGHLP

   HELP/MESSAGE/DELETE=filename.MSGHLP/LIBRARY=disk:[directory]filename.MSGHLP$DATA

   Save the .MSGHLP file if you might ever want to add the deleted messages back into the database prior to the next upgrade. You can store the file on tape to conserve disk space. If you delete and then reinsert messages, these messages are treated like user-supplied messages and are displayed with change bars.

   Any Digital-supplied messages that you delete are currently reinserted into the database at the next upgrade. You can delete the messages again using a saved .MSGHLP file or you can create a new .MSGHLP file. Note that if you

keep a .MSGHLP file for future *deletion* purposes only, you need save only the lines prefixed by 1 and 2.

4. To save disk space, you can compress the .MSGHLP$DATA file to close up any free space created by the deletions. Use the following command sequence to compress the file:

   CONVERT disk:[directory]filename.MSGHLP$DATA disk:[directory]filename.MSGHLP$DATA

   PURGE disk:[directory]filename.MSGHLP$DATA

Example

The following example extracts and then deletes all messages for the DDTM (DECdtm services) facility from the default database. The last two commands compress the Digital-supplied database file to conserve disk space after the deletions.

```
$ HELP/MESSAGE/FACILITY=DDTM/EXTRACT=DDTM.MSGHLP
$ HELP/MESSAGE/DELETE=DDTM.MSGHLP
$ CONVERT SYS$COMMON:[SYSHLP]MSGHLP$LIBRARY.MSGHLP$DATA -
_$ SYS$COMMON:[SYSHLP]MSGHLP$LIBRARY.MSGHLP$DATA
$ PURGE SYS$COMMON:[SYSHLP]MSGHLP$LIBRARY.MSGHLP$DATA
```

5.6.4 Adding Comments to Digital-Supplied Messages

You can add comments to Digital-supplied messages documentation. Comments display with change bars immediately following the Digital-supplied description. This feature is a handy way to publicize a site-specific solution for a common problem.

Note

Currently, user-supplied comments to Digital-supplied .MSGHLP$DATA files are not preserved through the next upgrade.

How to Perform This Task

1. Extract the message to which you want to add a comment. The following example extracts hypothetical message NOSNO:

   ```
   $ HELP/MESSAGE/EXTRACT=NOSNO.MSGHLP NOSNO
   ```

2. Edit the .MSGHLP file to add your comment. The .MSGHLP file format uses a unique numerical prefix to designate the message, facility, explanation, and user action sections of the message description. Add your comments at the end using a ‘‘5’’ prefix.

   ```
   1NOSNO, can't ski; no snow
   2XCSKI, XCSKI Program
   3Your attempt to ski failed because there is no snow.
   4Wait until there is snow and attempt the operation again.
   5If you don't want to wait, go to a location where there is
   5snow and ski there.
   5
   5Or, try ice skating instead!
   ```

   Tips for modifying files:

   - Limit your comments to 60 characters per line so that they do not exceed the terminal display area.
   - Use a ‘‘5’’ prefix on blank lines too.

- Do not edit Digital-supplied data. Any such edits are ignored when the comment is added to the database. Section 5.6.5 describes how you can alter Digital-supplied data.

3. Update the database by inserting the updated message:

   ```
   $ HELP/MESSAGE/INSERT=NOSNO.MSGHLP
   ```

   The comment is now displayed following the Digital-supplied message description.

Example

```
$ HELP/MESSAGE/EXTRACT=ACCVIO.MSGHLP ACCVIO
```

[Edit ACCVIO.MSGHLP to add your comment.]

```
$ HELP/MESSAGE/INSERT=ACCVIO.MSGHLP
```

5.6.5 Changing Digital-Supplied Data

You cannot use the procedure described in Section 5.6.4 to alter Digital-supplied information. The recommended way to permanently change Digital-supplied information is to file a Software Performance Report (SPR) or mail a Reader's Comment with your comments or corrections for a future release.

The sequence described in this section allows you to modify Digital-supplied data, with the following results:

- The Digital-supplied message is deleted from the database and your version of the message is inserted.
- The message you modify subsequently displays with change bars to designate it as unsupported, user-supplied data.

Note

Currently, the Digital-supplied message is reinserted into the database at the next upgrade and the user-supplied text is overwritten.

How to Perform This Task

1. Extract the message having the text or description you want to change:

   HELP/MESSAGE/EXTRACT=filename.MSGHLP search-string

2. Check the .MSGHLP file to ensure that your search did not pick up any messages that you do not want to change. Delete any such messages that you want to preserve out of the .MSGHLP file.
3. Delete the Digital-supplied version of the message from the Help Message database by specifying the .MSGHLP file as input. The following command deletes all messages in the .MSGHLP file from the default .MSGHLP$DATA file:

   HELP/MESSAGE/DELETE=filename.MSGHLP

4. Edit the .MSGHLP file to make your changes.
5. Insert the revised message description into the Help Message database:

   HELP/MESSAGE/INSERT=filename.MSGHLP

Your version of the message now appears in the database with change bars to indicate that it is not a Digital-supplied message.

Example

```
$ HELP/MESSAGE/EXTRACT=NOFILES.MSGHLP NOFILES
$ HELP/MESSAGE/DELETE=NOFILES.MSGHLP
```

[Edit NOFILES.MSGHLP to change the text.]

```
$ HELP/MESSAGE/INSERT=NOFILES.MSGHLP
```

5.6.6 Adding Messages to Digital-Supplied Database Files

The *OpenVMS System Messages: Companion Guide for Help Message Users* describes how to create your own .MSGHLP$DATA files to add new messages to the Help Message database. Keeping your messages in a separate file can simplify your messages bookkeeping and ensure that your messages are preserved through future upgrades.

With write access to Digital-supplied .MSGHLP$DATA files, you can alternatively insert your own messages into the Digital-supplied MSGHLP$LIBRARY.MSGHLP$DATA file. However, messages inserted using this technique will currently be overwritten at the next upgrade. You can, however, save your input .MSGHLP files and repeat the insertion process at the next upgrade.

How to Perform This Task

1. Create a .MSGHLP file with your message descriptions in it. (Section 5.6.4 includes an example of the .MSGHLP file format.)
2. Specify your .MSGHLP file as input to update the Digital-supplied .MSGHLP$DATA file. Assuming that MSGHLP$LIBRARY.MSGHLP$DATA is the default, all you must enter is:

   HELP/MESSAGE/INSERT=filename.MSGHLP

Example

```
$ HELP/MESSAGE/INSERT=MYMESSAGES.MSGHLP
```

5.7 Customizing MAIL

You can customize the operation of MAIL on your system, including sending mail directly to a user's mail file if the node the user is on is currently in your VMScluster system.

You customize MAIL by defining the logical name MAIL$SYSTEM_FLAGS as a system and executive mode logical name. For example:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE MAIL$SYSTEM_FLAGS 1
```

The value of the logical name MAIL$SYSTEM_FLAGS is interpreted in the following ways:

Value	Meaning
1	Indicates that this node is part of a homogeneous VMScluster system. In other words, all disks are accessible to the VMScluster system, and a common SYSUAF file and a common MAIL file exist for the VMScluster system. When this bit is set, the system checks the node to which you are sending mail to see if it is currently in the VMScluster system. If the node is in the VMScluster system, the system bypasses DECnet, and the message is written directly to the recipient's mail file. (Note that the node must be up to determine whether it is part of the VMScluster system.)
2	Directs MAIL to set the VMScluster breakthrough flag when issuing the $BRKTHRU service to notify the recipient of new mail. This flag is used only in VMScluster systems and, typically, only in homogeneous VMScluster systems (in other words, flag 1 is also set).
4	Directs MAIL to include the time the message was delivered in the notification message displayed on the recipient's terminal.
† 8	Directs MAIL to use DECnet for OpenVMS VAX address syntax when the system is running DECnet/OSI.
†16	Directs MAIL to use DECnet/OSI address syntax.

†VAX specific

For example, if MAIL$SYSTEM_FLAGS translates to 7, the system selects the first three flags. If the logical name does not translate, the default is 0, which indicates that no flags are set.

VAX

On VAX systems, if neither 8 nor 16 is in the value for MAIL$SYSTEM_FLAGS, the system checks to see whether DECnet for OpenVMS or DECnet/OSI is running on the system and requires the appropriate address syntax. Note that, ordinarily, you do not set either flag. ♦

If the number of new (unread) mail messages displayed on the user's screen is inconsistent with the actual number of new messages, enter the READ/NEW command once (for each nonexistent message) when there is no new mail.

In mail messages sent through DECnet, users can specify node names and user names as logical names. Any access control information in the node name or logical name is ignored.

VAX

On VAX systems running DECnet/OSI, users can also specify node synonyms. ♦

5.8 Setting Correct Time Zone Information on Your System

Beginning with OpenVMS Version 7.0, the DEC C RTL implements its default date/time support for programs compiled with DEC C Version 5.2 using a model based on **Coordinated Universal Time** (UTC), an international standard for measuring time of day.

Note

Even if you do not use the DEC C RTL directly, you *must* set correct time zone information on your system because other system utilities written in the DEC C language might require it.

To set the correct time zone information on your system, use the UTC$TIME_SETUP.COM command procedure to do the following:

1. Set the local time zone for your system.

2. Set the correct **time differential factor** (TDF) for your system.

Using UTC allows the DEC C RTL to implement ANSI C/POSIX functionality. In addition, the UTC model makes the DEC C RTL compatible with the Digital UNIX and POSIX RTL time functions. With a UTC-based system, users can do the following:

- Compute the time in any time zone
- Correctly compute past and future times, taking daylight saving time into account
- Use the ANSI C gmtime routine and make use of the tm-isdst field of the tm structure

The following sections explain these concepts and tasks:

Concept or Task	Section
Understanding time-setting concepts	Section 5.8.1
• Coordinated Universal Time (UTC)	
• Time differential factor (TDF)	
• Daylight saving time and standard time	
• Time zones	
Determining your system's TDF	Section 5.8.2
Using UTC$TIME_SETUP.COM	Section 5.8.3
• Setting the time zone on your system	
• Setting the TDF on your system	
Adjusting for daylight saving time and standard time	Section 5.8.4
Setting time in a VMScluster environment	Section 5.8.5

5.8.1 Understanding Time-Setting Concepts

Understanding some time concepts will help you see the importance of setting the correct time zone and TDF on your system.

5.8.1.1 Coordinated Universal Time

Coordinated Universal Time (UTC) is similar in most respects to Greenwich Mean Time (GMT). Under the UTC time standard, zero hours occurs when the Greenwich Meridian is at midnight. Unlike local time, which can go backward and forward depending on daylight saving time, UTC always increases.

Local times can be up to 12 hours behind Greenwich Mean Time or 13 hours ahead of it.

Because UTC is independent of time zones, you can use UTC around the world; for example, it is 2:00 UTC at the same moment in Paris as well as in Tokyo. You can examine data that is time-stamped with UTC values in Paris and Tokyo without complicated conversions to deal with local time zones.

5.8.1.2 Time Differential Factor

One of the steps in setting the correct time on your system is to calculate a time differential factor (TDF) for your time zone.

The TDF associates each local time zone with UTC; it is the difference between your local system time and UTC. The TDF changes each time your local system time undergoes a time zone change; the UTC, on the other hand, does not change.

The TDF value is expressed in signed (+ or -) *hh:mm* format. The Americas have negative TDFs, while Europe, Africa, Asia, and Australia have positive TDFs.

Section 5.8.2 explains how to select the correct TDF for your time zone.

5.8.1.3 Daylight Saving Time and Standard Time

Typically, you make seasonal changes to the local system time (for example, for daylight saving time and standard time). You usually adjust the local time one hour forward or backward.

You also need to adjust the TDF to compensate for the new local system time. You adjust the TDF in the same direction as the local time; that is, when you add an hour to the local time, you also add an hour to the TDF.

5.8.1.4 Time Zones

Time zones are names for geographical areas that share the same TDF; they also share the same rule or rules for seasonal changes between standard time and daylight saving time.

5.8.2 Determining Your System's Time Differential Factor

You can use the map in Figure 5–1 to determine the TDF for your time zone. If you prefer, you can use the tables in Appendix B in the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* to determine the standard or daylight saving time TDF for your time zone. The procedure described in Section 5.8.3 shows default TDF values for various time zones.

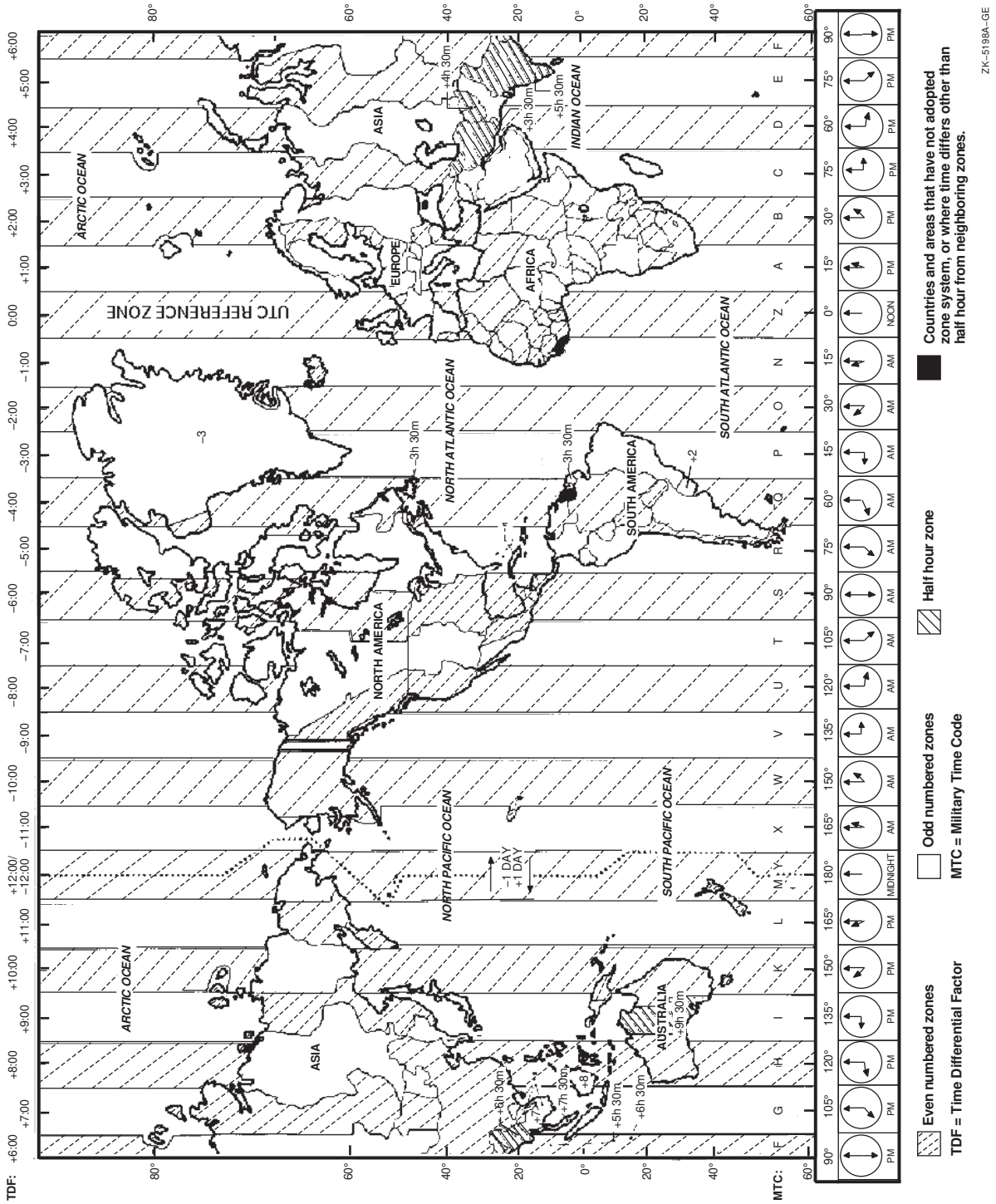
To use the map to determine the TDF of your time zone, follow these steps:

1. Find your location on the map and notice the time zone band at that location.
2. Follow the time zone band to the top of the map, and note the TDF that corresponds to your time zone. For example, the TDF for California is -8; the TDF for Italy is +1.

Some time zones do not have full-hour TDFs. In these cases, find the specific value on the map itself. For example, if you live in Adelaide, Australia, your TDF is +9:30.

If your time zone has daylight saving time, your TDF for daylight saving time is typically +1:00 from the standard time. For example, if your standard time TDF is +2:00, your daylight saving time TDF is +3:00; if your standard time TDF is -7:00, your daylight saving time TDF is -6:00.

Figure 5–1 Time Differential Factor Map



5.8.3 Using UTC$TIME_SETUP.COM

You can use this command procedure to set your local time zone or your TDF, or both. You can also use the procedure to modify your system's local time to adjust for daylight saving or standard time. Finally, you can use it to display the local time and TDF for your system.

If you set the time zone and the TDF on one node in a cluster, the values you set take effect on other nodes in the cluster when those nodes are rebooted.

For your convenience, the instructions for using the command procedure have been split into two sections:

- Setting the time zone
- Setting the TDF

Beginning the Procedure

To use SYS$MANAGER:UTC$TIME_SETUP.COM, follow these steps:

1. Log in to the SYSTEM account, or enter the following command to enable LOG_IO and OPER privileges:

   ```
   $ SET PROCESS/PRIVILEGE=(LOG_IO,OPER)
   ```

2. Enter the following command to start UTC$TIME_SETUP.COM:

   ```
   $ @SYS$MANAGER:UTC$TIME_SETUP.COM

   %UTC-I-UPDTIME, updating Time Zone information in SYS$COMMON:[SYSEXE]
   ```

3. Press Return to accept the default of BOTH, or enter one of the other choices in answer to the following question:

   ```
   Configure which time parameter (TIMEZONE/TDF/BOTH/NONE)? [BOTH]
   ```

   Note

   Digital recommends that you set BOTH the time zone and the TDF. If you set the TDF *without* setting the time zone, the procedure cannot provide default TDF values.

If you answer BOTH or TIMEZONE to the time parameter question in the command procedure, continue with the next section. If you answer TDF to the question, skip to Section 5.8.3.2.

5.8.3.1 Setting the Time Zone on Your System

The local time zone is the location you want to consider your default local time zone. Usually this local time zone is the same as the local time zone in which your system is located.

You set the local time zone by making choices in a command procedure.

The system first displays the following information:

```
Configuring the Local Time Zone

TIME ZONE SPECIFICATION -- Main Time Zone Menu
```

```
 1) Australia       11) GMT            21) Mexico         31) Turkey
 2) Brazil          12) Greenwich      22) NZ             32) UCT
 3) CET             13) Hong Kong      23) NZ-CHAT        33) US
 4) Canada          14) Iceland        24) Navajo         34) UTC
 5) Chile           15) Iran           25) PRC            35) Universal
 6) Cuba            16) Israel         26) Poland         36) W-SU
 7) EET             17) Jamaica        27) ROC            37) WET
 8) Egypt           18) Japan          28) ROK            38) Zulu
 9) Factory         19) Libya          29) Singapore
10) GB-Eire         20) MET            30) SystemV

 0) None of the above
```

Table 5–4 lists and describes the acronyms that appear in the Main Time Zone Menu.

Table 5–4 Time Zone Acronyms

Time Zone Acronym	Description
CET	Central European Time
EET	Eastern European Time
Factory	Specifies no time zone
GB-Eire	Great Britain/Ireland
GMT	Greenwich Mean Time
NZ	New Zealand
NZ-CHAT	New Zealand, Chatham Islands
MET	Middle European Time
PRC	Peoples Republic of China
ROC	Republic of China
ROK	Republic of Korea
SystemV	Specific to System V operating system
UCT	Coordinated Universal Time
US	United States
UTC	Coordinated Universal Time
Universal	Coordinated Universal Time
W-SU	Middle European Time
WET	Western European Time

To select a time zone, follow these steps:

1. Enter a number after the following question; for example, 33, for the United States:

   ```
   Select the number above that best describes your location: 33
   ```

   If you enter 0, the system defaults to GMT.

2. If you enter a country that has more than one time zone, the system displays a message and asks for a confirmation like the following (if not, skip to the explanation following step 4):

   ```
   You selected US as your time zone.
   Is this correct? (Yes/No) [YES]: Return
   ```

3. The system displays areas with different time zones and asks you to select your area. Enter a number, for example, 6:

```
US Time Zone Menu

   1) Alaska           4) Central          7) Hawaii          10) Mountain
   2) Aleutian         5) East-Indiana     8) Indiana-Starke  11) Pacific
   3) Arizona          6) Eastern          9) Michigan        12) Samoa

   0) None of the above

Select the number above that best describes your location: 6
```

4. Confirm the displayed information, or enter another number after the following; for example:

```
You selected US/Eastern as your time zone.
Is this correct? (Yes/No) [YES]: Return
```

When you confirm the last statement, the system redefines the following system logical names:

- sys$localtime
- sys$posixrules

The DEC C RTL uses these logical names to compute the time zone rules for your applications. The system also writes this information to sys$timezone.dat. The system uses this file to reset sys$localtime and sys$posixrules when you reboot.

The system then displays the TDFs for standard and daylight saving time that correspond to the time zone you have selected:

```
    Default Time Differential Factor for standard time is -5:00.
   *Default Time Differential Factor for daylight saving time is -4:00.
```

* The system displays daylight saving time only for time zones that use daylight saving time.

5.8.3.2 Setting the TDF on Your System

If you answer TDF or BOTH to the time parameter question at the beginning of the command procedure, the system displays prompts for the TDF on your system.

To set the TDF, answer the following questions:

1. Select option 2 to display the TDF the system has calculated for you:

```
Configuring the Time Differential Factor (TDF)

Enter ? anytime for help

[0]     Exit
[1]     Set the Time Differential Factor
[2]     Display the Time Differential Factor

Please pick an option number [2]: 2
```

The system displays information like the following:

```
SYSTEM TIME DIFFERENTIAL FACTOR = -4:00  (-14400 seconds).
LOCAL SYSTEM TIME               = 22-SEP-1995 10:49:45.20.
```

2. Select option 1 to verify the TDF displayed or to enter a new one:

```
Configuring the Time Differential Factor (TDF)

Enter ? anytime for help

[0]     Exit
[1]     Set the Time Differential Factor
[2]     Display the Time Differential Factor

Please pick an option number [2]: 1
```

   The system then displays the following information:

```
The Time Differential Factor (TDF) is the difference between your
system time and Coordinated Universal Time (UTC).  UTC is similar
in most repects to Greenwich Mean Time (GMT).

The TDF is expressed as hours and minutes, and should be entered
in the hh:mm format.  TDFs for the Americas will be negative
(-3:00, -4:00, etc.); TDFs for Europe, Africa, Asia and Australia
will be positive (1:00, 2:00, etc.).
```

   The system displays the following question only if you set the time zone as well. However, some time zones do not have daylight saving time; if they do not, the system also does not display this question.

3. Answer Yes or No to the following question:

```
Is Daylight Savings time in effect? (Yes/No):
```

4. After the following prompt, either press Return to accept the displayed default or enter the correct TDF. (If you have not set the time zone, the system does not display a default TDF value.)

```
Enter the Time Differential Factor [-4:00]:
```

   The system then explains the need to modify the system time as well for season time changes:

```
If this is a seasonal time change, it may also be necessary to
modify the system time.  Generally, seasonal time changes result
in adding 1:00 hour, or adding -1:00 hour to the system time.
```

5. To the following question, answer Yes if you need to modify the local system time or No if you do not:

```
Do you wish to modify the local system time [N]:
```

   If you answer Yes, the system will lead you through a dialogue similar to the one in Section 5.8.4.

   If you answer No, the system next displays the new TDF:

```
NEW SYSTEM TIME DIFFERENTIAL FACTOR = -4:00.
```

6. Answer Yes to confirm the displayed TDF or No if it is incorrect:

```
Is this correct? [Y]:
```

   If you answer No, the system returns to step 1 in this section.

   If you answer Yes, the system displays both the TDF and the local system time.

```
SYSTEM TIME DIFFERENTIAL FACTOR = -4:00  (-14400 seconds).
 LOCAL SYSTEM TIME               = 22-SEP-1995 10:52:37.36.
```

5.8.4 Adjusting for Daylight Saving Time and Standard Time

To adjust the local time to daylight saving or standard time, you can invoke the command procedure SYS$EXAMPLES:DAYLIGHT_SAVINGS.COM to do both of the following:

- Adjust the TDF
- Modify the local time

DAYLIGHT_SAVINGS.COM allows you to do either one of the following:

- Make the changes immediately. (Usually, however, you would use UTC$TIME_SETUP.COM to make changes immediately by answering Yes to Question 5 in Section 5.8.3.2.)
- Queue a batch job to make the changes at a future time. (This is the most common use of this command procedure.)

The following example of DAYLIGHT_SAVINGS.COM shows answers that cause the procedure to queue a batch job, DST_CHANGE, which will execute when the time changes from standard time to daylight saving time. Many of the questions are similar to those explained in Section 5.8.3.2.

In the example, the initial TDF value is -5:00. The local date and time are any time from the date in 1993 when the change to standard time was made, until 23-APR-1995:02:00, when the change to daylight saving time will be made.

```
$ SYS$EXAMPLES:DAYLIGHT_SAVINGS

        This procedure queues a batch job that changes the system time
        and system time differential around a daylight saving time
        change.  Press the question mark (?) key at any time for help;
        hit Control-C to exit.

        The Time Differential Factor (TDF) is the difference
        between your system time and Coordinated Universal Time (UTC).
        The difference is expressed in hh:mm format.  The Americas
        have negative offsets from UTC, while Europe, Africa, Asia
        and Australia have positive offsets from UTC.

       * Enter the Time Differential Factor: -4:00 Return

        If this is a seasonal time change, it may also be
        necessary to modify the system time.  Generally,
        seasonal time changes result in adding 1:00 hour,
        or adding -1:00 hour to the local time.

        * Do you wish to modify the local system time [N]: Y Return

        Enter the time value you would like  to add to
        the local time.  The value can be a positive or
        a negative (-hh:mm) value.

        * Enter the time value: +1:00 Return

        The process to modify your time zone offset and local
        time (if supplied) can occur now or in the future.
        Press Return to run the job now.

        * Enter the run time in the DD-MMM-YYYY:HH:MM:SS format: 23-apr-1995:02:00 Return
NEW SYSTEM TIME DIFFERENTIAL FACTOR = -4:00.
ADDING 1:00 TO THE LOCAL TIME.
JOB RUN TIME : 23-APR-1995:02:00

        * Continue? [Y]: Y Return
```

```
Job DST_CHANGE (queue SYS$BATCH, entry 2) holding until 23-APR-1995 02:00
Batch Job DST_CHANGE scheduled to run at 23-APR-1995:02:00
$
$!!The batch job DST_CHANGE will run on 23-Apr-1995 at 02:00
```

5.8.5 Setting Time in a VMScluster Environment

The TDF and the local time must be the same on all nodes in a VMScluster environment. You can use the System Management utility (SYSMAN) DO command to invoke the command procedure UTC$CONFIGURE_TDF.COM on one node in a cluster to do the following for one or more nodes in the cluster:

- Display the TDF
- Set or change the TDF
- Modify the local time

Note that UTC$CONFIGURE_TDF.COM is normally run by UTC$TIME_SETUP.COM when you select the TDF or BOTH option.

You specify the function SET or SHOW, the TDF value, and the local time modification as command procedure parameters. Note that you must express the TDF and the change to the local time in minutes format, not in hours and minutes *hh:mm* format, as you do if you use the command procedure interactively. For example, for +2:00, you would enter +120 or 120.

Examples

1. The following example changes the TDF clusterwide by +1:00 to 11:00 and also moves the local time ahead by 1 hour, as you would do if you were changing from standard time to daylight saving time. The initial TDF value is +10:00, and the local date and time are 12-MAR-1995:15:20.

```
$  RUN SYS$SYSTEM:SYSMAN

            SYSMAN>  SET ENVIRONMENT/CLUSTER

%SYSMAN-I-ENV, Current command environment:
         Clusterwide on local cluster
         Username SMITH will be used on nonlocal nodes

SYSMAN>  DO @SYS$MANAGER:UTC$CONFIGURE_TDF SET +660 +60
```

2. The following example displays the TDF for each node in the cluster:

```
$  RUN SYS$SYSTEM:SYSMAN
SYSMAN>  SET ENVIRONMENT/CLUSTER

%SYSMAN-I-ENV, Current command environment:
         Clusterwide on local cluster
         Username SMITH will be used on nonlocal nodes

SYSMAN> DO @SYS$MANAGER:UTC$CONFIGURE_TDF SHOW
```

   The system will display the TDF and the local time on each node in the cluster.

3. The following command procedure can be submitted as a batch job to run at a future time to do the following:

   - Change the TDF clusterwide by –1:00, to –4:30. (The initial TDF value is –3:30.)

- Move the local time back by 1 hour 23-JAN1995.

```
$! TO_STANDARD_TIME.COM
$! Command procedure to change the TDF by -1:00 and also modify local time
$ RUN SYS$SYSTEM:SYSMAN
        SET ENVIRONMENT/CLUSTER
        DO @SYS$MANAGER:UTC$CONFIGURE_TDF SET -270 -60
        EXIT
```

Note: Because UTC$CONFIGURE_TDF.COM accepts time in minutes, -4:30 is expressed as -((4 × 60) + 30) or -270; -1:00 is expressed as -60.

The following command submits this command procedure as a batch job to run at 23-JAN1995:02:00:

```
$ SUBMIT/AFTER=23-JAN1995:02:00 TO_STANDARD_TIME.COM
```

5.9 Setting Time Using the Battery-Backed Watch (BBW) (Alpha Only)

Alpha

The OpenVMS Alpha architecture maintains the current date and time in the Battery-Backed Watch (BBW) across power failures and system downtime. The BBW is functionally equivalent to the Time of Day Register (TODR) that the VAX architecture uses. One difference, however, is the BBW's constraint on the date range.

The BBW provides sufficient storage capability for only a century. The OpenVMS Alpha system date range has been redefined as 1957 to 2056 to maintain correct leap-year processing and to provide for the millennial transition.

In addition, the OpenVMS Alpha timing mechanisms have been changed to allow 2-digit year support in the $ASCTIM system service and the DCL SET TIME command. (Prior to this change, only 4-digit year fields were allowed.) With 2-digit support, you need to enter only the last 2 digits of a year. The century associated with the year field is derived from the placement of the 2 digits in the 1957-2056 date range. For example:

```
$ SET TIME = 1-NOV-95
```

In this example, 95 is the equivalent of 1995.

```
$ SET TIME = 1-NOV-05
```

In this example, 05 is the equivalent of 2005.♦

5.10 Choosing Languages, and Date and Time Formats

You can specify languages other than English. From the list that the system manager defines, users can later select a language that they want to display.

You can also select the time and date formats for many SHOW commands from a predefined list or define new time and date formats.

Note

The SHOW TIME command does not include this feature because the SHOW TIME command is processed completely by DCL, which does not have access to the LIB$ routines necessary to format the output.

In addition, the SHOW commands for batch and print operations were modified to include, in the default time-stamp, seconds as well as hours and minutes. These new features were not previously documented.

For example, rather than 15-JAN-1995 10:16:25.14, you can use a different format, such as the following:

```
$ SHOW USERS

      OpenVMS User Processes at JANUARY 15, 1995 10:16 AM
    Total number of users = 7,  number of processes = 11

 Username     Node     Interactive  Subprocess   Batch
 MCDERMOT    ARD26B            1
 PASTERNAK   ARD26B            -         2         1
   .
   .
   .
```

Later, users can override the system defaults set up by the system manager and select their own date and time formats.

Steps to Change Languages, and Dates and Times

For languages other than English or date/time formats other than the defaults, you must complete these steps:

Note

Digital recommends that you include these steps within the command procedure SYS$MANAGER:SYSTARTUP_VMS.COM.

---

1. Define the logical name SYS$LANGUAGES (plural) to specify the list of languages the users on your system might want to use. (If the language is English, skip this step.)
2. Invoke the command procedure SYS$MANAGER:LIB$DT_STARTUP.COM, which:
   - Defines output formats you can use to customize the display of dates and times
   - Loads support for languages other than English that you define with the SYS$LANGUAGES logical
3. Define date and time formats for the system using either:
   - User-defined formats
   - Predefined formats

5.10.1 Specifying Languages Other Than English

Use the SYS$LANGUAGES (plural) logical to define a list of languages other than English. (From this list, users can later select a language to be displayed on their processes, as explained in Section 5.10.4.)

Since English is the default language and must therefore always be available, English spellings are not taken from logical name translations; rather, they are looked up in an internal table.

The available languages and their logical names are shown in Table 5–5.

Table 5–5 Available Languages

Language	Equivalence String for SYS$LANGUAGES
Austrian	AUSTRIAN
Danish	DANISH
Dutch	DUTCH
Finnish	FINNISH
French	FRENCH
French Canadian	CANADIAN
German	GERMAN
Hebrew	HEBREW
Italian	ITALIAN
Norwegian	NORWEGIAN
Portuguese	PORTUGUESE
Spanish	SPANISH
Swedish	SWEDISH
Swiss French	SWISS_FRENCH
Swiss German	SWISS_GERMAN

For example, to specify the French, German, and Italian languages, you must define SYS$LANGUAGES:

```
$ DEFINE SYS$LANGUAGES FRENCH, GERMAN, ITALIAN
```

To add another language, for example, FINNISH, you must add FINNISH to the definition of SYS$LANGUAGES and execute the command procedure again.

Section 5.10.4 explains how users can select a language to be displayed for their process.

5.10.2 Invoking LIB$DT_STARTUP.COM

The SYS$MANAGER:LIB$DT_STARTUP.COM command procedure defines the possible choices for the following logicals:

- SYS$LANGUAGES

  The system loads the languages you have selected using the SYS$LANGUAGES (plural) logical.

  Users can later select their own choice of languages by defining the SYS$LANGUAGE (singular) logical, as explained in Section 5.10.4.

- LIB$DT_FORMAT

  The system loads output formats that you can then use to specify default system formats.

  Users can later define their own formats, as explained in Section 5.10.4.

To invoke the command procedure, enter the following command:

```
$ @SYS$MANAGER:LIB$DT_STARTUP
```

If the translation of SYS$LANGUAGES fails, then English is used. If the translation of LIB$DT_FORMAT or any logical name relating to format fails, the OpenVMS standard ($ASCTIM) representation of the date and time is used, that is, *dd-mmm-yyyy hh:mm:ss.cc*.

5.10.3 Defining System Default Date and Time Formats

To define default date and time formats, you can use either user-defined formats, which are shown in Table 5–6, or predefined formats, which are shown in Table 5–7 and Table 5–8.

To select a format for a date, time, or both, you must define the LIB$DT_ FORMAT logical name using the following logicals:

- LIB$DATE_FORMAT_*nnn*, where *nnn* ranges from 001 to 040
- LIB$TIME_FORMAT_*nnn*, where *nnn* ranges from 001 to 020

The order in which these logical names appear in the definition of LIB$DT_ FORMAT determines the order in which they are output. A single space is inserted into the output string between the two elements if the definition specifies that both are output. For example, to define systemwide formats:

```
$ DEFINE/SYSTEM LIB$DT_FORMAT LIB$DATE_FORMAT_006, LIB$TIME_FORMAT_012
```

This definition causes the date to be displayed systemwide in the specified format, followed by a space and the time in the specified format. For example:

```
13 JAN 95 9:13 AM
```

Section 5.10.4 explains how users can select their own date and time formats to be displayed for their process.

5.10.3.1 Defining Your Own Format

To define your own format, define LIB$DATE_FORMAT_*nnn* and LIB$TIME_ FORMAT_*nnn*, using the mnemonics shown in Table 5–6. Replace *nnn* with a number of your choice.

Note

For user-defined formats, Digital recommends that you use values of _500 and above for _*nnn*.

Table 5–6 Format Mnemonics

Date	Explanation
!D0	Day, Zero-Filled
!DD	Day, No Fill
!DB	Day, Blank-Filled
!WU	Weekday, Uppercase
!WAU	Weekday, Abbreviated, Uppercase
!WC	Weekday, Capitalized
!WAC	Weekday, Abbreviated, Capitalized

(continued on next page)

Table 5–6 (Cont.) Format Mnemonics

Date	Explanation
!WL	Weekday, Lowercase
!WAL	Weekday, Abbreviated, Lowercase
!MAU	Month, Alphabetic, Uppercase
!MAAU	Month, Alphabetic, Abbreviated, Uppercase
!MAC	Month, Alphabetic, Capitalized
!MAAC	Month, Alphabetic, Abbreviated, Capitalized
!MAL	Month, Alphabetic, Lowercase
!MAAL	Month, Alphabetic, Abbreviated, Lowercase
!MN0	Month, Numeric, Zero-Filled
!MNM	Month, Numeric, No Fill
!MNB	Month, Numeric, Blank-Filled
!Y4	Year, 4 Digits
!Y3	Year, 3 Digits
!Y2	Year, 2 Digits
!Y1	Year, 1 Digit

Time	Explanation
!H04	Hours, Zero-Filled, 24-Hour Clock
!HH4	Hours, No Fill, 24-Hour Clock
!HB4	Hours, Blank-Filled, 24-Hour Clock
!H02	Hours, Zero-Filled, 12-Hour Clock
!HH2	Hours, No Fill, 12-Hour Clock
!HB2	Hours, Blank-Filled, 12-Hour Clock
!M0	Minutes, Zero-Filled
!MM	Minutes, No Fill
!MB	Minutes, Blank-Filled
!S0	Seconds, Zero-Filled
!SS	Seconds, No Fill
!SB	Seconds, Blank-Filled
!C7	Fractional Seconds, 7 Digits
!C6	Fractional Seconds, 6 Digits
!C5	Fractional Seconds, 5 Digits
!C4	Fractional Seconds, 4 Digits
!C3	Fractional Seconds, 3 Digits
!C2	Fractional Seconds, 2 Digits

(continued on next page)

Table 5–6 (Cont.) Format Mnemonics

Time	Explanation
!C1	Fractional Seconds, 1 Digit
!MIU	Meridiem Indicator, Uppercase
!MIC	Meridiem Indicator, Capitalized (mixed case)
!MIL	Meridiem Indicator, Lowercase

5.10.3.2 Using Predefined Formats

Table 5–7 lists all predefined date format logical names, their formats, and examples of the output generated using those formats. The mnemonics used to specify the formats are listed in Table 5–6.

Table 5–7 Predefined Output Date Formats

Date Format Logical	Format	Example
LIB$DATE_FORMAT_001	!DB-!MAAU-!Y4	13-JAN-1995
LIB$DATE_FORMAT_002	!DB !MAU !Y4	13 JANUARY 1995
LIB$DATE_FORMAT_003	!DB.!MAU !Y4	13.JANUARY 1995
LIB$DATE_FORMAT_004	!DB.!MAU.!Y4	13.JANUARY.1995
LIB$DATE_FORMAT_005	!DB !MAU !Y2	13 JANUARY 95
LIB$DATE_FORMAT_006	!DB !MAAU !Y2	13 JAN 95
LIB$DATE_FORMAT_007	!DB.!MAAU !Y2	13.JAN 95
LIB$DATE_FORMAT_008	!DB.!MAAU.!Y2	13.JAN.95
LIB$DATE_FORMAT_009	!DB !MAAU !Y4	13 JAN 1995
LIB$DATE_FORMAT_010	!DB.!MAAU !Y4	13.JAN 1995
LIB$DATE_FORMAT_011	!DB.!MAAU.!Y4	13.JAN.1995
LIB$DATE_FORMAT_012	!MAU !DD, !Y4	JANUARY 13, 1995
LIB$DATE_FORMAT_013	!MN0/!D0/!Y2	01/13/95
LIB$DATE_FORMAT_014	!MN0-!D0-!Y2	01-13-95
LIB$DATE_FORMAT_015	!MN0.!D0.!Y2	01.13.95
LIB$DATE_FORMAT_016	!MN0 !D0 !Y2	01 13 95
LIB$DATE_FORMAT_017	!D0/!MN0/!Y2	13/01/95
LIB$DATE_FORMAT_018	!D0/!MN0-!Y2	13/01-95
LIB$DATE_FORMAT_019	!D0-!MN0-!Y2	13-01-95
LIB$DATE_FORMAT_020	!D0.!MN0.!Y2	13.01.95
LIB$DATE_FORMAT_021	!D0 !MN0 !Y2	13 01 95
LIB$DATE_FORMAT_022	!Y2/!MN0/!D0	95/01/13
LIB$DATE_FORMAT_023	!Y2-!MN0-!D0	95-01-13
LIB$DATE_FORMAT_024	!Y2.!MN0.!D0	95.01.13
LIB$DATE_FORMAT_025	!Y2 !MN0 !D0	95 01 13
LIB$DATE_FORMAT_026	!Y2!MN0!D0	950113

(continued on next page)

Table 5–7 (Cont.) Predefined Output Date Formats

Date Format Logical	Format	Example
LIB$DATE_FORMAT_027	/!Y2.!MN0.!D0	/95.01.13
LIB$DATE_FORMAT_028	!MN0/!D0/!Y4	01/13/1995
LIB$DATE_FORMAT_029	!MN0-!D0-!Y4	01-13-1995
LIB$DATE_FORMAT_030	!MN0.!D0.!Y4	01.13.1995
LIB$DATE_FORMAT_031	!MN0 !D0 !Y4	01 13 1995
LIB$DATE_FORMAT_032	!D0/!MN0/!Y4	13/01/1995
LIB$DATE_FORMAT_033	!D0-!MN0-!Y4	13-01-1995
LIB$DATE_FORMAT_034	!D0.!MN0.!Y4	13.01.1995
LIB$DATE_FORMAT_035	!D0 !MN0 !Y4	13 01 1995
LIB$DATE_FORMAT_036	!Y4/!MN0/!D0	1995/01/13
LIB$DATE_FORMAT_037	!Y4-!MN0-!D0	1995-01-13
LIB$DATE_FORMAT_038	!Y4.!MN0.!D0	1995.01.13
LIB$DATE_FORMAT_039	!Y4 !MN0 !D0	1995 01 13
LIB$DATE_FORMAT_040	!Y4!MN0!D0	19950113

Table 5–8 lists all predefined time format logical names, their formats, and examples of the output generated using those formats.

Table 5–8 Predefined Output Time Formats

Time Format Logical	Format	Example
LIB$TIME_FORMAT_001	!H04:!M0:!S0.!C2	09:13:25.14
LIB$TIME_FORMAT_002	!H04:!M0:!S0	09:13:25
LIB$TIME_FORMAT_003	!H04.!M0.!S0	09.13.25
LIB$TIME_FORMAT_004	!H04 !M0 !S0	09 13 25
LIB$TIME_FORMAT_005	!H04:!M0	09:13
LIB$TIME_FORMAT_006	!H04.!M0	09.13
LIB$TIME_FORMAT_007	!H04 !M0	09 13
LIB$TIME_FORMAT_008	!HH4:!M0	9:13
LIB$TIME_FORMAT_009	!HH4.!M0	9.13
LIB$TIME_FORMAT_010	!HH4 !M0	9 13
LIB$TIME_FORMAT_011	!H02:!M0 !MIU	09:13 AM
LIB$TIME_FORMAT_012	!HH2:!M0 !MIU	9:13 AM
LIB$TIME_FORMAT_013	!H04!M0	0913
LIB$TIME_FORMAT_014	!H04H!M0m	09H13m
LIB$TIME_FORMAT_015	kl !H04.!M0	kl 09.13
LIB$TIME_FORMAT_016	!H04H!M0’	09H13’
LIB$TIME_FORMAT_017	!H04.!M0 h	09.13 h
LIB$TIME_FORMAT_018	h !H04.!M0	h 09.13

(continued on next page)

Table 5–8 (Cont.) Predefined Output Time Formats

Time Format Logical	Format	Example
LIB$TIME_FORMAT_019	!HH4 h !MM	9 h 13
LIB$TIME_FORMAT_020	!HH4 h !MM min !SS s	9 h 13 min 25 s

5.10.4 User Definitions of Language, and Date and Time Formats

A user can specify a choice of language by defining the SYS$LANGUAGE logical. For example:

```
$ DEFINE SYS$LANGUAGE FRENCH
```

A user can also specify a date and time format by defining the LIB$DT_FORMAT logical. For example:

```
$ DEFINE LIB$DT_FORMAT LIB$DATE_FORMAT_002, LIB$TIME_FORMAT_006
```

5.11 Saving Your Customization

Once you have installed and customized your system, Digital recommends that you back up your system disk. To do so, follow the instructions in Section 10.17.

VAX

On VAX systems, back up the console volume (if applicable). If your computer has a console storage device, make a backup copy of your console volume in case your original becomes corrupted. The operating system provides a command procedure called CONSCOPY.COM (in the SYS$UPDATE directory), which copies your console volume to a blank one.

The procedure for backing up the console volume varies for different computers. For specific instructions on backing up the console volumes, see the upgrade and installation supplement for your VAX computer. ♦

6
Managing User Accounts

This chapter describes how to grant access to users on your system. It tells you how to add and maintain user accounts, and it describes the privileges that you can give and the resources that you can allocate to the users on your system. It also describes the system management features of the OpenVMS Mail utility (MAIL).

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Managing system-supplied UAF Accounts	Section 6.4
Preparing to add user accounts	Section 6.5
Adding user accounts	Section 6.6
Using command procedures for interactive accounts	Section 6.7.1
Modifying a user account	Section 6.7.2
Maintaining the user environment	Section 6.7.4
Deleting a user account	Section 6.7.5
Restricting the use of accounts	Section 6.8
Using login procedures for restricted accounts	Section 6.8.4
Setting up special accounts	Section 6.9
Managing MAIL	Section 6.10
Managing system resources	Section 6.11

This chapter explains the following concepts:

Concept	Section
Understanding the user authorization file (UAF)	Section 6.1
Understanding the protection of authorization files	Section 6.2
Understanding UAF login checks	Section 6.3
Understanding system-supplied UAF accounts	Section 6.4.1
Understanding account security	Section 6.5.3
Understanding network proxy accounts	Section 6.9.3
Understanding pages and pagelets	Section 6.11.1

6.1 Understanding the User Authorization File (UAF)

The system user authorization file (UAF), SYS$SYSTEM:SYSUAF.DAT, contains user **account records**. Each record consists of fields that provide information about the account's user name, login characteristics, login restrictions, and resource control attributes. You specify the account user name as a parameter to AUTHORIZE commands; the other fields are specified as qualifiers to AUTHORIZE commands.

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

You can assign the following resource control attributes in the UAF record:

- Priority
- Limits and quotas
- Privileges

The following sections briefly discuss these resource control attributes.

6.1.1 Priority

A user's priority is the **base process priority** that the system uses to schedule computer time for the process associated with the user's account.

VAX

On VAX systems, priorities range in value from a low of 0 to a high of 31. 0 through 15 are timesharing priorities; 16 through 31 are real-time priorities. ♦

Alpha

On Alpha systems, priorities range in value from a low of 0 to a high of 63. 0 through 15 are timesharing priorities; 16 through 63 are real-time priorities. ♦

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

Leave the base priority at the default of 4 for timesharing accounts.

6.1.2 Limits and Quotas

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

A process passes on its resources to a subprocess (for example, when you create a subprocess with the SPAWN command) in one of several ways, depending on the resource type. Table 6–1 lists the different resource types.

Table 6–1 Resource Types

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

(continued on next page)

Table 6–1 (Cont.) Resource Types

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

Normally, leave limits at their default values. For the default values for the system and user accounts, see the sample SYSTEM and DEFAULT user authorization file records supplied with the Authorize utility on your distribution kit. Also see Section 6.11 for a full description of limits and quotas.

6.1.3 Privileges

Privileges determine what functions users are authorized to perform on the system. System manager functions require privileges that are denied to most users. Because the SYSTEM account has full privileges by default, exercise caution in using it. For example, if you log in to the SYSTEM account, you can modify and delete any file regardless of its protection.

Table 6–2 categorizes system privileges and includes a brief definition of the activity permitted with each privilege. See the *OpenVMS Guide to System Security* for a full description of privileges.

Table 6–2 System Privileges

Category	Privilege	Activity Permitted
None	None	None requiring privileges
Normal	NETMBX	Create network connections
	TMPMBX	Create temporary mailbox
Group	GROUP	Control processes in the same group
	GRPPRV	Group access through system protection field
Devour	ACNT	Disable accounting
	ALLSPOOL	Allocate spooled devices
	BUGCHK	Make machine check error log entries
	EXQUOTA	Exceed disk quotas
	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
	AUDIT	Generate audit records
	OPER	Perform operator functions
	PSWAPM	Change process swap mode
	SECURITY	Control any process
	SYSLCK	Perform security-related functions
	WORLD	Lock systemwide resources

(continued on next page)

Table 6–2 (Cont.) System Privileges

Category	Privilege	Activity Permitted
Objects	DIAGNOSE	Diagnose devices
	IMPORT	Mount a nonlabeled tape volume
	MOUNT	Execute mount volume QIO
	SYSGBL	Create systemwide global sections
	VOLPRO	Override volume protection
	READALL	Bypass existing restrictions to read an object
All	BYPASS	Disregard protection
	CMEXEC	Change to executive mode
	CMKRNL	Change to kernel mode
	DETACH	Create detached processes of arbitrary UIC
	DOWNGRADE	Write to a lower secrecy object or lower an object's classification
	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 all system objects
	SETPRV	Enable any privilege
	SHARE	Access devices allocated to other users
	SYSNAM	Insert system logical names in the name table
	SYSPRV	Access objects through system protection field
	UPGRADE	Write to a higher integrity object or raise an object's integrity level

Because certain images (such as SET.EXE) require access to the system UAF and are normally installed with the SYSPRV privilege, make sure you always grant system access to SYSUAF.DAT.

6.2 Understanding the Protection of Authorization Files

To display the protection codes for any file, use the DCL command DIRECTORY/PROTECTION.

Authorization files are created with the following default protections:

- User authorization file, SYSUAF.DAT

  The user authorization file, SYSUAF.DAT, is created with the following default protection:

  ```
  SYSUAF.DAT      S:RWE, O:RWE, G:RWE, W
  ```

- Proxy authorization files, NETPROXY.DAT and NET$PROXY.DAT

  Two proxy authorization files, NETPROXY.DAT and NET$PROXY.DAT, are created with the following default protections:

  ```
  NETPROXY.DAT    S:RWED, O:RWED, G:RWE, W
  NET$PROXY.DAT   S:RWED, O, G, W
  ```

  The primary proxy database that the system uses is the NET$PROXY.DAT file. NETPROXY.DAT is maintained:

  - For use by DECnet for OpenVMS
  - For backward compatibility

  See Section 6.9.3 for more details about network proxy accounts.

- Rights database file, RIGHTSLIST.DAT

VAX

On VAX systems, the RIGHTSLIST.DAT authorization file is created with the following default protection:

```
RIGHTSLIST.DAT  S:RWED, O:RWED, G:R, W  ♦
```

Alpha

On Alpha systems, the RIGHTSLIST.DAT authorization file is created with the following default protection:

```
RIGHTSLIST.DAT  S:RWED, O:RWED, G:RE, W  ♦
```

The procedures for adding a user account are discussed in detail in Section 6.6. Because the UAF is the prime repository for storing information about user accounts, it is important to understand its components before you add accounts.

6.3 Understanding 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, by dialing in to a system and observing the remote connect protocol, or by connecting via a LAT), and that terminal is not allocated by a user process, the system prompts for a name and password. The person using the terminal must enter a name and password combination that exists in a UAF record, or the system denies the user further access. If the name and password are accepted, the system performs the operations in Table 6–3.

Table 6–3 System Login Flow

Step	Action	Result
1.	System examines the login flags.	The system begins with DISUSER. If the DISUSER flag is set, the login attempt fails. Note that setting this flag for powerful, infrequently used accounts (such as Field Service accounts) eliminates the risk of guessed passwords for those accounts.
2.	System 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. Digital recommends that you limit nonbatch access of the SYSTEM account by using access times and day types. See Section 6.8.1 and Section 6.8.2.
3.	System passes control to the command interpreter.	The command interpreter is named in the user's UAF record; for example, DCL.

(continued on next page)

Table 6–3 (Cont.) System Login Flow

Step	Action	Result
4.	System checks whether SYS$LOGIN is defined.	If SYS$LOGIN is defined, the logical name is translated (in the case of DCL, to SYS$MANAGER:SYLOGIN.COM) and that procedure executes. If SYS$SYLOGIN is not defined, no system login is run. If a command procedure is specified in the LGICMD field and that procedure exists, it executes. Otherwise, if the LGICMD field is blank, the user's command file (named LOGIN.COM if the CLI is DCL), which is located in the SYS$LOGIN directory, executes automatically (if it exists). The system will not execute *both* a command procedure specified in the LGICMD field *and* a user's LOGIN file; if a procedure is specified in the LGICMD field, the system uses that procedure by default. You can, however, instruct the system to execute a user's LOGIN by calling it from within the procedure specified in LGICMD.

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 *OpenVMS DCL Dictionary*.) However, the system prohibits activities that violate the user's privilege allowance or exceed resource quotas.

6.4 Managing System-Supplied UAF Accounts

Typically, 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.) Allow access to this file only to those with SYSTEM privileges. See the AUTHORIZE section in the *OpenVMS System Management Utilities Reference Manual* for guidelines on protecting system files.

The UAF is accessed as a shared file. 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 you enter AUTHORIZE commands, not after the termination of AUTHORIZE. (For this reason, do not enter temporary values with the intent of fixing them later in the session.)

The Authorize utility (AUTHORIZE) provides a set of commands and qualifiers to assign values to any field in a UAF record. See the Authorize utility section in the *OpenVMS System Management Utilities Reference Manual* for complete information about UAF record fields and the commands and qualifiers used to assign attributes to these fields.

6.4.1 Understanding System-Supplied UAF Accounts

VAX

On VAX systems, the UAF on software distribution kits contains five accounts: DEFAULT, FIELD, SYSTEM, SYSTEST, and SYSTEST_CLIG. ♦

Alpha

On Alpha systems, DEFAULT and SYSTEM accounts are created for you. You can use SYS$MANAGER:CREATE_SPECIAL_ACCOUNTS.COM to create SYSTEST, SYSTEST_CLIG, and Field Service accounts, as explained in Section 6.4.2. ♦

Table 6–4 describes system-supplied UAF accounts.

Table 6–4 System-Supplied UAF Accounts

UAF Record	Description
DEFAULT	Serves as a template for creating new user accounts. A new user account is assigned the values of the DEFAULT account 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. You cannot rename or delete the DEFAULT account from the UAF. The following AUTHORIZE command creates a new account having the same values as the DEFAULT account, except that the password, UIC, and default directory fields are changed: `UAF> ADD MARCONI/PASSWORD=QLP6YT9A/UIC=[033,004]/DIRECTORY=[MARCONI]` Section 6.6 gives an example of how to use AUTHORIZE to add a user account. Section 6.7.4 explains how to create and use additional default templates.
FIELD	Permits Multivendor Customer Services personnel to test a new system. † On VAX systems, the default Field Service account has the user name FIELD. ‡ On Alpha systems, you name Field Service accounts for specific Field Service representatives; for example, Mary_Smith or John_Jones.
SYSTEM	Provides a means for you to log in with full privileges. You can modify the record for the system manager's account but you cannot rename it or delete it from the UAF. **Caution** Do not change the SYSTEM account UAF record fields for the default device, directory, and privileges. Installation of maintenance releases of the operating system and optional software products depends on certain values in these fields. Take note of any hourly or daily restrictions that you have implemented on the SYSTEM account when performing upgrades from the SYSTEM account.
SYSTEST	Provides an appropriate environment for running UETP, a User Environment Test Package, on standalone systems. (See Chapter 17.)
SYSTEST_CLIG	Provides an appropriate environment for running UETP in a VMScluster environment. SYSTEST_CLIG accounts have no passwords associated with them. (See Chapter 17.)

†VAX specific
‡Alpha specific

6.4.2 Creating Accounts on Alpha Systems (Alpha Only)

Alpha

On Alpha systems, you can use the SYS$MANAGER:CREATE_SPECIAL_ ACCOUNTS.COM command procedure to create SYSTEST, SYSTEST_CLIG, and multiple Field Service accounts.

6.4.2.1 Creating Field Service Accounts (Alpha Only)

On Alpha systems, you can use the CREATE_SPECIAL_ACCOUNTS.COM command procedure to create accounts for Field Service representatives. In creating these accounts, follow these guidelines:

- Create an account for each Field Service representative rather than just one account called FIELD for all the representatives. Note that you must use the command procedure for each account you want to create.
- Use account names that represent the actual names of Field Service representatives, not a generic name such as FIELD.

The following example shows typical dialogue with the CREATE_SPECIAL_ ACCOUNTS.COM command procedure when used to create a field service representative account:

```
$ @CREATE_SPECIAL_ACCOUNTS.COM

    This procedure creates accounts.

    Passwords may be needed for the following accounts:

        SYSTEST, Field Service

    Passwords must be a minimum of 8 characters in length.  All passwords
    will be checked and verified.  Any passwords that can be guessed easily
    will not be accepted.

* Do you want to create an account for SYSTEST (Y/N): N Return
* Do you want to create an account for SYSTEST_CLIG (Y/N): N Return
* Do you want to create an account for Field Service (Y/N): Y Return
* Enter username for Field Service account: john_jones Return
* Enter password for JOHN_JONES:
* Re-enter for verification:
$
```

Note that the system does not display the password or password verification that you enter.

Disabling Field Service Accounts (Alpha Only)

You can use the Authorize utility (AUTHORIZE) to disable Field Service representatives' accounts when these accounts are not in use and enable them again when they are needed.

To disable an account, use the AUTHORIZE command in the following format:

MODIFY username/FLAGS=DISUSER

For example:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> MODIFY JOHN_JONES/FLAGS=DISUSER
```

The login flag DISUSER disables the account and prevents anyone from logging in to the account.

Reenabling Field Service Accounts (Alpha Only)

To reenable the account when it is needed again, run AUTHORIZE and enter the command in the following format:

MODIFY username /FLAGS=NODISUSER

For example:

```
UAF> MODIFY JOHN_JONES/FLAGS=NODISUSER
```

6.4.2.2 Creating SYSTEST and SYSTEST_CLIG Accounts (Alpha Only)

The following example shows typical dialogue with the CREATE_SPECIAL_ACCOUNTS.COM command procedure when used to create SYSTEST and SYSTEST_CLIG accounts:

```
$ @CREATE_SPECIAL_ACCOUNTS.COM

    This procedure creates accounts.

    Passwords may be needed for the following accounts:

        SYSTEST, Field Service

    Passwords must be a minimum of 8 characters in length.  All passwords
    will be checked and verified.  Any passwords that can be guessed easily
    will not be accepted.


* Do you want to create an account for SYSTEST (Y/N): Y
* Enter password for SYSTEST:
* Re-enter for verification:

* Do you want to create an account for SYSTEST_CLIG (Y/N): Y

    The SYSTEST_CLIG account will be disabled.  You must reenable
    it (/FLAGS=NODISUSER) before running UETP but do not assign a password.

* Do you want to create an account for FIELD_SERVICE (Y/N): N

$
```

Enabling SYSTEST_CLIG Accounts (Alpha Only)

Although you create a SYSTEST_CLIG account using CREATE_SPECIAL_ACCOUNTS, the account is disabled. Enable the account using the /FLAGS=NODISUSER command; for example:

```
UAF> MODIFY SYSTEST_CLIG/FLAGS=NODISUSER
```

Disabling SYSTEST_CLIG Accounts (Alpha Only)

To disable a SYSTEST_CLIG account again, use the /FLAGS=DISUSER command; for example:

```
UAF> MODIFY SYSTEST_CLIG/FLAGS=DISUSER ♦
```

6.4.3 Maintaining System-Supplied Accounts (VAX Only)

VAX

Immediately after installing a VAX system, make the following changes in the UAF:

1. Disable the FIELD and SYSTEST accounts. Also, disable any infrequently used accounts.

   To disable an account, use the AUTHORIZE command in the following format:

   MODIFY username/FLAGS=DISUSER

   For example:

   ```
   $ RUN SYS$SYSTEM:AUTHORIZE
   UAF> MODIFY WOLF/FLAGS=DISUSER
   ```

   The login flag DISUSER disables the account and prevents anyone from logging in to the account.

To enable the account when it is needed, run AUTHORIZE and enter the command in the following format:

MODIFY username /FLAGS=NODISUSER

2. Optionally, change fields in the DEFAULT account. For example:

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

In this example, 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. ♦

6.4.4 Using the SYSTEM Account

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 exercise caution when you use it. For example, because you have BYPASS privilege, the system allows you to delete any file, no matter what its protection. If you enter an incorrect name or spurious asterisk, you might destroy files that you or other users need to keep. Consider using an account with fewer privileges for daily system management activities.

If you decide not to use the SYSTEM account for daily system management activities, you can still receive mail from the SYSTEM account. To do this, log in to the SYSTEM account, invoke MAIL, and use the SET FORWARD command in the following format to forward the mail to another account:

SET FORWARD node::username

For example:

```
$ MAIL
MAIL> SET FORWARD WINSTON::WOLF
MAIL> EXIT
```

Caution

Do not use DISUSER for user name SYSTEM if SYSTARTUP_VMS.COM submits batch jobs. Disable all access except Batch in these cases.

Also, be careful not to disable all of your privileged system accounts. If you inadvertently do so, you can recover by setting the system parameter UAFALTERNATE during a conversational boot operation. See Chapter 4 for information on emergency startup procedures.

6.4.5 Using AUTHORIZE to Maintain UAF Accounts

Using the Authorize (AUTHORIZE) utility, you create and maintain UAF accounts by assigning values to various **fields** within each account record. The values you assign do the following:

- Identify the user
- Define the user's work environment
- Control use of system resources

How to Perform This Task

1. Set your default to the directory that contains the SYSUAF.DAT file, typically, SYS$SYSTEM.
2. Gain access to a specific user record by running AUTHORIZE.
3. Enter the SHOW command (see example) to display a specific user record.
4. Enter AUTHORIZE commands such as ADD and MODIFY to create or change the information in the fields of the UAF record.

See Section 6.11 for a list of privileges, limits, and quotas that you can specify in the resource control and privileges fields of the UAF record.

Example

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> SHOW WELCH
```

The following example shows a typical user record for a restricted user account. Callouts describe the fields.

```
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   000000000011111111112222  Secondary 000000000011111111112222
Day Hours 012345678901234567890123  Day Hours 012345678901234567890123
Network:  ------ No access -------             ----- Full access ------
Batch:    ########--------#######               ---------#########------
Local:    ########--------#######               ---------#########------
Dialup:   ----- Full access ------              ------ No access -------
Remote:   ----- Full access ------              ------ No access -------
Expiration:            (none)    Pwdminimum:  6   Login Fails:     0

Pwdlifetime:            30       Pwdchange:   15-APR-1995 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:
  TMPMBX NETMBX ❺
Default Privileges:
  TMPMBX NETMBX
Identifier ❻                    Value             Attributes ❼
  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 DIGITAL Command Language (DCL) by default.

- Name of the command procedure to be executed automatically at login time. If the field is blank, the system uses the default CLI (DCL), and executes SYS$LOGIN:LOGIN.COM 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 system resources such as physical memory and CPU time.
- 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** list the ACL identifiers that the user holds and that are recorded in the rights database file.

❼ **Attributes fields** list the characteristics specified when adding identifiers to the rights database or when granting identifiers to users.

6.5 Preparing to Add User Accounts

This section describes what to do before adding a user account.

6.5.1 Choosing an Account Type

How you set up a user account depends on the needs of the individual user. Table 6–5 lists the account types and their characteristics.

Table 6–5 Account Types

Account Type	Characteristics
Interactive	This account has access to the system software. Work of a general nature, such as program development or text editing, is performed in this account. Usually, such an account is considered an individual account.

(continued on next page)

Table 6–5 (Cont.) Account Types

Account Type	Characteristics
Limited Access	This account provides controlled login to the system and, in some cases, has only a subset of user software available. Limited-access accounts ensure that the system login command procedure (SYLOGIN.COM) and the process login command procedure (specified by the /LGICMD qualifier in the UAF), as well as any command procedures they call, are executed. (See the *OpenVMS Guide to System Security* for information about writing limited access account command procedures.) The two types of limited accounts are: restricted and captive.
	Restricted — Used for network objects like MAIL, for network proxy accounts, or for implementing user authentication systems like smart cards.
	Captive — Limited by function; that is, only those who perform a particular function can use it (for example, an inventory system). Anyone whose job entails inventory control can access your system, but that person cannot access other subsystems or the base software. You might also use a captive account to run batch operations during unsupervised periods or to run applications programs with information you want to keep private.

6.5.2 Performing Additional Tasks

When adding a user account, you must 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. You can do this 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 format:

   ```
   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 Section 6.6.

6.5.2.1 Selecting a 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 or restricted account for Robert Jones might have a user name of JONES, while a captive account for an inventory system might be called INV103289, which gives some indication of the function of the account but is not easy to guess. 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 password that is not easy to guess. The user will be forced to change the password at first login. Only the person using the account should know the password. Encourage all users to set obscure passwords of at least eight characters and to change them frequently, or force the use of generated passwords with the /FLAGS=GENPWD and /GENERATE_PASSWORD qualifiers.

You can use the /PWDMINIMUM and /PWDLIFETIME qualifiers with the AUTHORIZE command ADD or MODIFY to enforce timely password modifications. The following table lists the qualifiers and specific action.

Qualifier	Action
/PWDMINIMUM	Specifies the minimum password length in characters (default is 6).
/PWDLIFETIME	Specifies a delta-time value. One week before that date, the system issues a warning message to the user. On that date, the password expires if it has not been changed.
/GENERATE_ PASSWORD	Invokes a password generator to generate user passwords.
/FLAGS=GENPWD	Allows you to force use of the automatic password generator when a user changes a password. Consider using the password generator for privileged accounts or whenever a user has access to sensitive data.

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).

Prohibit users from changing the passwords of captive accounts. To do this, specify /FLAGS=LOCKPWD when you create the captive account. 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=*new_password* at the UAF> prompt.

See the *OpenVMS System Management Utilities Reference Manual* for more information about AUTHORIZE.

6.5.2.2 Assigning the User Identification Code (UIC)

Assign each account a unique user identification code (UIC). A UIC has two formats: alphanumeric and numeric.

The alphanumeric UIC consists of a member name and, optionally, a group name separated by a comma and enclosed within brackets (for example, [DOCO,PRICE]). These identifiers might also appear as numeric characters consisting of a group identifier and a member identifier in octal (for example, [11,200]).

Assign accounts the same group number if their owners perform similar work, access the same files frequently, or use many of the same logical names. See the *OpenVMS Guide to System Security* for a detailed discussion of the user identification code.

Note

Digital reserves UIC group 1 and groups 300–377.

6.5.2.3 Adding a Disk Quota Entry

Disk quotas limit the amount of disk space available to individual users on a particular volume. 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 as follows:

1. Invoke SYSMAN.
2. Define the management environment to be node LARRY.
3. 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.
4. Exit from the utility.

Example

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

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 SYSMAN and establishing disk quotas, see the *OpenVMS System Management Utilities Reference Manual*.

6.5.2.4 Setting the User Default Device for an Interactive Account

For each interactive account, 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.

Example

If you 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.

6.5.2.5 Setting the User Default Device for a Captive Account

For a captive account, whether you create a top-level directory depends on the nature of the user system. If people use files in a particular directory, 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, make the default device specification DISK$DATA: and make the default directory specification [INV].

6.5.3 Understanding Account Security

The level of security that you establish for an account depends on the purpose of the account and whether it is shared with other users or groups. For an interactive user account, the default UIC-based protection is usually adequate.

Protecting Users' Files

The default protection for top-level directories is no world access. However, for new user directories, you might want to change the default to *world execute* access so that users will not have to change directory protection to allow other users read access to files in that directory.

Users can further protect their files and subdirectories on an individual basis with the DCL command SET SECURITY.

Using Access Control Lists (ACLs)

In some cases, such as project accounts, you might 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 offer a way to grant or deny users access to any security-relevant object.

Section 6.9.2 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 *OpenVMS Guide to System Security* and the *OpenVMS System Management Utilities Reference Manual*.

Using AUTHORIZE to Maintain the Rights Database

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 database for the identifiers that the user holds. You use the Authorize utility (AUTHORIZE) to maintain the rights database by adding or deleting identifiers as needed.

By allowing a group of users to hold common identifiers, you can create a group protection scheme that is more intricate than that provided by the UIC-based protection.

Using Protected Subsystems

Protected subsystems provide conditional access to data. In a protected subsystem, an application protected by normal access controls serves as a gatekeeper to objects belonging to the subsystem. While users are running the application, their process rights list contains identifiers giving them access to objects owned by the subsystem. As soon as they exit from the application, these identifiers and, therefore, the users' access rights to objects are taken away. For more information, see the *OpenVMS Guide to System Security*.

6.6 Adding User Accounts

The following sections explain how to use two different methods for adding user accounts:

- The Authorize utility (AUTHORIZE)
- A command procedure

6.6.1 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 (AUTHORIZE) to create the account.

How to Perform This Task

1. Give yourself the SYSPRV privilege:

```
$ SET PROCESS/PRIVILEGE=SYSPRV
```

2. Enter the following commands to set your default device and directory to SYS$SYSTEM and invoke AUTHORIZE:

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

3. When the Authorize 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
```

Choosing Qualifiers

This section lists the qualifiers that you can use when setting up an account with AUTHORIZE. Table 6–6 lists the qualifiers under the account attribute that they affect. See Section 6.11.2 for a detailed description of each qualifier. For a complete list of Authorize qualifiers, see the *OpenVMS System Management Utilities Reference Manual: A–L*.

Table 6–6 Qualifiers Used with AUTHORIZE

Limits and Quotas[1]		
/ASTLM	/FILLM	/PRCLM
/BIOLM	/JTQUOTA	/TQELM
/BYTLM	/MAXACCTJOBS	/WSDEFAULT
/CPUTIME	/MAXDETACH	/WSEXTENT
/DIOLM	/MAXJOBS	/WSQUOTA
/ENQLM	/PGFLQUOTA	
Priority[2]		
/PRIORITY		
Privileges		
/DEFPRIVILEGES	/PRIVILEGES	

[1]Default values are adequate in most cases.

[2]Default values are usually adequate for accounts not running real-time processes.

(continued on next page)

Table 6–6 (Cont.) Qualifiers Used with AUTHORIZE

Login Access Controls [3]		
/ACCESS	[4]/FLAGS	/PRIMEDAYS
/DIALUP	/INTERACTIVE	/REMOTE
/EXPIRATION	/LOCAL	

[3]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 6.8 for a discussion of using these fields to restrict login times and functions.

[4]Not all FLAGS fit into this category.

6.6.2 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.

You can modify ADDUSER.COM as appropriate for the needs of your system. 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
Owner	No default; must supply
Password	User name specified
UIC group number	200
UIC member number	No default; must supply number
Account name	Optional
Privileges	TMPMBX,NETMBX
Login directory	User name specified
Login device	$DISK1
Disk quota	1000
Overdraft quota	100

The UIC must be unique for the system. For example, each account in the UIC group 200 must have a unique member number. You can list the UICs currently assigned to users by entering a question mark (?) after the UIC member number prompt. The account is not 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 enter NO, the account is removed.

Note

If you press Ctrl/Y before, during, or directly after the system displays the characteristics of the account (that is, before you respond to the ''satisfactory?'' prompt), the account, or portions of it, will still be added.

Make sure users log in to their accounts promptly to change the password.

6.7 Maintaining User Accounts

As system manager, you perform a certain number of user account maintenance tasks, such as modifying and deleting accounts. The following sections explain how to perform these tasks:

Task	Section
Using command procedures for interactive accounts	Section 6.7.1
Modifying a user account	Section 6.7.2
Listing user accounts	Section 6.7.3
Maintaining the user environment	Section 6.7.4
Deleting a user account	Section 6.7.5
Using BACKUP to remove user files	Section 6.7.6
Disabling a user account	Section 6.7.7

6.7.1 Using Command Procedures for Interactive Accounts

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

- Define symbols
- Assign logical names
- Display messages and the time of day
- Set terminal characteristics
- Define keys to perform certain functions
- Set process default file protection (SET PROTECTION/DEFAULT)

Login command procedures are useful for saving keystrokes and standardizing operations.

In establishing login command procedures for interactive accounts, you have the following choices:

Login Command Procedure	Description
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 can specify an additional login command procedure with the /LGICMD qualifier of the AUTHORIZE commands 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 system (and, optionally, individual) login command procedures are not implemented, the system looks for a command file called LOGIN.COM in the user's login directory as defined by the UAF (user authorization file) record device and directory fields. If the file is found, the system executes it. The user develops and maintains this command file, which should follow these conventions: • Device and directory names must take the default file specification for the account. • The file name and file type 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 (user identification code) of the file to that of the user. You make this change with the DCL command SET FILE/OWNER.

Example 6–1 and Example 6–2 illustrate typical systemwide and user-specified login command procedures.

Example 6–1 Sample Systemwide 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
$ !
$ !     Allow network jobs to start faster
$ !
$ IF F$MODE() .EQS. "NETWORK" THEN GOTO EXIT
$ !
$ !     Enable Ctrl/T handling by DCL
$ !
$ SET CONTROL=T
```

(continued on next page)

Example 6–1 (Cont.) Sample Systemwide SYS$MANAGER:SYLOGIN.COM Login Command Procedure

```
$ !
$ !     Define Foreign Commands For Installed Utilities
$ !
$ USERS             ==    "SHOW USERS"
$ DISPLAY           ==    "MONITOR PROCESSES/TOPCPU"
$ INFO              ==    "SHOW PROCESS/CONTINUOUS"
$ SUSPEND           ==    "SET PROCESS/SUSPEND"
$ RESUME            ==    "SET PROCESS/RESUME"
$ SETNAME           ==    "SET PROCESS/NAME"
$ !
$ !     Define a symbol indicating whether the terminal
$ !     is on a dialup port
$ !
$ TT == F$GETDVI("TT","DEVNAM")-"_"
$ DIALUP == ((TT .GES. "TTG0:" .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 6–1 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, add the DCL command that resets the Ctrl/Y function (SET CONTROL=Y).

Example 6–2 shows typical abbreviations and symbols that a user might define in a login file.

Example 6–2 Sample Login Command Procedure (LOGIN.COM) for a User Account

```
$ SET NOON
$ SET PROTECTION=(S=RD,O=RWED,G=R,W=R)/DEFAULT
$ !
$ ! Define abbreviations for often used commands
$ !
$ DIR*ECTORY    ==     DIRECTORY/DATE/SIZE
$ PU*RGE        ==     PURGE/LOG
$ DE*LETE       ==     DELETE/LOG/CONFIRM
$ !
```

(continued on next page)

Example 6–2 (Cont.) Sample Login Command Procedure (LOGIN.COM) for a User Account

```
$ !
$ ! 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"
$ SYS           ==     "SHOW SYSTEM"
$ DAY           ==     "SHOW TIME"
$ !
$ ! 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
```

Using 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.

How to Perform This Task

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.

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 add the following command, just above the last line:

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

Note that this technique works in some situations but it is not foolproof; there are many alternative ways to terminate a process.

6.7.2 Modifying a User Account

To change a user account's quotas, default directory, password, authorized privileges, or any other characteristics assigned by AUTHORIZE, use the MODIFY command. You can use the MODIFY command to change any of the fields in an existing user account. However, a user must log out and log in again for the modifications to take effect.

Examples

When a user forgets a password and cannot log in, use the MODIFY/GENERATE command in AUTHORIZE to reset a user password. For example, the following command generates a new password for user WELCH:

```
UAF> MODIFY WELCH/GENERATE
```

By default, after logging in, user WELCH must change the password.

Any changes that you make to a user's record will take effect *after* the user next logs in. For example, suppose that user JONES currently has an open file quota (FILLM) of 20. To increase user Jones' open file limit to 40, you would use the following command in AUTHORIZE:

```
UAF> MODIFY JONES/FILLM=40
```

Any process of user JONES that is logged in at the time that you modify the user authorization file continues to have a file limit of 20. In order to have an open file limit of 40, user JONES must log out and then log in again, after you have made the modification to the user authorization file using AUTHORIZE.

6.7.3 Listing User Accounts

Use the AUTHORIZE command LIST to create the file SYSUAF.LIS, containing a summary of all user records in the UAF. By default, the LIST command produces a brief report containing the following information from the UAF:

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

Use the /FULL qualifier to create a full report of all the information (except user passwords) contained within the UAF.

Examples

1. The following example writes a brief report of the UAF to the output file SYSUAF.LIS:

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

2. The following example lists a full report of the UAF, including identifiers held by the users:

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

6.7.4 Maintaining the User Environment

As the work requirements of your system change, you might have 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.

Creating Additional Default Record Templates

On systems where all users perform the same type of work, you typically use the system-supplied default record, DEFAULT, as the template for adding new user records. You might 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 decide the following:

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

How to Perform This Task

Once you define a user category and establish which record attributes are needed, you can create the default record.

Examples

1. 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 uses the specified default record except where you explicitly override field values.

2. You could use the following command to create for a new user a record 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.

6.7.5 Deleting a User Account

The main problem in deleting an account, especially an interactive or restricted account, is deleting the files used by the account.

How to Perform This Task

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 save 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 might 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 6–3 illustrates a command procedure that deletes an account's files from the bottom level up.
4. Exit from the user account and return to a privileged account. Remove the user's account, using the Authorize utility (AUTHORIZE).

   When you run AUTHORIZE to remove a user's UAF record, AUTHORIZE also removes the user's connections as a holder of identifiers in the rights database. However, if a departed user is the only remaining holder of a given identifier, remove that identifier to avoid future confusion. See the *OpenVMS Guide to System Security*.
5. Remove the user's disk quota entry from the disk quota file, if one existed, with SYSMAN.

6. Remove associated mail information by entering the MAIL command REMOVE *username*. (See the *OpenVMS User's Manual* for more information.)

Example 6–3 is a command procedure that deletes an account's files.

Note

Do not execute this command procedure from a privileged account.

Example 6–3 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'
```

6.7.6 Using BACKUP to Remove User Files

If each user has a unique UIC, you can use the Backup utility (BACKUP) to remove the user's files, even if the files are scattered throughout the directory structure. See the Backup utility section in the *OpenVMS System Management Utilities Reference Manual: A–L* for more information.

Examples

1. The following is an example of a BACKUP command used to remove files:

   ```
   $ BACKUP/DELETE PUBLIC:[...]/BY_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.SAV on device MTA0. Note that the BACKUP/DELETE command does not delete the directory files (file type .DIR) for the account.

2. To recover lost files, enter the ANALYZE/DISK_STRUCTURE command in the following format:

   ANALYZE/DISK_STRUCTURE/REPAIR/CONFIRM device-name:

   See Section 8.12.3 for a complete description of how to recover lost files. See the *OpenVMS System Management Utilities Reference Manual* for information on using the Analyze/Disk_Structure utility.

6.7.7 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.

6.8 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 might want to control when certain users are allowed to log in. Using the Authorize utility (AUTHORIZE), 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 following sections describe how to perform these tasks:

Task	Section
Setting day types	Section 6.8.1
Restricting logins to specific times	Section 6.8.2
Restricting login functions	Section 6.8.3
Using login command procedures for restricted or captive accounts	Section 6.8.4
Setting priorities for user processes	Section 6.8.5

For a detailed description of the qualifiers used to restrict the use of accounts, see the Authorize utility section in the *OpenVMS System Management Utilities Reference Manual*.

6.8.1 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 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.

You can assign two types of login restrictions to either day type:

Restriction	Description
Time restrictions	Limits logins to specific hours of the day
Function restrictions	Limit types of login

The default user record defines the five weekdays (Monday through Friday) as PRIMARY days, and the two weekend days (Saturday and Sunday) 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.

6.8.2 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 from dialup terminals
/[NO]INTERACTIVE	Specifies access hours for interactive logins from any terminal
/[NO]LOCAL	Specifies access hours for interactive logins from local terminals
/[NO]REMOTE	Specifies access hours for interactive logins from network remote terminals (SET HOST)

Users still logged in when the access time has expired receive the following warning message and have 2 minutes to log out before their processes are terminated by the job controller:

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

6.8.3 Restricting Login Functions

In addition to specifying hourly login restrictions, you can 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.
[NO]AUTOLOGIN	[Do not] prevent access except by automatic login when automatic logins are enabled.
[NO]CAPTIVE	[Do not] prevent user from changing any defaults at login (implies DISCTLY).
	[Do not] deny user access to the DCL command level.
[NO]DEFCLI	[Do not] prevent user from changing default CLI or CLI tables.
[NO]DISCTLY	[Do not] disable Ctrl/Y interrupts.

Keyword	Meaning
[NO]DISFORCE_PWD_CHANGE	[Do not] remove requirement that user must change an expired password at login.
[NO]DISIMAGE	[Do not] prevent user from using the RUN or MCR commands or from executing ‘‘foreign’’ commands.
[NO]DISMAIL	[Do not] prevent mail delivery to the user.
[NO]DISNEWMAIL	[Do not] suppress ‘‘New Mail . . . ’’ announcements.
[NO]DISPWDDIC	[Do not] disable automatic screening of new passwords against a system dictionary.
[NO]DISPWDHIS	[Do not] disable automatic checking of new passwords against list of user’s old passwords.
[NO]DISREPORT	[Do not] disable reporting of login information (last login date, login failures, and so on).
[NO]DISUSER	[Do not] disable 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]PWD_EXPIRED	[Do not] mark password as expired.
[NO]PWD2_EXPIRED	[Do not] mark second password as expired.
[NO]RESTRICTED	[Do not] prevent user from changing any defaults at login.

6.8.4 Using Login Command Procedures for Restricted or Captive Accounts

Using the /LGICMD qualifier with the AUTHORIZE commands ADD, MODIFY, or COPY defines the login procedure for a restricted or captive account. A person logging in to such an account cannot modify the procedure with any of the login qualifiers: /CLI, /DISK, /COMMAND, /NOCOMMAND, /TABLES.

The CAPTIVE and RESTRICTED flags do the following:

- Disable the use of Ctrl/Y (/FLAG=DISCTLY); however, a system manager can enable the Ctrl/Y sequence for a restricted account by adding the DCL command SET CONTROL=Y at the end of the login procedure.
- Prevent the use of the SPAWN command from MAIL or the use of the SPAWN built-in procedure from the DEC Text Processing Utility (DECTPU).

Once logged in, a person using a restricted account operates from the DCL level and can access any available software.

A person using a captive account is locked into the application software where access to the DCL level is denied, provided the system manager observes the following practices:

- When user input is required (and it is best to avoid it), *never* execute the input directly, but rather compare it to what is expected and screen for illegal characters, such as the ampersand (&), apostrophe (’), at sign (@), dollar sign ($), hyphen (-), and quotation mark (").
- Use the DCL command READ/PROMPT in captive account login command procedures because the INQUIRE command is not allowed.
- Set the subprocess limit to 0 to prevent the user from spawning out of the account. Set both the /PRCLM qualifier and the SYSGEN parameter PQL_MPRCLM.

Example

A simple login command procedure for a captive account used for an inventory system might consist of the following commands:

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

The application program INVENTORY assumes control when the user logs in to the account. Assign the CAPTIVE flag to the login flags field of the captive account UAF record by specifying the AUTHORIZE qualifier /FLAGS=CAPTIVE. Section 6.7.4 shows how to use AUTHORIZE to create a UAF record for a captive account.

Example 6–4 shows a command procedure for a highly secure captive account, which restricts the user to a very limited set of commands. System managers must be sure to deny the account owner any write access to the login command procedure and its directory. Note also that the security manager would use the AUTHORIZE qualifier /NOINTERACTIVE when establishing this account.

For more information about captive and restricted accounts, see the *OpenVMS Guide to System Security*.

Example 6–4 Sample Captive Command Procedure

```
$ deassign sys$input
$ previous_sysinput == f$logical("SYS$INPUT")
$ on error then goto next_command
$ on control_y then goto next_command
$ set control=(y,t)
$
$next_command:
$ on error then goto next_command
$ on control_y then goto next_command
$
$ if previous_sysinput .nes. f$logical("SYS$INPUT") then deassign sys$input
$ read/end=next_command/prompt="$ " sys$command command
$ command == f$edit(command,"UPCASE,TRIM,COMPRESS")
$ if f$length(command) .eq. 0 then goto next_command
$
$ delete = "delete"
$ delete/symbol/local/all
$ if f$locate("@",command) .ne. f$length(command) then goto illegal_command
$ if f$locate("=",command) .ne. f$length(command) then goto illegal_command
$ if f$locate("F$",command) .ne. f$length(command) then goto illegal_command
$ verb = f$element(0," ",command)
$
$ if verb .EQS. "LOGOUT" then goto do_logout
$ if verb .EQS. "HELP" then goto do_help
$
$ write sys$output "%CAPTIVE-W-IVVERB, unrecognized command \",verb,"\"
$ goto next_command
```

(continued on next page)

Example 6–4 (Cont.) Sample Captive Command Procedure

```
$
$illegal_command:
$ write sys$output "%CAPTIVE-W-ILLEGAL, bad characters in command line"
$ goto next_command
$
$do_logout:
$ logout
$ goto next_command
$
$do_help:
$ define sys$input sys$command
$ help
$ goto next_command
```

6.8.5 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.

VAX

On VAX systems, priorities range in value from a low of 0 to a high of 31; 0 through 15 are timesharing priorities; 16 through 31 are real-time priorities. ♦

Alpha

On Alpha systems, priorities range in value from a low of 0 to a high of 63; 0 through 15 are timesharing priorities; 16 through 63 are real-time priorities. ♦

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 0. 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 results, because the system reacts sharply to even small base priority differences.

6.9 Setting Up Special Accounts

As system manager, you might need to set up a variety of special accounts, such as automatic login accounts, project accounts, and proxy accounts. The following sections explain how to perform these tasks:

Task	Section
Setting up an automatic login account with SYSMAN	Section 6.9.1
Setting up a project account with ACL identifiers	Section 6.9.2
Creating network proxy authorization files	Section 6.9.4
Adding proxy accounts	Section 6.9.5
Removing proxy accounts	Section 6.9.6
Displaying proxy accounts	Section 6.9.7
Controlling proxy logins	Section 6.9.8

Section 6.9.3 explains what network proxy accounts are.

6.9.1 Setting Up an Automatic Login Account with SYSMAN

The System Management utility (SYSMAN) includes the functions of the automatic login facility (ALF). Using SYSMAN ALF commands, you can set up a terminal that automatically logs in a user to a certain user name. For example, a terminal might be set up for the account INVENTORY, which automatically logs in a user to a captive account when the user presses the Return key.

First, you must follow the steps described in the previous sections to create the top-level default directory and to add the account. Then you can associate the account with a particular terminal or port using the following format:

ALF ADD device username [/TERMINAL] [/PORT] [/PROXY] [/LOG]

where:

device	is the terminal or port name that you want to assign to a user name. Note that *device* must be a terminal name if you do not specify qualifiers on the command line.
username	is the account user name that you want to assign to a particular terminal or port.
/TERMINAL	causes SYSMAN to treat *device* as a terminal name. This is the default behavior.
/PORT	causes SYSMAN to treat *device* as a port name. If the port name contains a special character such as a slash (/) or if it contains lowercase letters that you want to preserve, you must enclose the port name within quotes. Be aware that anything within quotation marks is written literally to the ALF database file. For example, if the actual port name contains uppercase letters as well as special characters, be sure to specify uppercase letters within the quotation marks. Otherwise, a mismatch will occur between the actual port name and what is specified in the SYSALF.DAT file.
/PROXY	causes SYSMAN to check that *device* is in the NODE::USERNAME format, which can be up to 63 characters in length.
/LOG	causes SYSMAN to display a message that the *device* and *username* have been added to the ALF database.

To protect automatic login accounts, set the AUTOLOGIN flag in the account's UAF record. This flag makes the account available only by autologin, batch, and network proxy.

Examples

The following example shows how to invoke SYSMAN and assign terminal TTA0 to the INVENTORY25 account:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> ALF ADD TTA0 INVENTORY25
```

When you create ALF records for proxy accounts, the device parameter can be as long as 63 characters. For example:

```
SYSMAN> ALF ADD VMS:.ZKO.VMSORG.SYSMAN.CLIENT1::SYSTEM FOOBAR
```

In this command, VMS:.ZKO.VMSORG.SYSMAN.CLIENT1::SYSTEM is the value of the device parameter.

For more information about autologin accounts and the SYSMAN ALF commands, see the *OpenVMS System Management Utilities Reference Manual* and the *OpenVMS Guide to System Security*.

6.9.2 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 *OpenVMS Guide to System Security* for complete details on setting up accounts with ACLs.

How to Perform This Task

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

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.

Example

The following example 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 Section 6.5 and Section 6.6 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}
UAF> EXIT
```

3. Create the disk quota authorization for the project identifier. For example, the following command invokes SYSMAN 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
SYSMAN> EXIT
```

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 (access control list entry) as a default:

```
$ SET SECURITY [000000]KITE_FLYING.DIR;1 -
_$ /ACL=((DEFAULT_PROTECTION,S:RWED,O:RWED,G,W) -
_$ (IDENTIFIER=KITE_FLYING, ACCESS=READ+WRITE+EXECUTE), -
_$ (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, write, and execute 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 appears only 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)
```

You can use the Creator ACE command in the ACL editor to add an extra ACE to the ACL for a file created within the directory to which you assign the Creator ACE. The Creator ACE applies only when the following conditions exist:

- The file being created is not owned by the user identification code (UIC) of the process creating the file.
- The process creating the file does not have system privileges.

See the *OpenVMS System Management Utilities Reference Manual* and the *OpenVMS Guide to System Security* for more information about the Creator ACE.

6.9.3 Understanding Network Proxy Accounts

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. Very often, system managers set up proxy accounts as restricted. 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.

To properly maintain proxy database consistency across all members of a mixed-version VMScluster system, you must perform all proxy modifications from either of the following:

- A VAX Version 6.1 (or later) system
- An Alpha Version 6.2 system

This restriction ensures that the NET$PROXY.DAT database is updated with correct proxy information.

For more information about proxy accounts, see the *OpenVMS Guide to System Security*.

6.9.4 Creating Network Proxy Authorization Files

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 (AUTHORIZE). See the *OpenVMS System Management Utilities Reference Manual* for more information about the Authorize utility.

On OpenVMS systems, the following information is stored in two proxy authorization files, NETPROXY.DAT and NET$PROXY.DAT:

- On systems running DECnet for OpenVMS, both NETPROXY.DAT and NET$PROXY.DAT contain proxy information stored using DECnet for OpenVMS VAX node names.
- On systems running DECnet/OSI:
  - NET$PROXY.DAT contains proxy information stored using DECnet/OSI full names.
  - NETPROXY.DAT contains proxy information stored using DECnet/OSI node synonyms.

Note

Deleting either NETPROXY.DAT or NET$PROXY.DAT will result in incorrect functioning of proxy access on systems running DECnet Phase IV. Also, many applications such as ALL–IN–1 rely on NETPROXY.DAT.

The SYSUAF.DAT file, on your local system, must associate proxy accounts with user accounts. 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_MKT 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_MKT 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.

How to Perform This Task

Use the AUTHORIZE command CREATE/PROXY to create and initialize network proxy authorization files:

```
UAF> CREATE/PROXY
```

6.9.5 Adding Proxy Accounts

You create a proxy account by adding entries to the network proxy authorization file; these entries equate one or more users on a remote node to users on your home node.

How to Perform This Task

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 total local accounts: 1 default proxy account and 15 alternate proxy accounts. Use the /DEFAULT qualifier to specify the default proxy account.

Examples

1. The following command adds a user proxy account:

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

   Assume that you have created three accounts named REMOTE_MKT, 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_MKT account on your home node. WALTER can access any data from his node that REMOTE_MKT 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.

   Caution

   Because the remote user receives the same privileges as the local user, do 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.

2. You can specify remote users 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
   ```

3. A number of your users might 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 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. On systems running DECnet/OSI, the system adds network proxies to the network proxy authorization file NET$PROXY.DAT using DECnet/OSI full names. For example:

```
UAF> ADD/PROXY RUBY::DELAPORT DELAPORT/DEFAULT,SYSTEM
%UAF-I-NAFADDMSG, proxy from OMNI:.US.RUBY::DELAPORT to DELAPORT added
%UAF-I-NAFADDMSG, proxy from OMNI:.US.RUBY::DELAPORT to SYSTEM added
```

This example adds proxy access from RUBY::DELAPORT to the local accounts DELAPORT (the default) and SYSTEM.

The system additionally stores node synonyms in NETPROXY.DAT for use by DECnet for OpenVMS and for backward compatibility for layered products.

6.9.6 Removing Proxy Accounts

To remove proxy accounts, you use the AUTHORIZE command REMOVE/PROXY; for example:

```
UAF> REMOVE/PROXY RUBY::DELAPORT SYSTEM
```

This command removes proxy access from RUBY::DELAPORT to the local SYSTEM account.

6.9.7 Displaying Proxy Accounts

You can display proxy accounts by using the AUTHORIZE command SHOW /PROXY. This command displays only the first 255 characters of the node name, although the command can handle a maximum of 1024 characters.

The following example assumes that proxy access from RUBY::DELAPORT to the local account DELAPORT has been added to the network proxy authorization file as the default.

Examples

1. On systems running DECnet for OpenVMS, the system displays the following type of information:

```
UAF> SHOW/PROXY *::*

RUBY::DELAPORT
     DELAPORT (D)
```

(D) indicates that DELAPORT is the default proxy.

2. On systems running DECnet/OSI, the system displays information in full names format:

```
UAF> SHOW/PROXY *::*

OMNI:.US.RUBY::DELAPORT
    DELAPORT (D)
```

(D) indicates that DELAPORT is the default proxy.

You can display information in the old format NETPROXY.DAT database by using the /OLD qualifier with the SHOW/PROXY command; for example:

```
UAF>  SHOW/PROXY/OLD *::*
```

6.9.8 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.)

On systems running DECnet for OpenVMS, you can change proxy access or disable it totally by using the Network Control Program (NCP). The *DECnet for OpenVMS Networking Manual* describes how to use NCP to control proxy access.

On systems running DECnet/OSI for OpenVMS software, you can change proxy access by using the Network Control Language utility (NCL). The *DECnet/OSI Network Control Language Reference* manual describes how to use NCL.

6.10 Managing MAIL

When managing user accounts, you might have to manage a user's mail account. For example, you may want to perform the following tasks:

- Change a user's MAIL profile; for example, change a user name or specify a default print queue for the account
- Set a forwarding address for a user who has moved to another location
- Disable a user's account from receiving mail
- Customize a user's MAIL environment by setting flags in a user's UAF record

The following sections explain how to perform all of these tasks.

6.10.1 Modifying a User Record

All users can modify and display information about their own records using various SET and SHOW commands available within MAIL. These commands access SYS$SYSTEM:VMSMAIL_PROFILE.DATA, which is a single-key indexed sequential file containing information for each user.

With SYSPRV, you can modify the records of other users. Table 6–7 lists the fields in the user profile record and the MAIL command you can use to modify those fields.

Table 6–7 MAIL User Profile Record

Field	Command
Username	———
Forwarding address	SET FORWARD
Personal name	SET PERSONAL_NAME
Editor name	SET EDITOR
Copy SEND/REPLY flags	SET COPY_SELF
Autopurge flag	SET AUTO_PURGE
Mail file subdirectory name	SET MAIL_DIRECTORY
New mail count	READ/NEW
Default queue	SET QUEUE
Default form	SET FORM

(continued on next page)

Table 6–7 (Cont.) MAIL User Profile Record

Field	Command
Carbon copy enabled	SET CC_PROMPT

You can arrange mail forwarding for users without accounts on the system by using the command SET FORWARD/USER=*user*.

6.10.2 Removing a User Record

Typically, once you have deleted a user's record in the UAF, you will also want to remove that user's information from the user profile database. You can remove a record from the user profile database with the REMOVE command.

6.10.3 AUTHORIZE Flags and MAIL

Certain flags set in a user's UAF record can affect the user's MAIL environment. You can set the appropriate flags in a user account by specifying the following flags with the /FLAGS qualifiers using AUTHORIZE:

Flag	Meaning
[NO]DISNEWMAIL	Enables or disables the display of the new mail count when the user logs in to the system.
[NO]DISMAIL	Allows or restricts the receipt of new mail.

6.11 Managing System Resources

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

6.11.1 Understanding Pages and Pagelets

VAX and Alpha systems allocate and deallocate memory for processes in units called **pages**. A page on a VAX system is 512 bytes. On Alpha systems, the page size will be one of four values: 8 kilobytes (KB) (8192 bytes), 16KB, 32KB, or 64KB. A particular Alpha system will implement only one of the four page sizes and the initial set of machines use an 8 kilobyte page.

Alpha

In most cases, Alpha systems present to users, and accept from users, units of memory in a 512-byte quantity called a **pagelet**. Thus, one pagelet is the same size as one VAX page. Also, on an Alpha 8KB computer, 16 pagelets equal 1 Alpha page. The following conversion table shows the relationship between pages, pagelets, and bytes:

```
One Alpha pagelet = one VAX page      = 512 bytes
One Alpha page    = 16 Alpha pagelets = 16 VAX pages = 8192 bytes ♦
```

Authorize utility commands, parameters, and default values are identical. However, the default values for process quotas related to memory use might be inappropriate for some Alpha system users.

See *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for more information about Alpha system management.

6.11.2 Setting Limits on 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 create an account for the user with the Authorize utility.

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 has at any given time, the system uses four types of limits for sharing resources. Table 6–8 lists and describes these four types of limits.

Table 6–8 Types of Limits

Limit Type	Description
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.

When you create an account, you assign values to the limits shown in Table 6–9. These limits are described in the following sections. Usually, you simply assign the default values for these limits. However, see the *Guide to OpenVMS Performance Management* for a discussion of how to evaluate and adjust the limits in the context of performance optimization strategies. See also *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for more information about the differences between VAX and Alpha systems.

Table 6–9 summarizes each of these limits, the value supplied in the UAF record for the SYSTEM and DEFAULT accounts, and the type of limit.

Table 6–9 Limits and Suggested Values for SYSTEM and DEFAULT Accounts

Limit	VAX SYSTEM Value	VAX DEFAULT Value	Alpha SYSTEM Value	Alpha DEFAULT Value	Type[1]	Description
ASTlm	50	40	250	250	N	AST queue limit
BIOlm	40	40	150	150	N	Buffered I/O count limit
Bytlm	32768	32768	64000	64000	P	I/O byte count limit
CPU	0	0	0	0	D	CPU time limit (0 = no limit)
DIOlm	40	40	150	150	N	Direct I/O count limit

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

(continued on next page)

Table 6–9 (Cont.) Limits and Suggested Values for SYSTEM and DEFAULT Accounts

Limit	VAX SYSTEM Value	VAX DEFAULT Value	Alpha SYSTEM Value	Alpha DEFAULT Value	Type[1]	Description
Enqlm	300	200	2000	2000	P	Enqueue quota
Fillm	300	300	100	100	P	Open file limit
JTquota	4096	4096	4096	4096	P	Initial byte quota for jobwide logical name table
Maxacctjobs	0	0	0	0	S	Maximum active processes for a single account (0 = no limit)
Maxdetach	0	0	0	0	S	Maximum detached processes for a single user name (0 = no limit)
Maxjobs	0	0	0	0	S	Maximum active processes for a single user name (0 = no limit)
Pgflquo	20480 pages	32768 pages	50000 pagelets	50000 pagelets	P	Paging file limit
Prclm	10	2	10	8	P	Subprocess creation limit
TQElm	30	40	20	10	P	Timer queue entry limit
WSdef	256 pages	256 pages	2000 pagelets	2000 pagelets	N	Default working set size
WSextent	40960 pages	1024 pages	16384 pagelets	16384 pagelets	N	Working set extent
WSquo	512 pages	512 pages	4000 pagelets	4000 pagelets	N	Working set quota

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

Table 6–10 shows the names and descriptions of the SYSTEM and DEFAULT accounts whose values are listed in Table 6–9. The /EXPIRATION qualifier is also described in Table 6–10.

Table 6–10 Descriptions of SYSTEM and DEFAULT Accounts

Account	Description
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 all system services that accept an AST address as an argument, and 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.
Buffered I/O Count Limit (BIOlm)	Limits the number of outstanding buffered I/O operations permitted for 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.
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.
CPU Time Limit (CPU)	Limits the amount of CPU time that a user's process can use per session. 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 applies only to this instance or to other instances of the user's processes. CPU is not cumulative across separate sessions or batch jobs.
Direct I/O Count Limit (DIOlm)	Limits the number of outstanding direct I/O operations permitted to 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.

(continued on next page)

Table 6–10 (Cont.) Descriptions of SYSTEM and DEFAULT Accounts

Account	Description
Enqueue Quota (Enqlm)	Limits the number of locks a process (and its subprocesses) can own. OpenVMS Record Management Services (RMS) uses the Lock Management facility to synchronize shared file access, global buffers, and record locks. Because RMS removes one lock for every shared file, local buffer, global buffer section, and outstanding record lock, users who expect to perform large amounts of RMS file sharing should have Enqlm set to a large value. If your process performs extensive RMS file sharing without sufficient enqueue quota, you could receive the SS$_EXENQLM error message. Furthermore, if your system performs extensive RMS file sharing and the value of the LOCKIDTBL system parameter is too low, you could receive the SS$_NOLOCKID error message. Note that whenever you increase the value of LOCKIDTBL, you might have to increase the value of the RESHASHTBL system parameter. (See the *OpenVMS System Management Utilities Reference 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 a value other than the default value for the multibuffer counts, estimate the number of locks per process per file as 1 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.
Expiration Date and Time (EXPIRATION)	Qualifier that specifies the expiration date and time of the account. The /NOEXPIRATION qualifier removes the expiration date on the account or resets the expiration time for expired accounts. The /EXPIRATION qualifier does not affect the expiration of passwords.
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. Note that each open file also requires at least 96 bytes of Bytlm.
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.
Maximum Account Jobs Limit (Maxacctjobs)	Specifies the maximum number of batch, interactive, and detached processes that might be active at one time for all users of a single account. Maxacctjobs is a systemwide limit.
Maximum Detached Processes Limit (Maxdetach)	Specifies the maximum number of detached processes that might be active at one time for all users of a single account. Maxdetach is a systemwide limit.
Maximum Process Jobs Limit (Maxjobs)	Specifies the maximum number of interactive, batch, and detached processes that can be active at one time for the cited user name. Maxjobs is a systemwide limit.

(continued on next page)

Table 6–10 (Cont.) Descriptions of SYSTEM and DEFAULT Accounts

Account	Description
Paging File Limit (Pgflquota)	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. Pgflquota 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. Pgflquota is a pooled limit.
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.
Timer Queue Entry Limit (TQElm)	Limits either 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.
Default Working Set Size (WSDEF)	Sets the initial working set size limit for a user's process. WSDEF is a nondeductible limit. If the value specified exceeds the value of WSquo, the lesser value is used.
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 operating system in response to heavy page faulting by the user. WSextent is a nondeductible quota. This value should always be greater than or equal to WSQUO. The value is controlled by the system parameter WSMAX. Note that PQL_MWSEXTENT will overwrite the account's value for WSextent if PQL_MWSEXTENT is larger than WSextent.
Working Set Quota (WSQUO)	Specifies the working set quota. This is the maximum amount of physical memory a user process can lock into its working set. It also represents the maximum amount of swap space that the system reserves for this process and the maximum amount of physical memory that the system allows the process to consume if the systemwide memory demand is significant. This parameter guarantees the user that the number of physical pages specified will be available. The maximum value of WSQUO is 64K pages. WSQUO is a nondeductible quota. This value should be greater than or equal to WSDEF. The value is capped by the system parameter WSMAX.

7
Managing Peripheral Devices

System managers are often responsible for setting up and managing peripheral devices such as terminals and printers. This chapter describes these tasks. For information on managing storage media such as disks and tapes, see Chapter 8.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Getting information about devices on the system	Section 7.2
Setting security protection characteristics on devices	Section 7.3
Connecting devices and loading device drivers	Section 7.4
Managing terminals	Section 7.5
Managing printers	Section 7.6
Managing tape drives	Section 7.7
†Managing card readers	Section 7.8

†VAX specific

This chapter explains the following concepts:

Concept	Section
Device names	Section 7.1
Spooled printers	Section 7.6.2

7.1 Understanding Device Names

On some systems, device names follow the format *ddcu*, where *dd* is the device code, *c* is the controller designation, and *u* is the unit number.

Local DSA devices use a controller letter of ''A,'' regardless of the physical controller the device resides on. All local DSA disk devices are named DUA*n*, where *n* is a unique disk unit number. All local DSA tape devices are named MUA*n*, where *n* is a unique tape unit number. Use of a single controller letter requires that the unit number for each local DSA device be unique. Duplicate unit numbers are possible if the local disks reside on different controllers.

If the system is part of a VMScluster environment, device names are formatted in one of the following ways:

- If the device is attached to a single computer or HSC subsystem, the device name includes the node name in the format *node$ddcu*, where *node* refers to the node name of the system that the device resides on.

- If the device is to be accessed (or dual-ported) by two computers or HSC subsystems, you must identify the device with a unique, path-independent name that includes an **allocation class**. The allocation class is a numeric value from 1 to 255, which is used to create a device name in the following format:

  $allocation-class$device-name

  For example, the allocation class device name 11DUA8 identifies a disk that is accessed by two computers or HSC subsystems, both having an allocation class of 11.

For more information on the device name format in VAXcluster or VMScluster environments, see *VMScluster Systems for OpenVMS*.

7.2 Getting Information About Devices on the System

Use the DCL command SHOW DEVICES to retrieve information about devices on your system.

When you invoke the SHOW DEVICES command and do not specify a device or use a qualifier, the system displays information about all recognized devices.

Note

If a device does not appear in the display, it is not recognized by the system. The device may not be connected, or the driver may not be loaded. For certain devices, you must manually connect them and load their device drivers. For more information, see Section 7.4.

If you specify a device name with the SHOW DEVICE command, the system displays information about the device you specified. If you use certain qualifiers with SHOW DEVICES, information is displayed about those devices that currently have volumes mounted or that have been allocated to processes. See the *OpenVMS DCL Dictionary* for a list of qualifiers that can be used with the SHOW DEVICES command.

Examples

The following examples use the SHOW DEVICES command. Device protection is RWPL (read, write, physical, logical).

The SHOW DEVICES/FULL examples include both volume protection and device protection. In addition, if a volume has a protected subsystem enabled, it also appears in the display.

1. The following command shows all devices on the system:

```
$ SHOW DEVICES

Device                 Device        Error   Volume         Free   Trans  Mnt
 Name                  Status        Count    Label        Blocks  Count  Cnt
$11$DUA9:      (SNAP)  Online            0
$11$DUA10:     (SNAP)  Mounted           2  PAGE             83643     3  26
$11$DUA13:     (SNAP)  Mounted           0  WORK1           192297    36  26
$11$DUA23:     (SNAP)  Online            0
$11$DUA24:     (SNAP)  Mounted           0  MONITORPLUS     776808    86  26
```

```
DAD0:          (TULIP)  Online           0
DAD9:          (TULIP)  Online           0
DAD44:         (TULIP)  Mounted wrtlck   0  CDBIN06JUL23     97947     1   1
ROSE$MUA0:              Online           0
LAVNDR$MUA0:            Online           0
TULIP$MUA1:             Online           0
IRIS$MUA1:              HostUnavailable  0
OPA0:                   Online           0
DBA0:                   Offline          0
FTA0:                   Offline          0
FTA239:                 Online           0
LTA0:                   Offline          0
LTA3401:                Online spooled   0
LTA3402:                Online spooled   0
RTA0:                   Offline          0
RTA1:                   Mounted          1
RTA2:                   Mounted          0
RTB0:                   Offline          0
TXA0:                   Online           0
TXA1:                   Online           0
XT0:                    Offline          0
```

2. The following command requests a full listing of the status of the DAD42: RRD40 device. The device is located on node IRIS in a VMScluster environment.

```
$ SHOW DEVICES/FULL DAD42:

Disk DAD42: (IRIS), device type RRD40, is online, mounted, software write-
    locked, file-oriented device, shareable, error logging is enabled.

 Error count                    0   Operations completed               146
 Owner process                 ""   Owner UIC                      [SYSTEM]
 Owner process ID        00000000   Dev Prot    S:RWPL,O:RWPL,G:RWPL,W:RWPL
 Reference count                1   Default buffer size                512
 Total blocks             1218000   Sectors per track                    4
 Total cylinders            50750   Tracks per cylinder                  6
 Allocation class              11

 Volume label      "CDBIN06JUL21"   Relative volume number               0
 Cluster size                   3   Transaction count                    1
 Free blocks                15153   Maximum files allowed           152083
 Extend quantity                5   Mount count                          1
 Mount status              System   Cache name        "_$11$DUA21:XQPCACHE"
 Extent cache size             64   Maximum blocks in extent cache    1515
 File ID cache size            64   Blocks currently in extent cache     0
 Quota cache size               0   Maximum buffers in FCP cache      1330

Volume status:  subject to mount verification, file high-water marking,
  write-through caching enabled.
```

3. The following command requests a full informational display about each DU device. This display shows only the first two devices: the mounted DUA8: device and the unmounted DUA9: device.

```
$ SHOW DEVICES/FULL DU

Disk $11$DUA8: (SNAP), device type RA90, is online, mounted, file-oriented
    device, shareable, served to cluster via MSCP Server, error logging is
    enabled.
```

```
 Error count                    0   Operations completed               334
 Owner process                 ""   Owner UIC                      [SYSTEM]
 Owner process ID        00000000   Dev Prot    S:RWED,O:RWED,G:RWED,W:RWED
 Reference count                1   Default buffer size                512
 Total blocks             2376153   Sectors per track                   69
 Total cylinders             2649   Tracks per cylinder                 13
 Host name                 "SNAP"   Host type, avail              HS70, yes
 Alternate host name      "DRAGON"  Alt. type, avail              HS70, yes
 Allocation class              11

 Volume label             "DOCRES"  Relative volume number               0
 Cluster size                   3   Transaction count                    1
 Free blocks              1211964   Maximum files allowed           297019
 Extend quantity                5   Mount count                         26
 Mount status              System   Cache name        "_$11$DUA21:XQPCACHE"
 Extent cache size             64   Maximum blocks in extent cache   121196
 File ID cache size            64   Blocks currently in extent cache      0
 Quota cache size               0   Maximum buffers in FCP cache       1330

Volume status:subject to mount verification, write-through caching enabled.
Volume is also mounted on TANGO, WHOCRS, SIRE, BARKIN, SALT, SNEEZY, ARTIC,
       HEMMING, ROCKUP, HUTZPA, HAPPY, TOSIR, CHAIN, BOXER, SPELLN, WAYLND,
       ADAM, PINE, MONA, PALMER, WLDWND, ERICA, VALLEY.

Disk $11$DUA9: (SNAP), device type RA90, is online, file-oriented device,
    shareable, served to cluster via MSCP Server, error logging is enabled.

 Error count                    0   Operations completed               544
 Owner process                 ""   Owner UIC                      [SYSTEM]
 Owner process ID        00000000   Dev Prot    S:RWED,O:RWED,G:RWED,W:RWED
 Reference count                0   Default buffer size                512
 Total blocks             2376153   Sectors per track                   69
 Total cylinders             2649   Tracks per cylinder                 13
 Host name                 "SNAP"   Host type, avail              HS70, yes
 Alternate host name      "DRAGON"  Alt. type, avail              HS70, yes
 Allocation class              11
```

7.2.1 Determining If Volumes Need Rebuilding

If a volume was improperly dismounted, it may require rebuilding. Volumes are improperly dismounted when, for example, the system crashes. Use the /REBUILD_STATUS qualifier with the SHOW DEVICES command to determine if a volume needs rebuilding. Do not use the /REBUILD_STATUS qualifier with any other SHOW DEVICES qualifiers, except the /OUTPUT qualifier.

For each volume, SHOW DEVICES/REBUILD_STATUS returns one of the following values:

Value	Meaning
Yes	Rebuild needed
No	Rebuild not needed
Not applicable	The volume cannot be rebuilt; the volume is not a disk or the volume is write-locked
Information unavailable	Rebuild information is not available; the volume is not mounted or mount verification is taking place

Do either of the following steps to rebuild a volume:

- Use SET VOLUME/REBUILD
- Dismount the volume and then mount the volume again using MOUNT /REBUILD

Device EMUL$DKB500, in the following example, needs rebuilding.

```
$ SHOW DEVICES/REBUILD_STATUS

Device Name             Rebuild needed?

ADU15$DKA300:           Information unavailable
EDIV$DKA300:            Information unavailable
EMUL$DKB200:            No
EMUL$DKB300:            No
EMUL$DKB500:            Yes
FTA0:                   Not applicable
OPA0:                   Not applicable
```

7.2.2 Getting Information About ISO 9660-Formatted Devices

You can use the SHOW DEVICE command to retrieve information about ISO 9660-formatted devices. The following example illustrates the use of the SHOW DEVICES/FULL command to obtain information about an ISO 9660-formatted CD–ROM. Note that the ACP process name is given and that the volume status is listed as ISO 9660. The display tells the user that the mounted members of the volume set are relative volume numbers (RVN) 1, 64, and 65535.

```
$ SHOW DEVICE DKA1/FULL

Disk $1$DKA1: (VMSRMS), device type RRD40, is online, allocated,
    deallocate on dismount, mounted, software write-locked, file-oriented
    device, shareable, served to cluster via MSCP Server.

    Error count                    0    Operations completed                 9
    Owner process           "_FTA5:"    Owner UIC                    [FIN,USER]
    Owner process ID        20200066    Dev Prot            S:RWPL,O:RWPL,G:R,W
    Reference count                2    Default buffer size                512
    Total blocks                 256    Sectors per track                   32
    Total cylinders                1    Tracks per cylinder                  8
    Allocation class               1

    Volume label          "VOLUME_1"    Relative volume number               1
    Cluster size                   0    Transaction count                    1
    Free blocks                    0    Maximum files allowed                0
    Extend quantity                0    Mount count                          1
    Mount status             Process    ACP process name             "DKA1CACP"

  Volume status:  ISO 9660.
  Members of this volume set are $1$DKA1: (rvn 1), $1$DKA7: (rvn 64), $1$DKA16:
          (rvn 65535).
```

7.3 Setting Security Protection Characteristics on Devices

You can set security protection characteristics on devices using the following commands:

- INITIALIZE
- MOUNT
- SET SECURITY/PROTECTION
- SET VOLUME

For more information on these commands, see the *OpenVMS DCL Dictionary*.

By default, allocating a tape or disk device requires VOLPRO privilege. However, you can grant access to unprivileged users in two ways:

- Set device protection to grant control access to group or world, as appropriate.

- Add an ACL to the device and grant a general identifier to all users who should have access.

Section 8.3.2 has more information on the VOLPRO privilege.

7.3.1 Granting Access to a Specific Device

To grant access to a specific device, use the SET SECURITY command as shown in either of the following examples:

```
$ SET SECURITY/CLASS=DEVICE DKA300/PROT=W:RWC
```

This example grants world read, write, and control access for the device DKA300.

```
$ SET SECURITY/CLASS=DEVICE DKA300/ACL=(IDENTIFIER=CHEKOV, ACCESS=CONTROL)
```

This example grants control access for the device DKA300 to users with the CHEKOV identifier.

7.3.2 Granting Access to All Devices

The following method allows you to grant a specified class of users access to all devices:

1. Set the security template for a particular device type to allow access to a desired class of users. Use a command such as the following, which sets the template for disk devices:

   ```
   $ SET SECURITY/CLASS=SECURITY_CLASS/PROFILE=TEMPLATE=DISK -
   _$ DEVICE/ACL=(ID=CHEKOV, ACCESS=R+W+D+C)
   ```

   This access will apply to all specified devices initialized in the future.

2. Run the following command procedure:

   ```
   SYS$EXAMPLES:RESET_DEVICE_PROTECTION.COM
   ```

   This command procedure applies the protection specified in the security template to all current devices.

7.4 Connecting Devices and Loading Device Drivers

The system uses a software component called a **device driver** to control I/O operations for a particular device type. For a device to function on a system, the device must be connected, and the device driver must be loaded into memory.

The AUTOCONFIGURE command connects all devices physically attached to the system and loads their device drivers. Using AUTOCONFIGURE saves effort and reduces the possibility of error.

The site-independent startup command procedure, STARTUP.COM, automatically configures devices, because it includes the AUTOCONFIGURE command.

VAX

On VAX systems, the following commands in STARTUP.COM perform autoconfiguration:

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

Alpha

On Alpha systems, the following commands in STARTUP.COM perform autoconfiguration:

```
$ SYSMAN := $SYSMAN
$ SYSMAN IO AUTOCONFIGURE ♦
```

During autoconfiguration, the CONFIGURE phase of STARTUP.COM creates a detached process to do the following:

- Detect any devices connected to HSC devices (hierarchical storage controller devices)
- Load the drivers for HSC devices
- Make the HSC devices known to the system

The AUTOCONFIGURE command does not connect certain devices. Instead, you must manually connect these devices and load their device drivers. For example, you must perform manual autoconfiguration whenever you add a new disk to a StorageWorks BA353 or BA350 that is directly connected to a VAX or Alpha system.

In certain cases, you might want to suppress autoconfiguration of devices in system startup. See the following sections for more details.

Topic	For More Information
† Manually connecting devices and loading drivers	Section 7.4.1
‡ Manually connecting devices and loading drivers	Section 7.4.2
Suppressing autoconfiguration	Section 7.4.3

†VAX specific
‡Alpha specific

7.4.1 Manually Connecting Devices and Loading Device Drivers (VAX Only)

VAX

On VAX systems, whenever possible, use the AUTOCONFIGURE command to connect standard devices and load device drivers. However, in some cases, such as connecting non Digital devices, you cannot use the AUTOCONFIGURE command. In addition, AUTOCONFIGURE does not connect the following devices or load their device drivers:

- Console storage device
- Network communications logical device
- Virtual terminals

In addition to these devices, other devices and drivers might be present that AUTOCONFIGURE does not connect and load. On VAX systems, use the System Generation utility (SYSGEN) to manually connect devices and load device drivers.

For more information, see the SYSGEN section of the *OpenVMS System Management Utilities Reference Manual* and the *OpenVMS VAX Device Support Manual*.

Caution

Exercise extreme caution when issuing SYSGEN CONNECT and LOAD commands because the system does little error checking. An incorrect

vector address or misspelled device name, for example, will damage the I/O database and could cause the system to fail.

To manually connect special devices each time the system starts up, add these SYSGEN commands to the site-specific startup command procedure SYCONFIG.COM. For more information, see Section 5.2.4.1.

Console Storage Device

To connect the console storage device on VAX systems, use the following CONNECT command:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT CONSOLE
SYSGEN> EXIT
```

Network Communication Device

To connect the network communications logical device on VAX systems, run SYS$MANAGER:STARTNET:

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

Virtual Terminals

For information on connecting virtual terminals and loading their driver, see Section 7.5.2.

Event-Handling Device Driver

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 specific device type. (See the *OpenVMS VAX Device Support Manual* for further information.)

Example

The commands in the following example autoconfigure the devices attached to a VAX system, and connect the console block storage device and the network software device:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> AUTOCONFIGURE ALL
SYSGEN> CONNECT CONSOLE
SYSGEN> EXIT
$ @SYS$MANAGER:STARTNET ♦
```

7.4.2 Manually Connecting Devices and Loading Device Drivers (Alpha Only)

Alpha

On Alpha systems, commands for connecting devices and loading their drivers are in the System Management utility (SYSMAN). All SYSMAN commands that control and display the I/O configuration on an Alpha system contain the prefix IO.

Whenever possible, it is preferable to use the IO AUTOCONFIGURE command to connect standard devices and load device drivers.

IO AUTOCONFIGURE does not connect or load the device driver for the network communications logical device. In addition, other devices and drivers might exist that IO AUTOCONFIGURE does not connect and load.

You can connect unattached devices and devices that have nonstandard names, as well as load device drivers with the SYSMAN commands IO CONNECT and IO LOAD.

For more information, see the SYSMAN section of the *OpenVMS System Management Utilities Reference Manual* and the *OpenVMS Alpha Device Support Manual*.

Caution

Exercise great care in issuing IO CONNECT and IO LOAD commands. Incorrect use of these commands could cause the system to fail.

To connect the network communications logical device on Alpha systems, run SYS$MANAGER:STARTNET:

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

Example

The commands in the following example autoconfigure the devices physically attached to the Alpha system, load their drivers, and connect the network software device:

```
SYSMAN> IO AUTOCONFIGURE ALL
SYSMAN> EXIT
$ @SYS$MANAGER:STARTNET ♦
```

7.4.3 Suppressing Autoconfiguration of Devices

Autoconfiguration of devices saves effort and reduces the possibility of error. However, you might want to suppress autoconfiguration for the following reasons:

- To customize the order in which you configure devices
- To troubleshoot booting problems
- To allow Small Computer System Interface (SCSI) based workstations to use devices on another workstation's SCSI bus without causing conflicts during boot time

To suppress autoconfiguration, add the following command as the last line of SYS$MANAGER:SYCONFIG.COM:

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

Caution

If you set STARTUP$AUTOCONFIGURE_ALL to 0 in the last line of SYCONFIG.COM, the CONFIGURE phase of STARTUP.COM will not execute. As a result, HSC controllers (except for an HSC controller through which the system booted) and MSCP-served devices on remote nodes will not be available and satellite nodes will not be able to access network devices and boot disks. This could prevent satellite nodes from booting.

To suppress autoconfiguration, and still configure HSCs and MSCP-served devices on remote nodes, add the following lines to the end of SYCONFIG.COM:

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

These commands suppress autoconfiguration and yet still execute the CONFIGURE phase of STARTUP.COM.

Note, however, that if you add the command @SYS$SYSTEM:STARTUP CONFIGURE to SYCONFIG.COM, AUTOGEN will fail with the following error:

```
%RUN-F-CREPRC, process creation failed
-SYSTEM-F-DUPLNAM, duplicate name
```

This error is caused because SYCONFIG.COM is invoked by both STARTUP.COM and AUTOGEN. When AUTOGEN runs, the CONFIGURE process already exists (it was started when SYCONFIG.COM was executed by STARTUP.COM.) When AUTOGEN invokes SYCONFIG.COM, the command you added attempts to start a second CONFIGURE process. This command fails, causing AUTOGEN to fail.

7.5 Managing Terminals

To manage terminals, you perform the following tasks:

- Physically attach terminals to the system
- Set terminal characteristics
- Set up virtual terminals

The following sections explain setting terminal characteristics and setting up virtual terminals.

7.5.1 Setting Terminal Characteristics

Terminal device characteristics—for example, the number of characters displayed on a line—have certain default values. Changing these values might be necessary, depending on the characteristics you use with each terminal.

To change the terminal device characteristics, use a SET TERMINAL command with the appropriate qualifiers in the following format:

SET TERMINAL[/qualifier,...] [device-name[:]]

For example, the following command indicates that the width of terminal lines is 132 characters and that the size of each page is 60 lines. The /NOBROADCAST qualifier disables the reception of broadcast messages. The /PERMANENT qualifier allows you to keep terminal characteristics between terminal sessions. (You must reset characteristics each time the system reboots by adding these commands to a site-specific startup command procedure.)

```
$ SET TERMINAL/WIDTH=132/PAGE=60/NOBROADCAST/PERMANENT
```

For more detailed information about the SET TERMINAL command and its qualifiers, see the *OpenVMS DCL Dictionary*.

7.5.1.1 Setting Default Characteristics with System Parameters

To change the default terminal characteristics for all terminals on a node, you can specify values for the system parameters TTY_DEFCHAR and TTY_DEFCHAR2. For more information on these parameters, see the *OpenVMS System Management Utilities Reference Manual*. For more information on setting system parameters, see Section 14.5.

7.5.1.2 Setting Characteristics in System Startup

To execute SET TERMINAL commands each time your system boots, add these commands to a site-specific startup command procedure. If your configuration is simple, you can add the commands to SYSTARTUP_VMS.COM. If your configuration requires a large number of commands, create a separate command procedure (for example, TERM_SETUP.COM) and execute it from the SYSTARTUP_VMS.COM. When the device setup command procedure finishes executing, control returns to SYSTARTUP_VMS.COM.

Caution

Digital recommends you limit the number of SET TERMINAL commands you include in startup command procedures. Large numbers (for example, hundreds) of SET TERMINAL commands can significantly slow down system startup. If you have a large number of terminals, change the default characteristics using the system parameters TTY_DEFCHAR and TTY_DEFCHAR2 as explained in Section 7.5.1.1.

You may want to include comments to provide the names of terminal owners, as shown in the following example.

Example

The following example provides sample commands you could include in your startup procedure to set up terminal devices:

```
$ 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
```

7.5.2 Managing Virtual Terminals

Virtual terminals allow users 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

Enabling Virtual Terminals

VAX

On VAX systems, you set up virtual terminals by entering the following commands:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT VTA0/NOADAPTER/DRIVER=SYS$TTDRIVER
SYSGEN> EXIT ♦
```

Alpha

On Alpha systems, you set up virtual terminals by entering the following commands:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> IO CONNECT VTA0/NOADAPTER/DRIVER=SYS$TTDRIVER
SYSMAN> EXIT ♦
```

Virtual terminals are identified by the VTA*n*: device name. After the SYSGEN or IOGEN command is entered, any terminal with the TT2$M_DISCONNECT characteristic set prior to login is treated as a virtual terminal.

Note

LAT terminals (LTA*n*:) can be disconnected if the TT2$M_DISCONNECT characteristic is set, but remote terminals (RTA*n*:) cannot be disconnected.

You can set the TT2$M_DISCONNECT characteristic in one of two ways:

- Enable the feature on a systemwide basis by setting the appropriate bit in the system parameter TTY_DEFCHAR2. You must use this method for dynamically created terminal devices; for example, RTA*n*: devices.
- Enable the feature on a per-terminal basis by using the DCL command SET TERMINAL/DISCONNECT.

Controlling the Use of Virtual Terminals

You can control the use of virtual terminal sessions in the following ways:

- Specify the maximum number of detached processes that individual users can create by specifying a value for the UAF resource limit MAXDETACH.
- Restrict the use of virtual terminals by enabling them on a per-terminal basis.
- Restrict individual users from being able to reconnect to disconnected terminal sessions by specifying the UAF flag DISRECONNECT.

7.5.2.1 Using Virtual Terminals for Dynamic Asynchronous DECnet for OpenVMS (VAX Only)

VAX

Virtual terminals are required for dynamic asynchronous DECnet communication. A dynamic asynchronous line differs from a static asynchronous line or other DECnet line in that it is normally switched on for network use only for the duration of a dialup 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 licenses registered and loaded
- Both nodes have the asynchronous DDCMP driver NODRIVER loaded
- Both nodes have DYNSWITCH installed as a privileged shareable image
- The remote node has virtual terminals enabled

See the *DECnet for OpenVMS Networking Manual* for a detailed description of the procedure for setting up dynamic asynchronous DECnet lines. ♦

7.5.2.2 Determining the Physical Terminal Type of a Virtual Terminal

You might want to determine the physical terminal type associated with a virtual terminal. Because both direct connect and LAT lines can be virtual, you might 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 Section 24.1 for more information on LAT software.)

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 switches 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 and specifying the item TT_ACCPORNAM. The function returns the terminal server node name and port name.

7.6 Managing Printers

To manage printers attached to your system, you perform the following tasks:

Task	For More Information
Set printer characteristics	Section 7.6.1
Spool printers	Section 7.6.2.1
Despool printers	Section 7.6.2.2
Test spooling of printers	Section 7.6.2.3

The following sections explain these tasks.

7.6.1 Setting Printer Characteristics

Printer characteristics must be set prior to starting queues for the printers. The SET PRINTER command establishes characteristics for a line printer. The SET TERMINAL command can be used to set characteristics for a printer connected to a terminal or LAT port.

In addition, if you want to spool your printers, you must do so before starting the queues to be associated with those printers. For information about spooled printers, see Section 7.6.2.

To execute these commands each time your system boots, add these commands to your site-specific startup command procedure. If your configuration is simple, you can add the commands to SYSTARTUP_VMS.COM. If your configuration requires a large number of commands, create a separate command procedure (for example, PRINTER_SETUP.COM) and execute it from SYSTARTUP_VMS.COM. When the device setup command procedure finishes executing, control returns to SYSTARTUP_VMS.COM.

Example

The following example provides sample commands you could include in your startup procedure to set device characteristics for printers. This example also includes the commands used to spool printers. You generally include the commands to spool printers along with the commands to set device characteristics.

```
$! Set up line printer devices
$!
$ SET PRINTER/PAGE=60/LOWERCASE/TRUNCATE LPA0:
$ SET PRINTER/LA11/UPPERCASE/WRAP LPB0:
$ SET DEVICE/SPOOLED=(LINE_PRINT,SYS$SYSDEVICE) LPA0:
$ SET DEVICE/SPOOLED=(SYS$PRINT,SYS$SYSDEVICE) LPB0:
$!
$! Set up LAT printers
$!
$ SET TERMINAL LTA331:/SPEED=9600/DEVICE=LN03 -
  /NOBROADCAST/NOECHO/HARDCOPY/NOTYPE_AHEAD/PERMANENT
$ SET DEVICE LTA331:/SPOOLED=(MKTG$LN03_1,SYS$SYSDEVICE)
$!
$ SET TERMINAL LTA332:/DEVICE=LA210/PAGE=66  -
  /NOBROADCAST/PERMANENT
$ SET DEVICE LTA332:/SPOOLED=(LA210$PRINT,SYS$SYSDEVICE)
```

7.6.2 Using Spooled Printers

Certain application programs print output by writing or copying data directly to a printer rather than submitting it to a queue. A **spooled printer** causes such an application program to write output to an intermediate storage device (such as a disk) so that the printer targeted to print the output remains available to other system users while the program is running.

When you spool a printer, you specify a storage device and an output queue to be associated with that printer. When a process running an application directs its output to the spooled printer, the output is instead placed in a temporary file on the storage device. When the file is closed, the system submits the file for printing on the associated output queue. Both the spooling of the output file to an intermediate storage device and the subsequent queuing of a job consisting of this file occur without the direct intervention of the user.

If your system runs application programs that might write output directly to a printer, Digital recommends you spool your printers. Digital recommends you also spool your LAT printers to prevent privileged users from writing directly to a LAT printer. Writing directly to a LAT printer can cause problems for output queues that use the printer.

If you want to spool your printers, you must do so before starting the queues to be associated with those printers.

Figure 13–9 illustrates a sample configuration using spooled printers. Section 7.6.2.1 describes how to set up a spooled printer.

7.6.2.1 Spooling Printers

To spool a printer, use the SET DEVICE/SPOOLED command. This command associates the printer with a storage device (such as a disk) and an output queue.

You must spool a printer before you start the queue to be associated with the printer.

Enter the DCL command SET DEVICE/SPOOLED in the following format:

SET DEVICE/SPOOLED[=(queue-name[:],intermediate-disk-name[:])] output-device-name

You should always specify the intermediate disk and queue explicitly. If the queue you associate with the spooled output device is a generic queue, a file written to that device is sent to the generic queue, which in turn places the job in one of its target queues. As a result, a job copied to the LPA0: device, for example, might not necessarily print on the printer LPA0:, but instead might print on one of the other printers targeted by the generic queue.

When you select an intermediate storage device, make sure that it has sufficient free space for the volume of spooled output. If you plan to enforce disk quotas on the intermediate device, make sure that all expected users have a quota authorized on the intermediate device. The intermediate device must be mounted before files can be written to it.

After establishing an output device as spooled, you should test the device, because errors in disk or queue names are not detected until spooling is attempted. This step is described in Section 7.6.2.3.

You should create a command procedure to set up your output devices each time the system reboots. Include the commands to set up spooled devices in this command procedure. For more information, see Section 7.6.1.

Example

The following example illustrates sample commands used to set up spooled printers. This example also includes the command used to set device characteristics. You generally include the commands to spool printers along with the command to set device characteristics in a startup command procedure to set up output devices.

```
$! Set up and spool line printer devices
$!
$ SET PRINTER/PAGE=60/LOWERCASE/TRUNCATE LPA0:
$ SET PRINTER/LA11/UPPERCASE/WRAP LPB0:
$ SET DEVICE/SPOOLED=(SYS$PRINT,SYS$SYSDEVICE) LPA0: ❶
$ SET DEVICE/SPOOLED=(SYS$PRINT,SYS$SYSDEVICE) LPB0:
$!
$! Set up and spool LAT printers
$!
$   SET TERMINAL LTA331:/SPEED=9600/DEVICE=LN03 -
    /NOBROADCAST/NOECHO/HARDCOPY/NOTYPE_AHEAD/PERMANENT
$   SET DEVICE LTA331:/SPOOLED=(MKTG$LN03_1,SYS$SYSDEVICE)  ❷
$!
$   SET TERMINAL LTA332:/DEVICE=LA210/PAGE=66 -
    /NOBROADCAST/PERMANENT
$   SET DEVICE LTA332:/SPOOLED=(LA210$PRINT,SYS$SYSDEVICE) ❸
```

❶ Spools the output device LPA0: by associating it with the storage device SYS$SYSDEVICE and the queue SYS$PRINT. When output from an application is directed to LPA0:, the data is temporarily stored on

SYS$SYSDEVICE until the application completes. This keeps the output device LPA0: available for other jobs until the application's output is ready for printing. When the application completes, its output is submitted to the queue SYS$PRINT.

❷ Spools the LN03 device on LAT port LTA331: by associating it with the storage device SYS$SYSDEVICE and the queue MKTG$LN03_1. Digital recommends you spool LAT printers.

❸ Spools the LA210 device on LAT port LTA332: by associating it with the storage device SYS$SYSDEVICE and the queue LA210$PRINT.

7.6.2.2 Despooling a Spooled Printer

Occasionally, you might need to disable spooling on a device. For example, the SET TERMINAL command can be executed only on a despooled output device. If you need to disable spooling to an output device, use the SET DEVICE command with the /NOSPOOLED qualifier.

You must stop the corresponding queues before you can change the spooling status.

For more information about the SET DEVICE/NOSPOOLED command, see the *OpenVMS DCL Dictionary*.

7.6.2.3 Testing a Spooled Printer

After establishing an output device as spooled, you should test the device, because errors in disk or queue names are not detected until spooling is attempted. To test a spooled device, use a command procedure similar to the following:

```
$!          *****TESTING SPOOLED DEVICE***
$!
$! set the device spooled
$  SET DEVICE/SPOOLED=(SYS$PRINT,SYS$SYSDEVICE:) LPA0:
$!
$! create a test file
$  CREATE TEST.LIS
     !Add the first test record here.
     !Ctrl/Z to exit the file
$!
$! write the file to the output device
$  COPY TEST.LIS LPA0:
$  EXIT
```

7.7 Managing Tape Drives

When managing tape drives, you perform the following tasks:

Task	For More Information
Get information about tape drives	Section 7.7.1
Change tape drive characteristics	Section 7.7.2

For information about managing volumes on tape drives, see Section 8.2.

7.7.1 Getting Magnetic Tape Device Information

You can use the DCL command SHOW MAGTAPE to display the current characteristics and status of a specified magnetic tape device.

You can enter the SHOW DEVICES command to find available magnetic drives on your system. The SHOW DEVICES/FULL command enables you to retrieve additional information about the characteristics of a particular magnetic tape device.

7.7.2 Modifying Magnetic Tape Device Characteristics

Use the DCL command SET MAGTAPE to define the default characteristics associated with a specific magnetic tape device for subsequent file operations. The device must not be currently allocated to any other user.

The following examples illustrate uses of the SET MAGTAPE command in conjunction with the MOUNT command.

Examples

1.
```
$ MOUNT MTB1:/FOREIGN
$ SET MAGTAPE MTB1:/DENSITY=800
```

   The MOUNT command mounts a tape on the MTB1: device. The /FOREIGN qualifier indicates that the tape is not in the standard format used by the operating system. For example, certain Backup utility (BACKUP) operations require you to mount a tape with the /FOREIGN qualifier.

   The SET MAGTAPE command defines the density at 800 bpi for writing to the magnetic tape. (The density is reset only if the magnetic tape has never been written before.)

2.
```
$ MOUNT MTA0:/FOREIGN
$ SET MAGTAPE MTA0:/SKIP=FILES:4
```

   The MOUNT command mounts a foreign magnetic tape on the MTA0: device; the SET MAGTAPE command directs the I/O subsystem to position the magnetic tape to skip four files.

3.
```
$ MOUNT MTA1:/FOREIGN
$ SET MAGTAPE/REWIND MTA1:
```

   The MOUNT command mounts a foreign tape on the MTA1: device; the SET MAGTAPE/REWIND command rewinds the volume.

7.8 Managing a Card Reader (VAX Only)

VAX

On VAX systems, the Digital CR–11 card reader reads computer card decks. Users can submit the two following types of card decks for processing:

- Batch job card decks
- Data card decks

To ensure that card decks are processed efficiently, you must understand their characteristics and the use of the card reader. The following sections describe which cards you should check before processing a deck through a card reader, and how to determine which cards are damaged.

7.8.1 Distinguishing the Type of Card Deck (VAX Only)

Before loading a card deck into the card reader, you should determine:

- Whether the deck is a batch job or a data deck, because their processing requirements differ
- Whether the card reader is set to the correct translation mode

The following sections describe how to make these determinations.

7.8.1.1 Batch Job Card Deck (VAX Only)

A batch job card deck consists of three segments:

- Initial cards
- Program cards
- Last card

The initial two cards in a batch job card deck are the $JOB and the $PASSWORD cards. These cards log in the user and the batch job to the system. Following the initial two cards are program cards. Program cards contain instructions that direct the system to libraries, routines, and data needed to complete the batch job. The last card must be either an end-of-job command ($EOJ) card or an end-of-file (EOF) card. Either of these cards tells the system that this is the end of the job.

Checking Input

The system cannot execute the job without $JOB and $PASSWORD cards. If you are given a card deck with these cards omitted, you should return the deck so that the user can insert them.

Since the card deck contains the user's password, you must ensure that it is always handled with care to preserve the security of the user's account.

The last card in the deck must be either an $EOJ or an EOF card.

If the last card is not one of these end cards, you can type an end card on the card punch (12-11-0-1-6-7-8-9 overpunch in column 1) and place it at the end of the deck.

Checking Output

The log file produced by a card reader batch job is queued for printing to the default system printer queue, SYS$PRINT. To have the log file queued to a different queue, the user can specify the /PRINTER qualifier on the $JOB card.

If an error occurs while the system is attempting to validate the $JOB and $PASSWORD cards, the Operator Communication Manager (OPCOM) sends to the card operator an error message that reports the job card and the error.

7.8.1.2 Data Card Deck (VAX Only)

A data deck contains data that will be either read by a program or copied to a file for later use. The process that reads the data deck is usually associated with an interactive user at a terminal or with a batch job submitted by an interactive user. Since the user and process already are logged in to the system, the first card can contain any data the user specifies. Then, either the program must read the exact number of cards supplied, or the last card must be an EOF card to inform the program that this is the end of the data deck.

When a user wants a data deck to be read, you must make sure the user has allocated the card reader. If the card reader is not allocated, the system tries to submit the deck as a batch job and subsequently flushes the deck through the reader, rejecting the job.

If the program does not read the exact number of cards (as with the COPY command), the EOF card must be the last card in the deck, to inform the program that this is the end of the deck. Without this card, the program waits indefinitely for more cards, and the system prints ''card reader off line'' messages on the operator's terminal. If the card deck lacks an EOF card, you can type one on a card punch and insert it at the end of the deck.

7.8.1.3 Setting Card Reader Translation Modes (VAX Only)

For the system to read input properly, the card reader must be set to the correct translation mode—the same as the translation mode of the card punch that prepares the deck. The system supports 026 and 029 card punches.

Make sure the following conditions exist so you can set the card reader to the correct translation mode:

- The first card in the deck must be the translation mode card.
- You must know the mode in which the cards were punched.

To set the translation mode of the card reader for many decks of the same type, use the SET CARD_READER command. This command is described in the *OpenVMS DCL Dictionary*. By default, when the system is booted, the translation mode is set to 029.

7.8.2 Running the Input Symbiont Interactively (VAX Only)

To run the input symbiont interactively and take card image input from an OpenVMS Record Management Services (RMS) file, follow these steps:

1. Enter a command in the following format:

   DEFINE/USER SYS$INPUT filename

   For example:

   ```
   $ DEFINE/USER SYS$INPUT SPECIAL_FILE.DAT
   ```

2. Enter the following command:

   ```
   $ RUN SYS$SYSTEM:INPSMB
   ```

Running the input symbiont interactively requires the following:

- CMKRNL privilege
- Read access to the UAF
- Write access to the default directory of the user

All messages are displayed to the terminal rather than to the card operator. ♦

8
Managing Storage Media

This chapter discusses the following:

- Storage media terms and concepts
- Volumes and volume sets—Tasks related to setting up both disks and magnetic tape drives: initialization and mounting
- Disks—Tasks related to disk maintenance: the management of disk space and methods for detecting and repairing disk errors

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Allocating and deallocating disk and tape drives	Section 8.2
Initializing volumes	Section 8.3
Protecting volumes	Section 8.4
Mounting volumes	Section 8.5
Setting up disk volume sets	Section 8.6
Mounting ISO 9660 volume sets and groups	Section 8.7
Mounting tape volume sets	Section 8.8
Dismounting volumes and volume sets	Section 8.9
Using command procedures for media setup	Section 8.10
Managing disk space	Section 8.11
Using the Analyze/Disk_Structure utility to check and repair disks	Section 8.12
Using mount verification for recovery	Section 8.13
Using the Bad Block Locator utility to detect media errors	Section 8.14

This chapter explains the following concepts:

Concept	Section
Disks, including Files–11 structures	Section 8.1.1
Magnetic tape	Section 8.1.2
Public and private disk volumes	Section 8.1.3
Tape and disk volume protection	Section 8.4
Disk volume sets	Section 8.6.1
Disk quotas	Section 8.11.1
Mount verification	Section 8.13.1

8.1 Understanding Storage Media Concepts

Concepts related to storage media in general include the following:

Term	Definition
Device (or **Drive**)	Hardware that allows access to storage media.
Media	Physical items on which you can store data.
Volume	Logical unit of data storage; one or more media units. A disk or tape must be mounted on a device for the operating system to recognize it as a volume.

The following sections explain these concepts:

Concept	For More Information
Terms and file structure concepts related to disk media	Section 8.1.1
Terms and concepts related to magnetic tape media	Section 8.1.2
Public and private volumes that reside on storage media	Section 8.1.3
Disk and tape volume protection	Section 8.4

8.1.1 Disk Concepts

This section defines terms related to disks and describes reserved files and file structures.

8.1.1.1 Disk Terminology

A **disk** is a physical medium on which files reside. **Files–11 On–Disk Structure** refers to a logical structure given to information stored on a disk; it is a hierarchical organization of files, their data, and the directories needed to gain access to them. The OpenVMS file system implements the Files–11 disk structure and provides access control to the files located on the disk. (The term *Files–11* used alone always refers to *Files–11 On–Disk Structure*.)

Table 8–1 defines terms as they apply to disks. Appendix A discusses these terms in more detail.

Table 8–1 Files–11 On-Disk Structure Terminology

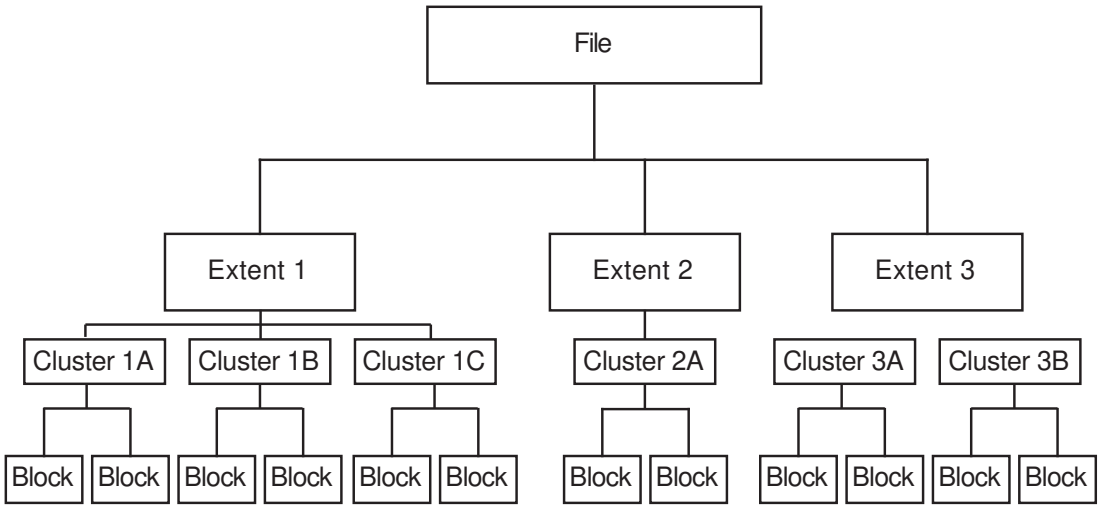
Term	Definition
Block	Files–11 defines a block as 512 8-bit bytes.
Cluster [1]	Logical grouping of blocks; basic unit by which disk space is allocated.
Extent	Contiguous blocks allocated to a particular file.
File	Array of consecutive virtual blocks, numbered 1 to *n*, plus a set of attributes with values. A file is either a data file or a directory file. Directories can contain both data files and directory files.
Volume	Disk that has been prepared for use by placing a new file structure on it.
Volume set	Combination of several volumes; has the appearance of one large volume.

[1]This usage of *cluster* does *not* refer to a set of nodes that form a VMScluster environment.

Information on a disk can be accessed as logical blocks on the disk or as virtual blocks of files on the disk.

Figure 8–1 shows the hierarchy of blocks, clusters, extents, and files in the Files–11 On-Disk Structure. The number of blocks in any one extent is a multiple of the cluster size. The figure assumes a cluster size of 2 blocks.

Figure 8–1 Files–11 On-Disk Structure Hierarchy



8.1.1.2 Disk File Structures

The operating system recognizes the following disk file structures.

File Structure	Description
†Files–11 ODS Level 1	On VAX systems, this structure is used by Digital's RSX–11M, RSX–11D, RSX–11M–PLUS, and IAS operating systems.
Files–11 ODS Level 2	This is the default disk structure of the OpenVMS operating system.
ISO 9660 volume structure	This format is used on CD–ROMs.
Foreign	A structure that is not related to a Files–11 structure. When you specify a foreign structure, you make the contents of a volume known to the system, but the system makes no assumptions about its file structure.

†VAX specific

A Files–11 volume is an array of 512-byte logical blocks. The blocks are numbered consecutively from 0 to *n–1*; the value of *n* is the size of the disk in blocks.

When you create a file, you normally specify a file name to OpenVMS Record Management Services (RMS), which assigns this name to the file on a Files–11 volume. RMS places the file name and file ID associated with the newly created file in a directory, which contains an entry defining the location for each file.

When you access the file, you supply the file name, which supplies a path to the file identifier through the directory entry. The file identifier, in turn, points to the location of the file header, which contains a listing of the extent or extents that locate the actual data.

Reserved files control the structure of a Files–11 ODS Level 2 volume. (Only five of these files are used for a Level 1 volume.) Table 8–2 identifies all reserved files, and indicates to which Files–11 ODS level they pertain.

The files listed in Table 8–2 are in the master file directory (MFD), [000000]. Appendix A contains a description of each reserved file.

Table 8–2 Reserved Files

Reserved File	File Name	†Structure Level 1	Structure Level 2
Index file	INDEXF.SYS;1	X	X
Storage bit map file	BITMAP.SYS;1	X	X
Bad block file	BADBLK.SYS;1	X	X
Master file directory	000000.DIR;1	X	X
Core image file	CORIMG.SYS;1	X	X
Volume set list file	VOLSET.SYS;1		X
Continuation file	CONTIN.SYS;1		X
Backup log file	BACKUP.SYS;1		X
Pending bad block	BADLOG.SYS;1		X
Volume security profile	SECURITY.SYS;1		X

†VAX specific

8.1.1.3 CD–ROM File Structures

Compact Disc Read Only Memory (CD–ROM) discs and readers used with computers are very similar to the CD–ROMs used for audio applications. The major differences are that CD–ROM computer disc players have a digital (rather than an audio) interface; they also incorporate internal circuitry that detects and corrects errors, thus improving the integrity of the disc data.

Following are three advantages to storing data on CD–ROM media:

- You can access data directly.
- CD–ROMs are typically less expensive than other direct-access media.
- CD–ROMs have exceptional storage space capability. Currently, a CD–ROM can hold approximately 650 megabytes of data.

You can use one or more formats to incorporate a volume and file structure that is compatible with the input/output processing used by the system. Table 8–3 shows the media formats that the OpenVMS operating system supports for CD–ROMs.

Table 8–3 CD–ROM On-Disc Formats

Format	Description
Files–11 ODS-2	The operating system's format for read/write discs
ISO 9660	An international standard for CD–ROMs: ISO 9660 : 1988
High Sierra	An initial proposal that was changed and became ISO 9660

How Data Is Arranged on a CD–ROM

Table 8–4 defines terms that describe the arrangement of data on a CD–ROM.

Table 8–4 CD–ROM Volume Structure

Term	Description
Sector	Uniquely addressable unit; each sector on a CD–ROM comprises a sequence of 2048 8-bit bytes.
Physical sector	Division of a system or data area; smallest addressable unit on an ISO 9660 CD–ROM.
Logical sector	Organizational unit of a volume; consists of one or more physical sectors. No more than one logical sector can begin in any physical sector. Logical sectors are numbered in ascending order, with 0 assigned to the logical sector having the lowest physical address containing recorded data. Each logical sector includes a data field made up of 2048 or more bytes (the number of bytes always equals a power of 2).
Volume space	Set of all logical sectors on a volume containing information about the volume.
System area	One of two divisions of CD–ROM volume space; includes logical sectors 0 through 15; reserved for system use.
Data area	One of two divisions of CD–ROM volume space; includes the remaining volume space, beginning with logical sector 16.
Logical block	Organizational unit of volume space. The logical block size cannot exceed the logical sector size.
Logical block numbering	Begins with the first byte in the volume space and continues in a sequentially ascending order through the remainder of the volume space. Logical block numbering is consecutive throughout the volume and continues through area boundaries. For example, if the last logical block in the system area is 255, the first logical block in the data area is assigned number 256.

OpenVMS System Requirements

The OpenVMS implementation of Files–11 CD conforms to the file structures defined by the ISO 9660 standard. Two aspects of the implementation are required to support ISO 9660 file structures in an OpenVMS environment: partially recorded data blocks and data interleaving.

- Partially recorded data blocks

  ISO 9660 files are recorded in the data area of the media, using extents that consist of blocks. The blocks might or might not be filled with data. The system assumes that all file blocks are filled with data, with the possible exception of the final block. When the system finds a data block that is not filled, it attempts to process the end of a file. The OpenVMS implementation of Files–11 ensures that the system does not treat an unfilled block as the end of the file unless it actually is the final file block. This is not visible to the user.

- Data interleaving

  The use of data interleaving improves access to file data. Within an extent, data is recorded using file units that are separated by an interleave gap; this gap consists of a specified number of blocks.

8.1.2 Tape Concepts

The file storage system for magnetic tapes is based on the standard magnetic tape structure as defined by ISO 1001–1986, which is compatible with several national standards such as ANSI X3.27–1987.

For more information about tape concepts, see the *Guide to OpenVMS File Applications*.

Table 8–5 defines terms that apply to magnetic tapes.

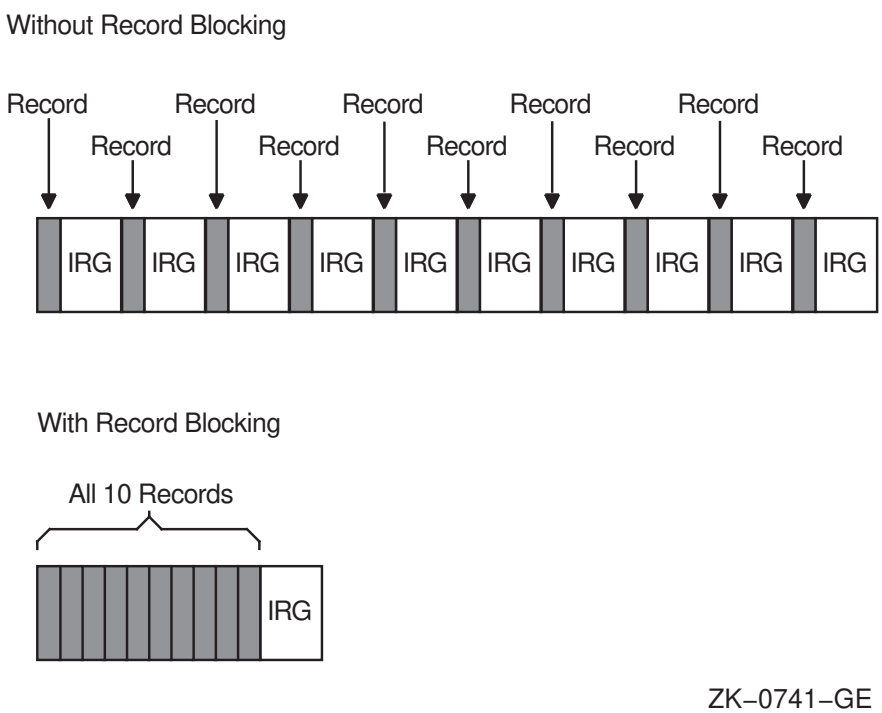
Table 8–5 Terms Related to Magnetic Tapes

Term	Definition
Block	On magnetic tape, the size of a block is determined by the user. On ODS disks, a block is fixed at a size of 512 bytes.
bpi	Bits per inch; measure used for characters of data on tape on OpenVMS systems. Also called *density*.
IRG	Interrecord gap; interval of space between blocks.
Record blocking	Grouping of individual records into a block, thereby reducing wasted space.
Sequential	Organization of magnetic tape data; that is, data is organized in the order in which it is written to the tape.
Magnetic Tape Ancillary Control Process (MTACP)	Internal software process that interprets the logical format of standard-labeled volumes.
Beginning-of-tape (BOT) marker and **end-of-tape (EOT) marker**	Pieces of photoreflective tape that appear on every volume to delimit the writable area on the volume. ANSI standards require that a minimum of 14 feet to a maximum of 18 feet of magnetic tape precede the BOT marker; a minimum of 25 feet to a maximum of 30 feet of tape, of which 10 feet must be writable, must follow the EOT marker. The EOT marker indicates the start of the end of the writable area of the tape, rather than the physical end of the tape. Therefore, data and labels can be written after the EOT marker.
Data record storage	Within tape files, data records are stored in variable-size data blocks. Each block contains one or more records. RMS provides management of records.
Header labels	Set of labels that the tape file system writes on the tape immediately preceding the data blocks when you create a file on tape. These labels contain information such as the user-supplied file name, creation date, and expiration date. To access a file on magnetic tape by the file name, the file system searches the tape for the header label set that contains the specified file name.
Trailer labels	Set of labels that the tape file systems writes on the tape following the file.
Multivolume file	Additional volume on which a file is continued when the data blocks of a file or related files do not physically fit on one volume (a reel of magnetic tape).
Volume set	The volumes on which a set of files is recorded.

Record Blocking

Figure 8–2 shows how record blocking can save space.

Figure 8–2 Record Blocking



Assume that a 1600-bits-per-inch magnetic tape contains 10 records that are not grouped into a block. Each record is 160 characters long (0.1 inch at 1600 bits per inch) with a 0.6-inch IRG after each record, which uses 7 inches of tape. However, placing the same 10 records into one block uses only 1.6 inches of tape (1 inch for the data records and 0.6 inch for the IRG).

Record blocking also increases the efficiency of the flow of data into the computer. For example, 10 unblocked records require 10 separate physical transfers, while 10 records placed in a single block require only one physical transfer. Moreover, a shorter length of magnetic tape is traversed for the same amount of data; thus, the operation is completed in less time.

However, record blocking requires more buffer space to be allocated for your program. The greater the number of records in a block, the greater the buffer size requirements. You must determine the point at which the benefits of record blocking cease. Base this determination on the configuration of your computer system and your environment.

8.1.3 Public and Private Disk Volumes

A volume is one or more units of storage media that you can mount on a device. The volume is the largest logical unit of disk file structure.

This section explains the concepts of public and private volumes.

8.1.3.1 Public Disk Volumes

A **public volume** is a file-structured disk volume that can contain both private and public files. Public volumes can be either of the following:

Type of Volume	Description
System volumes	Available to all the users on a system
Group volumes	Available to all the users in a group

As long as file protections permit it, all users have access to public volumes and to the files on them.

One way to permit users to create and store files on a public volume is to create a default directory on the public volume for each authorized user. You control access to public files and volumes by the protection codes that you establish.

A user is free to create, write, and manipulate files on a public volume only under the following conditions:

- Volume and file protection allow access, or you have privileges that allow you to access the files.
- The user has appropriate access to a directory on the volume.
- Disk quota permits usage.

Following are guidelines for setting up and maintaining public files and volumes.

Planning Public Volumes

As a system manager, you must balance users' space needs with the system's available mass storage resources. These determinations depend, in part, on whether you have relatively small or large mass storage capability. A comparison of the two follows.

Configuration	Characteristics
Small mass storage	Both system files and user files are on the same public volume. You might want to set disk quotas to ensure that user files do not exhaust the free space on the disk volume.
Large mass storage	Keep all system files on one disk volume (known as a system disk or a system volume), and keep all user files on separate volumes. The system disk is kept active reading system images, paging and swapping, spooling files, maintaining system logs, and so forth.

The most common arrangement is to have one public volume with system files and the directories of privileged users, and other public volumes dedicated to user directories, databases, and applications required by your site.

Whichever arrangement you select, plan each public volume and monitor disk performance once the system is running:

- Be sure the system disk has enough space for the operating system to boot and accept updates.
- Once the system is running, use the Monitor utility (MONITOR) to analyze disk use to determine whether disk I/O is balanced across the configuration. Section 18.8 discusses this utility in detail.

You can often move system files off the system disk and use search lists or logical names to access them. See Section 16.8 for more information.

In large configurations, you can place secondary paging and swapping files on other devices to balance disk load. See Section 15.11 for more information. The *Guide to OpenVMS Performance Management* provides detailed information on redistributing system files and achieving a balanced disk load.

8.1.3.2 Private Disk Volumes

A **private volume** is a file-structured volume that contains only private files.

Under some circumstances, users might want to perform their work on a device that unauthorized users cannot access. By creating a private volume and mounting it on a device allocated exclusively to a user's process, you ensure that users can perform their work without fear of interference from others.

Users can often prepare and manipulate their own private volumes. They might, however, need your assistance if the computer and its peripheral devices are off limits to or remotely located from them. Users requiring assistance can use the operator communication manager (OPCOM) to communicate with an operator. See Section 8.5.3 for instructions on answering users' requests for assistance.

8.2 Allocating and Deallocating Drives

This section explains how to allocate and deallocate drives. The only situation in which the ALLOCATE command is required, however, is when you must retain control of the same volume across dismounts. An example of this is when you alternate between mounting a tape /FOREIGN and /NOFOREIGN.

8.2.1 Allocating Drives

You use the DCL command ALLOCATE to logically assign a disk drive or a tape drive to your process. You might do this if you suspect an error and want to reserve a disk while you repair the error.

The ALLOCATE command allocates only one device to a process. To allocate several devices, you must use multiple commands.

How to Perform This Task

Enter the ALLOCATE command using the following format:

```
ALLOCATE device-name[:] [logical-name]
```

where:

device-name	Specifies the drive on which the volume will be loaded. The device name can be a physical, generic, or logical name.
logical-name	Specifies an optional logical name to be associated with the specified disk or tape drive.

Examples

1.
```
$ ALLOCATE DUA2:
%DCL-I-ALLOC, _MARS$DUA2: allocated
```

   In this example, the ALLOCATE command specifies a physical device named DUA2:, which requests the allocation of a specific RK10 disk drive; that is, unit 2 on controller A. The response from the ALLOCATE command indicates that the device was successfully allocated.

2. `$ ALLOCATE/GENERIC RA90 MYDISK`

   This example shows how to use the /GENERIC qualifier with the ALLOCATE command to allocate a particular type of device. In this case, the system allocates the first available RA90 drive to your process.

For further discussion of the /GENERIC qualifier and other qualifiers that you can use with the ALLOCATE command, see the *OpenVMS System Management Utilities Reference Manual*. The *OpenVMS User's Manual* contains additional examples of the ALLOCATE command.

8.2.2 Deallocating Drives

Allocating a device reserves that device for exclusive use by your process. The device remains allocated to your process until you explicitly deallocate it or until you log out.

Logging out of a process from which drives have been allocated automatically deallocates all explicitly and implicitly allocated drives; therefore, explicitly deallocating a disk or a tape drive that has been allocated to your process is not necessary. Digital, however, recommends that you use the DEALLOCATE command (or a command procedure containing this command) explicitly to deallocate all the drives you allocated with the ALLOCATE command.

How to Perform This Task

Use the DCL command DEALLOCATE to explicitly deallocate a disk drive or tape drive that has been allocated to your process. A complement to the ALLOCATE command, the DEALLOCATE command logically disconnects a drive from your process and returns it to the pool of devices.

Enter the DEALLOCATE command using the following format:

DEALLOCATE device-name[:]

where:

device-name	Specifies the drive on which the volume will be loaded. The device name can be a physical, generic, or logical name.

Example

The following example shows how to explicitly deallocate a tape drive or a disk drive:

```
$ DEALLOCATE MUA1:
```

In this example, the DEALLOCATE command logically disconnects tape drive MUA1: from your process. The system returns you to DCL level.

8.3 Initializing Volumes

To set up a disk or tape volume, issue DCL commands in the following order:

1. INITIALIZE	Formats the volume and writes an identifying label on it. This effectively removes the previous contents of the volume. (Initializing a volume each time you use it is not necessary.)
2. MOUNT	Provides the user's process with access to a volume's files or data.

This section contains instructions for initializing volumes. Section 8.5 contains instructions for mounting volumes. Before you initialize a volume, you might want to refer to Section 8.4, which contains information about volume protection.

Setting Up Media on a Workstation

For workstations with removable media, users can perform the tasks shown in Table 8–6 unassisted.

Table 8–6 Tasks Users Can Perform Unassisted

Task	Description
Load	Insert the media into the drive.
Initialize	Remove all previous contents from the media. (VOLPRO privilege is required for most operations.)
Mount	Logically mount the media and allocate the device (requires SYSNAM, GRPNAM, or VOLPRO privilege for various operations). To mount a volume on a device, you must have read (R), write (W), or control (C) access to that device.
Perform file operations	Access files and perform the desired operations on them.
Dismount	Logically dismount the media and deallocate the device (requires GRPNAM and SYSNAM user privileges to dismount group and system volumes).
Unload	Remove the media from the drive compartment.

For additional information on manipulating removable media on your workstation, see the hardware manuals that accompany your workstation.

VAX

On VAX systems, also see the upgrade and installation supplement for your computer. ♦

You initialize a disk or tape volume for one or both of the following reasons:

- To delete all old information from the volume.
- To give the volume a structure that is recognized by the operating system.

  This structure prepares a volume to receive data and stores it so that the operating system can locate it easily.

Before you or any user can write files or data to a disk or a tape volume, you must initialize a volume.

Caution

Initializing a disk volume removes links to existing files on the volume, which, in effect, deletes (but does not erase) the files. To erase the data in a file, use the INITIALIZE/ERASE command.

Do not initialize a volume that contains data that users want to keep. (Initializing a volume each time you use it is not necessary.)

How to Perform This Task

Use the DCL command INITIALIZE to format and write a label to the volume. To initialize a disk or tape volume, enter the INITIALIZE command using the following format:

```
INITIALIZE device-name[:] volume-label
```

where:

device-name	Specifies the name of the device on which the volume is to be physically mounted and then initialized. To prevent initializing another user's volume, allocate a device before you initialize the volume. (Prior allocation is not required, however.)
volume-label	Specifies the identification to be encoded on the volume; is 1 to 12 alphanumeric characters.

To initialize a public volume, you must specify the /SYSTEM qualifier with the DCL command INITIALIZE:

```
INITIALIZE/SYSTEM device name[:] volume-label
```

For more details on INITIALIZE command format, see the *OpenVMS DCL Dictionary*.

Examples

1. `$ INITIALIZE DUA2: TEMP`

   The command in this example initializes the disk volume DUA2: and labels the volume TEMP.

2. `$ INITIALIZE MUB2: TEST`

   The command in this example initializes the tape volume on MUB2: and labels the volume TEST.

The *OpenVMS User's Manual* contains additional examples of the INITIALIZE command.

8.3.1 Using INITIALIZE Command Qualifiers

Table 8–7 describes a number of qualifiers you can use with the INITIALIZE command. Selecting appropriate values for these qualifiers and selecting the appropriate position for the index file involve tradeoffs. The *OpenVMS DCL Dictionary* contains more information about each qualifier.

Table 8–7 INITIALIZE Command Qualifiers

Qualifier	Description
/CLUSTER_SIZE=*n*	Specifies minimum allocation unit in blocks.
/HEADERS=*n*	Specifies the number of file entries, called **file headers**, that you expect to have in INDEXF.SYS, the index file. It controls how much space is initially allocated to INDEXF.SYS for headers. (The system accesses the index file each time it locates a file on disk.) Each file on a disk requires at least 1 file header and each header occupies 1 block within INDEXF.SYS. Files that have many access control entries (ACEs) or that are very fragmented might use more than 1 header. The default value of 16 leaves room for fewer than 10 files to be created before INDEXF.SYS must extend. Therefore, estimate the total number of files that will be created on the disk and specify it here. A good estimation improves performance of disk access. Setting the number too low can result in a fragmented index file. However, if you set the number too high, space allocated to headers cannot be made available later for file storage and can lead to wasted disk space. This value cannot be changed without reinitializing the volume. INDEXF.SYS is limited as to how many times it can extend. When the map area in its header (where the retrieval pointers are stored) becomes full, files cannot be created and the message SYSTEM-W-HEADERFULL is displayed.
/INDEX=*position*	Determines the location of the index file on a volume, using the keyword BEGINNING, MIDDLE, or END. The index file lists the names and addresses of all disk files, so it is constantly referenced.
/MAXIMUM_FILES=*n*	Specifies the maximum number of entries in the index file, and therefore limits the number of files that a volume can contain. Once set, the maximum number of files for a volume cannot be increased without reinitializing the disk.
/PROTECTION=*keyword*	Specifies the protection code to be assigned to a volume. See Section 8.4 for details.
/WINDOWS=*n*	Sets the default number of mapping pointers to be allocated for file windows. When a file is opened, the file system uses mapping pointers to access data in the file. The file system can read one file segment into memory for each available pointer.

Caution

The default value for the /HEADER qualifier is generally insufficient for ODS-2 disks. To improve performance and avoid SYSTEM-F-HEADERFULL errors, Digital strongly recommends that you set this value to be approximately the number of files that you anticipate having on your disk. However, grossly overestimating this value will result in wasted disk space.

Examples

1. `$ INITIALIZE/HEADERS=100000 DUA3:`

   This example shows how many entries to allocate in the index files for a large disk (a small disk might allocate 2000 entries).

2. `$ INITIALIZE/MAXIMUM_FILES=20000 DUA3:`

   This example shows how to specify the characteristics of a small disk. Note that each directory and each extension header of a multiheader file counts as a file against this maximum value.

3. `$ INITIALIZE/WINDOWS=10 DUA3:`

   This example shows how to cite a large number of pointers for a large disk of 500 MB.

8.3.2 Assisting Users in Accessing and Initializing Volumes

Initializing volumes for users might be necessary in some circumstances:

- If the volume previously contained data, the protection code might prevent users from accessing and initializing the volume.
- If the first file section on the volume has not reached its expiration date, users might not be able to initialize a tape volume.
- If the volume is owned by anyone except [0,0], the user must have VOLPRO privilege to override volume protection. If users do not have VOLPRO privilege, they must ask the previous owner of the volume or you, as system manager, to initialize it for them.
- If a tape is blank, the user must have VOLPRO and OPER privileges to access and initialize it.

8.4 Protecting Volumes

Protection based on user identification codes (UICs) restricts users' access to volumes. By assigning access types to volumes, you determine the kinds of actions various groups of users can perform on volumes. Section 8.4.1 and Section 8.4.2 explain the differences between UIC-based protection for disk and tape volumes.

For additional access control, you can set access control lists (ACLs) on volumes. Volume ACLs are copied from the VOLUME.DEFAULT security class template. See Section 11.5 for more information about ACLs.

For a full explanation of OpenVMS protection, see Chapter 11 or the *OpenVMS Guide to System Security*.

Table 8–8 shows the types of access you can assign to disk and tape volumes.

Table 8–8 Access Types for Disk and Tape Volumes

Access Type	Description
Read	Gives you the right to examine file names, print, or copy files from the volume. System and owner categories always have read access to tape volumes.
Write	Gives you the right to modify or write to existing files on a volume. The protection of a file determines whether you can perform a particular operation on the file. To be meaningful, write access requires read access. System and owner categories always have write access to tape volumes.

(continued on next page)

Table 8–8 (Cont.) Access Types for Disk and Tape Volumes

Access Type	Description
Create	Gives you the right to create files on a disk volume and subsequently modify them. Create access requires read and write access. This type of access is invalid for tape volumes.
Delete	Gives you the right to delete files on a disk volume, provided you have proper access rights at the directory and file level. Delete access requires read access. This type of access is invalid for tape volumes.
Control	Gives you the right to change the protection and ownership characteristics of the volume. Users with the VOLPRO privilege always have control access to a disk volume, with the following exceptions: • Mounting a file-structured volume as foreign requires control access or VOLPRO privilege. • Mounting a volume containing protected subsystems requires SECURITY privilege. Control access is not valid with tapes.

For more information on specifying protection codes, see the *OpenVMS Guide to System Security*. Chapter 11 discusses protection in general.

The following sections explain how to perform these operations:

Task	Section
Protecting disk volumes	Section 8.4.1
Protecting tape volumes	Section 8.4.2
Auditing volume access	Section 8.4.3

8.4.1 Protecting Disk Volumes

For file-structured ODS-2 volumes, the OpenVMS operating system supports the types of access shown in Table 8–8. The system provides protection of ODS-2 disks at the volume, directory, and file levels. Although you might have access to the directories and files on the volume, without the proper volume access, you are unable to access any part of a volume.

The default access types for the disk volume owner [0,0] are:

S:RWCD, O:RWCD, G:RWCD, W:RWCD.

The system establishes this protection with the default qualifier of the INITIALIZE command (/SHARE). Any attributes that you do not specify are taken from the current default protection.

Ways to Specify Protection

You can change permanently stored protection information in the following ways:

- Use ACLs. The entire security profile (owner UIC, protection code, and ACL) is stored on the volume. If you change the volume security profile for a volume mounted clusterwide, the change is distributed to all nodes in the cluster. If you dismount and remount a volume, the security profile for the volume is preserved.
- Use the DCL command SET SECURITY to modify the default security profile after a volume is mounted, including UIC- and ACL-based protection.

- Use protection qualifiers with the DCL command INITIALIZE to specify UIC-based protection when you initialize a volume.

  You can also specify protection when you mount disk volumes, but you ordinarily do not do so because the protection that you specify is in effect only while the volume is mounted. For details, refer to the Mount utility (MOUNT) in the *OpenVMS System Management Utilities Reference Manual*.

- Use the DCL command SET VOLUME after you mount a volume to change UIC-based volume protection.

The following sections explain how to perform these tasks:

Task	Section
Specify protection when you initialize volumes	Section 8.4.1.1
Change protection after volumes are mounted	Section 8.4.1.2
Display protection	Section 8.4.1.3

8.4.1.1 Specifying Protection When You Initialize Disk Volumes

This section explains how to specify UIC-based volume protection and ISO 9660-formatted media protection when you initialize volumes.

Specifying UIC-Based Protection

You can specify protection in one of the following ways when you initialize volumes:

- Use the /PROTECTION qualifier with the INITIALIZE command. For example:

  ```
  $ INITIALIZE DUA7: ACCOUNT1/PROTECTION=(S:RWCD,O:RWCD,G:R,W:R)
  ```

  This example specifies a protection code for the disk volume ACCOUNT1 on the DUA7: device. The UIC of the volume is set to your process UIC.

- Use one or more of the qualifiers shown in Table 8–9 with the INITIALIZE command. However, the protection that you set using the /PROTECTION qualifier overrides any of the defaults set when you use any other qualifier.

  ##### Using INITIALIZE Command Qualifiers for Protection

  You usually do not change volume protection after you initialize a volume. By specifying a protection qualifier with the INITIALIZE command, you can establish the default protection of a volume. (The default qualifier of the INITIALIZE command is /SHARE, which grants all types of ownership all types of access.)

  Table 8–9 explains the qualifiers you can use to specify disk volume protection when you initialize disk volumes.

Table 8–9 INITIALIZE Command Qualifiers for Protection

Qualifier	Explanation
/PROTECTION	The protection you specify with this qualifier overrides any protection you specify with other qualifiers.
/SYSTEM	All processes have read, write, create, and delete access to the volume, but only system processes can create first-level directories. ([1,1] owns the volume.) See the note following this table.
/GROUP	System, owner, and group processes have read, write, create, and delete access to the volume. World users have no access.
/NOSHARE	System and owner processes have read, write, and delete access to the volume. World users have no access. Group users also have no access unless you specify the /GROUP qualifier.

Note

The /SYSTEM qualifier grants all users complete access. However, users cannot create directories or files unless you do one of the following:

- Change the protection on the newly created master file directory (MFD), [000000]000000.DIR;1 to allow users to create their own directories under this parent directory.
- Under the master file directory, create user directories that give users write access so that they, in turn, can create their own directories.

System managers usually choose the second method.

Table 8–10 shows the UIC and protection that the system sets for disk volumes when you use the default, /SHARE, and other qualifiers with the INITIALIZE command.

Table 8–10 Protection Granted with INITIALIZE Command Qualifiers

Qualifier	UIC	Protection
/SYSTEM	[1,1]	S:RWCD,O:RWCD,G:RWCD,W:RWCD
/SYSTEM/NOSHARE	[1,1]	S:RWCD,O:RWCD,G:RWCD,W:RWCD
/GROUP	[x,0]	S:RWCD,O:RWCD,G:RWCD,W
/SHARE (the default)	[x,x][1]	S:RWCD,O:RWCD,G:RWCD,W:RWCD
/NOSHARE	[x,x][1]	S:RWCD,O:RWCD,G,W

[1] x,x is the UIC of the process that performs the initialization.

Specifying ISO 9660-Formatted Media Protection

The OpenVMS implementation of ISO 9660 does not include volume or volume set protection. The protection specified for the device on which the media is mounted determines accessibility to the ISO 9660 volumes or volume sets.

By default, the device protection is assigned to ISO 9660 files and directories. When you mount the volume, you can specify additional file protection using the UIC and PERMISSION protection fields included in the Extended Attribute Records (XARs) that might be associated with each file.

You can enable the protection fields by specifying either of the following:

- XAR mount option, using the following format:

  MOUNT/PROTECTION=XAR

  When you specify the XAR option for a file that has an associated XAR, the protection fields in the XAR are enabled.

- Digital System Identifier (DSI) mount option, using the following format:

  MOUNT/PROTECTION=DSI

  If you specify the DSI option, you enable the XAR permissions Owner and Group for XARs containing DSI.

For more information about the XAR and DSI options, see the *OpenVMS Record Management Utilities Reference Manual*.

8.4.1.2 Changing Protection After Disk Volumes Are Mounted

You can change protection by using the SET SECURITY/CLASS=VOLUME command with the /PROTECTION, /OWNER, or /ACL qualifier to change any aspect of the volume security profile.

Changing UIC-Based Protection

To change UIC-based protection after a volume is mounted, use the SET SECURITY/CLASS=VOLUME/PROTECTION command. For example:

```
$ SET SECURITY/CLASS=VOLUME/PROTECTION=(S:RWCD,O:RWCD,G:RC,W:RC) DUA0:
```

The protection set in this example allows the system and owner all types of access. Group and world access types can only read files and run programs. Any category not specified in the protection code (S,O,G,W) is unchanged.

Changing ACL-Based Protection

To change ACL-based protection after a volume is mounted, use the SET SECURITY/CLASS=VOLUME/ACL command. To change the ACL, for example:

```
$ SET SECURITY/CLASS=VOLUME/ACL=(IDENTIFIER=DOC,ACCESS=READ+WRITE+EXECUTE) -
_$ $1$DSA7:
```

This example gives holders of the DOC identifier read, write, and execute access to the 1DSA7: volume.

8.4.1.3 Displaying UIC- and ACL-Based Protection

You can use the SHOW SECURITY/CLASS=VOLUME command to display protection. For example:

```
$ SHOW SECURITY/CLASS=VOLUME $1$DSA27:
```

Following is an example of the resulting display:

```
$1$DSA27: object of class VOLUME
     Owner: [1,1]
     Protection: (System: RWCD, Owner: RWCD, Group: RWCD, World: RWCD)
     Access Control List:
          (IDENTIFIER=[ABC,SADAMS],ACCESS=READ+WRITE+CREATE+DELETE)
```

In the display are the name and profile of the VOLUME class object 1DSA27. The profile includes the owner UIC, the protection code, and the access control list (ACL) of the protected object.

8.4.2 Protecting Tape Volumes

The system protects magnetic tapes only at the volume level. You establish protection when you initialize tape volumes; after that, the Mount utility (MOUNT) enforces the protection that you have established.

You can use two levels of protection for tape volumes:

Level of Protection	Description
Guidelines of the ISO standard	The ISO standard, which is the first level of protection, is encoded in the accessibility field of the first volume label written on the magnetic tape. With this protection scheme, you can protect tape volumes in environments where interchange exists between the OpenVMS system and the operating system that is not OpenVMS.
UIC-based protection scheme supported by system software	This second level of protection is encoded in the second volume label written on the magnetic tape. Only OpenVMS systems check this scheme; it is ignored in any interchange with operating system that are not OpenVMS.

Standard-Labeled Tape Protection

The OpenVMS tape file system bases its accessibility protection on the ISO standards. This protection allows an installation routine to use a routine that interprets the contents of the volume- and header-label accessibility field. See the $MTACCESS system service in the *OpenVMS System Services Reference Manual* for more information on installation routines.

Access Types with Default Protection

When you do not supply a protection code during initialization, all users receive read and write access, explained in Table 8–11.

Table 8–11 Access Types for Tape Volume Protection

Access Type	Description
Read	Gives you the right to examine, print, or copy files from the volume.
Write	Gives you the right to append or write files to the volume.

The security profile of a tape volume is stored in the tape volume 1 and volume 2 labels. Volume 2 contains system-specific information, which is not written if you specify the /INTERCHANGE qualifier.

Foreign Volume Protection

The operating system also supports foreign tape volumes. (**Foreign** volumes either lack the standard volume label or have been mounted /FOREIGN.) When a tape volume is mounted with the /FOREIGN qualifier, users in the system and owner categories are always given full access (read, write, logical, and physical), regardless of what is specified in the protection code.

8.4.2.1 Using the /PROTECTION Qualifier with Tape Volumes

If you use the /PROTECTION qualifier when you initialize tape volumes, the protection code is written to a system-specific volume label.

With the /PROTECTION qualifier, the system applies only read (R) and write (W) access restrictions. (Execute [E] and delete [D] access do not apply.) The system and the owner always receive both read (R) and write (W) access to magnetic tapes, regardless of the protection code you specify.

8.4.2.2 Protecting Tape Volumes for Interchange Environments

You can protect tape volumes for interchange between OpenVMS and other operating systems.

Following are guidelines for protecting specific types of magnetic tapes:

- With tapes processed on any operating system that supports a version of the ANSI standard later than Version 3, the system processes accessibility information in the first volume label.
- Tapes processed on operating systems other than VMS VAX Version 4.0 or later have protection characteristics encoded in the ANSI standard tape label. (Within the tape label is the volume label of the tape.)
- To process magnetic tapes created on a Digital operating system other than the OpenVMS operating system, a user must have VOLPRO privilege and must explicitly override the check on the protection as follows:
  - If the tape was created with a specified accessibility, then a user must have the appropriate privilege and must explicitly override the check on accessibility.
  - If the tape volume was not created with such a protection scheme, then a user is granted read and write access to that tape volume.
- The tape file system allows you to specify values for the fields in which other Digital operating systems currently write their protection information. Except under conditions described in the two previous sections, the OpenVMS operating system does not process these fields. Thus, you can use these fields to store the protection values for another operating system without affecting the system protection characteristics on that particular volume.

8.4.3 Auditing Volume Access

You can enable auditing for the volume object class; the system then audits disk volume access, with the following exceptions:

- The system does not audit volume creation or deletion.
- The system does not audit access for tapes, ODS-1, or foreign-mounted volumes.

8.5 Mounting Volumes

Mounting a disk or tape volume establishes a relationship between the volume and the device on which the volume is physically loaded. After you mount a volume, the system knows it exists, and users can access it. (This section assumes that you are performing the mount operation yourself.)

File-Structured and Foreign Volumes

Ordinarily, when you mount volumes, the system imposes a format on each volume that allows you to read, write, create (or execute), and delete files. These mounted volumes have the format of the OpenVMS operating system.

If you specify the /FOREIGN qualifier when you mount a volume, the system does not impose a format on the media, and you cannot access the files on the mounted volume. You would use the /FOREIGN qualifier if you were mounting volumes with formats of operating system that are not OpenVMS or with private formats.

Because foreign volumes are not file-structured, you must access them as follows:

- Disks—sequentially or by logical block number
- Tapes—sequentially

At times, the Backup utility (BACKUP) requires you to mount volumes with the /FOREIGN qualifier, when you restore an entire disk, for example. For details, see the *OpenVMS System Management Utilities Reference Manual*.

How to Perform This Task

When mounting volumes, follow these steps:

1. Physically mount all disks and put them on line.
2. Enter the MOUNT command (which invokes the Mount utility), using the following format:

   ```
   MOUNT device-name volume-label logical-name
   ```

   where:

device-name	Specifies the physical device name or logical name of the device on which the volume is to be mounted.
volume-label	Specifies the label on the volume.
logical-name	Defines a logical name to be associated with the device.

Once invoked, the Mount utility does the following:

1. Allocates the device
2. Checks to see that the device is correctly loaded
3. Reads and verifies the volume identification

Using Qualifiers with the MOUNT Command

Under special conditions, you must add qualifiers to the MOUNT command; for example:

- To mount a public volume, include the /SYSTEM qualifier with the DCL command MOUNT using the following format:

  ```
  MOUNT/SYSTEM device-name volume-label logical-name
  ```

- In a VMScluster environment, also specify the /CLUSTER qualifier:

  ```
  MOUNT/SYSTEM/CLUSTER device-name volume-label logical-name
  ```

Table 8–12 and Table 8–13 show, respectively, the qualifiers you can use when you mount disks and tapes.

The following sections explain how to perform these tasks:

Task	Section
Use MOUNT command qualifiers when you mount disks	Section 8.5.1
Use MOUNT command qualifiers when you mount tapes	Section 8.5.2
Assist users with mounting	Section 8.5.3
Mount a volume with a protected subsystem	Section 8.5.4
Modify disk volume characteristics	Section 8.5.5

8.5.1 Using MOUNT Command Qualifiers When You Mount Disks

Table 8–12 lists MOUNT command qualifiers you can use to mount disks. The *OpenVMS System Management Utilities Reference Manual* has more information about each qualifier.

Table 8–12 MOUNT Command Qualifiers for Mounting Disks

Qualifier	Description
/ACCESSED=*n device-name*	Requires OPER privilege; specifies the approximate number of directories that will be in use concurrently on the volume. (This qualifier is obsolete for ODS-2.) For example, on a large 500 megabyte (MB) disk you might select a value of 40, but on a small disk you might specify the following: `$ MOUNT/ACCESSED=2 DUA3:`
/ASSIST *device-name*	Directs the mount operation to allow operator or user intervention if the mount request fails. The /ASSIST qualifier is the default except during system startup. Encourage users to take advantage of this feature, which repeatedly alerts the operator of a mount request until the request is satisfied. To disable operator-assisted mounts, enter a command like the following: `$ MOUNT/SYSTEM/NOASSIST DUA1: SALES_93`
/BIND=*volume-set-name device-name volume-label*	Creates a volume set of one or more disk volumes or adds one or more volumes to an existing volume set. For example: `$ MOUNT/SYSTEM/BIND=CLIENTS DUA0:,DUA1: EUROPE,ASIA` See Section 8.6.1.2 for details.

/CACHE=*keyword*	Controls whether caching limits established at system generation are disabled or overridden. For example:
	`$ MOUNT/CACHE=(EXTENT=60,FILE_ID=60,QUOTA=20) -`
	`_$ DMA0: FILES WORK`
	`%MOUNT-I-MOUNTED, FILES  mounted on _NODE$DMA0:`
	This command mounts an RK07 device labeled FILES and assigns the logical name WORK. The /CACHE qualifier enables an extent cache of 60 entries, a file identification cache of 60 entries, and a quota cache of 20 entries.

/CLUSTER *device-name*	Requires SYSNAM privilege; specifies that after a volume is successfully mounted on the local node, or if it is already mounted with the /SYSTEM qualifier on the local node, it is to be mounted on every other node in the existing VMScluster environment (that is, the volume is to be mounted clusterwide). For example: `$ MOUNT/SYSTEM/CLUSTER DUA1: SALES_93`
/COMMENT=*"string" device-name*	Specifies additional information to be included with the operator request when the mount operation requires operator assistance. For example: `$ MOUNT/SYSTEM DYA1: SALES_93/COMMENT="Vol. in Rack 2."`
/EXTENSION=*n device-name*	Requires OPER privilege; specifies the number of blocks by which disk files are to be extended on the volume unless otherwise specified by an individual command or program request. The cluster size sets the initial disk block allocation; the /EXTENSION qualifier determines how the file grows. For example, for a small disk with a cluster size of 1 disk block, you might select an extension size of 2 disk blocks: `$ MOUNT/EXTENSION=2 DUA3:`

(continued on next page)

Table 8–12 (Cont.) MOUNT Command Qualifiers for Mounting Disks

Qualifier	Description
/FOREIGN *device-name*	Indicates that the volume is not in the standard format used by the operating system. Use this qualifier if you want to mount a disk volume with a file structure other than Files–11 or ISO 9660; for example (using DISK as a logical name): `$ MOUNT/FOREIGN DISK`
/MEDIA_ FORMAT=CDROM	Mounts a volume assuming the media to be ISO 9660 (or High Sierra) formatted.
/[NO]MOUNT_ VERIFICATION *device-name*	Enables or disables the mount verification feature on disks. By default, the mount verification feature is enabled. If a device goes off line or becomes write-locked, mount verification notifies the operator of the error condition, and then checks to see that the volume identification before and after the error condition are identical. To disable mount verification, enter a command like the following: `$ MOUNT/SYSTEM/NOMOUNT_VERIFICATION DUA1: ACCOUNTS_DUE`
/OVERRIDE=*keyword*	Inhibits one or more protection checks that the MOUNT command performs.
/PROTECTION=*keyword*	Specifies the protection code to be assigned to the volume. Keywords are the following: • Protection code: specifies the protection code according to the standard syntax rules for specifying user protection (that is, system/owner/group /world). • XAR: enables enforcement of the extended record attribute (XAR) access controls (ISO 9660 only). • DSI: enables XAR permissions owner and group for XARs containing Digital System Identifiers (DSI). (ISO 9660 only.) See Section 8.4.1 for details.
/SHARE *device-name*	Specifies that other users can access the volume. (However, you must use the /SYSTEM qualifier to mount public volumes.) Two users can access a private volume simultaneously if they both use MOUNT/SHARE. For example: `$ MOUNT/SHARE DLA0: COST_ACCOUNT` Using the MOUNT/SHARE command on disks already mounted with the /SYSTEM qualifier retains a lock on disk availability even if the disk is dismounted on a systemwide basis. This practice is not usually used for the system disk, but it can occur as a result of invoking a general-purpose command procedure that is sometimes used on system and nonsystem disks. If the DISMOUNT.EXE program is opened by a user and another user enters the MOUNT/SHARE command on the system disk, a subsequent dismount may produce a warning message that the disk cannot be dismounted. To prevent the message, install the DISMOUNT.EXE image.
/SUBSYSTEM	Enables the processing of subsystem ACEs. (The command MOUNT /SUBSYSTEM requires the SECURITY privilege.) By default, the disk from which you boot has /SUBSYSTEM enabled but other disks do not. The following command uses the MOUNT command with the /SUBSYSTEM qualifier to enable the processing of subsystem ACEs on the DUA0: device (DOC is the volume label; WORK8 is an optional logical name for the volume): `$ MOUNT/SUBSYSTEM/SYSTEM DUA0: DOC WORK8`

(continued on next page)

Table 8–12 (Cont.) MOUNT Command Qualifiers for Mounting Disks

Qualifier	Description
/SYSTEM *device-name*	Requires SYSNAM privilege; makes the volume public, that is, available to all users of the system, as long as the UIC-based volume protection allows them access. The following command mounts the volume labeled WORK and makes it available systemwide: `$ MOUNT/SYSTEM DUA1: WORK`
/UCS_ SEQUENCE=(escape_ sequence)	Supplies the escape sequence to select the coded graphic character set, a requirement when mounting an ISO 9660 volume for one of its supplementary volume descriptors (SVDs).
/UNDEFINED_FAT	Establishes default file attributes to be used for records on ISO 9660 media for which no record format has been specified.
/WINDOWS=*n device-name*	Requires OPER privilege; specifies the number of mapping pointers to be allocated for file windows. The default number of windows is set with the INITIALIZE command. The following example specifies a modest number of pointers: `$ MOUNT/WINDOWS=4 DUA3:`

8.5.2 Using MOUNT Command Qualifiers When You Mount Tapes

Table 8–13 lists MOUNT command qualifiers you can use to mount a tape volume. For a complete list of MOUNT command qualifiers, see the *OpenVMS System Management Utilities Reference Manual*.

Unless otherwise noted, you must have VOLPRO privilege to use any of these qualifiers when the volume is a standard-labeled volume containing protection that disallows your process from accessing the volume.

Table 8–13 MOUNT Command Qualifiers for Mounting Tapes

Qualifier	Description
/BLOCKSIZE=*n*	Specifies the block size for the magnetic tape. The range of valid values for *n* varies, depending on the density of the volume, whether the data is for input or output, and whether the operation uses OpenVMS RMS. By default, the system writes 2048-byte blocks.
/CACHE=TAPE_DATA	Requires OPER privilege; enables the write cache for a tape device if the tape controller supports one. /NOCACHE is the default for mounting tape devices. You must specify TAPE_DATA to enable write caching. The write buffer stays enabled even after you dismount the tape.
/FOREIGN *device-name*	Indicates that the volume is not in the standard format used by the operating system.
/HDR3 *device-name*	Controls whether special header labels are written on a tape volume. This is the default.

(continued on next page)

Table 8–13 (Cont.) MOUNT Command Qualifiers for Mounting Tapes

Qualifier	Description
/[NO]MOUNT_ VERIFICATION *device-name*	Enables or disables the mount verification feature on magnetic tapes. By default, the mount verification feature is enabled. If a device goes off line or becomes write-locked, mount verification notifies the operator of the error condition, and then checks to see that the volume identification before and after the error condition are identical. To disable mount verification, enter a command like the following: `$ MOUNT/SYSTEM/NOMOUNT_VERIFICATION MUA1: ACCOUNTS_DUE`
/OVERRIDE=*keyword*	Inhibits one or more of the access checks that the MOUNT command performs. For example: `$ MOUNT/OVERRIDE=IDENTIFICATION MFA0:` This command overrides the volume identification field, thus mounting a magnetic tape on MFA0: without a label specification.
/OWNER_UIC=*uic device-name*	Requests that the specified UIC be assigned ownership of the volume while it is mounted, overriding the ownership recorded on the volume. Or, if you are mounting a volume using the /FOREIGN qualifier, requests an owner UIC other than your current UIC.
/PROCESSOR=*keyword*	For magnetic tapes and Files–11 Structure Level 1 disks, requests that the MOUNT command associate an ancillary control process (ACP) to process the volume. You must have the operator user privilege OPER to use the /PROCESSOR qualifier. Keywords are the following: • UNIQUE For magnetic tape and Files–11 Structure Level 1 disks, creates a new process to execute a copy of the default ACP image for the specified device type or controller. For Files–11 Structure Level 2 disks, allocates a separate block cache. • SAME:*device* For magnetic tape and Files–11 Structure Level 1 disks, uses the same ACP process currently being used by the device specified. For Files–11 Structure Level 2 disks, takes the block cache allocation from the specified device. • *filespec* Creates a new process to execute the ACP image specified by the file specification (for example, a modified or a user-written ACP). You cannot use wildcard characters or node and directory names in the file specification. To use this keyword, you must have CMKRNL and OPER privileges. The /PROCESSOR qualifier causes MOUNT to override the default manner in which ACPs are associated with devices. For example: `$ MOUNT/PROCESSOR=SAME:MTA1: MFA0:` This command directs MOUNT to mount a magnetic tape on MFA0: using the same ACP process currently associated with the MTA1: device.
/PROTECTION=*code device-name*	Specifies the protection code to be assigned to the volume for the duration of the mount. See Section 8.4.2 for details.

(continued on next page)

Table 8–13 (Cont.) MOUNT Command Qualifiers for Mounting Tapes

Qualifier	Description
/RECORDSIZE=*n* *device-name*	Specifies the number of characters in each record of a magnetic tape volume. Use this qualifier when you mount a volume that has a file without a second header label (such as RT–11 volumes), or when you mount volumes with the /FOREIGN qualifier, to provide RMS with the size of fixed-length records or the maximum size of variable-length records.

Two other qualifiers that are important for mounting tape volumes are /INITIALIZE and /AUTOMATIC, which are explained in Section 8.8.2.2 and Section 8.8.2.3, respectively.

Example

```
$ MOUNT MU: TEST_FILES
%MOUNT-I-OPRQST, Please mount volume TEST_FILES in device _MUA2:
%MOUNT-I-MOUNTED, TEST_FILES mounted on _MUA2:
```

In this example, the MOUNT command requests an available RA90 device for the volume labeled TEST_FILES. After you physically mount the volume in the device named in the response from MOUNT, the system completes the operation. Note that the device is automatically allocated by MOUNT.

Upon successful completion of the operation, MOUNT notifies you with a message sent to SYS$OUTPUT. If the operation fails for any reason, MOUNT notifies you with an error message.

8.5.3 Assisting Users in Mounting Volumes

Large sites often have operators assigned to assist users with mounting volumes. Section 2.4.5 explains how users can send requests to operators. Section 2.4.6 briefly explains how operators reply to those requests.

When a user requests you to mount a specific disk or tape on a device, the following type of message appears on the operator terminal:

```
%%%%%%%%%%  OPCOM  <dd-mmm-yyyy:hh:mm:ss:cc>  %%%%%%%%%%
request <request-id>, from user <user-name>
```

The following steps indicate the sequence of events:

1. A user requests that you mount the volume TEST_FILES on the device DUA2: by entering the following command:

   ```
   $ MOUNT DUA2: TEST_FILES/COMMENT="Shelf slot 6B"
   ```

2. OPCOM notifies you of the request by displaying a message similar to the following at the operator terminal:

   ```
   %%%%%%%%%%  OPCOM, 28-MAY-1995 15:47:50.26  %%%%%%%%%%
   request 5, from user MALCOLM
   Please mount volume TEST_FILES in device _DUA2:
   Shelf slot 6B
   ```

3. Once you receive the request, OPCOM delivers a confirmation to the user, in a format similar to the following:

   ```
   %MOUNT-I-OPRQST, Please mount volume TEST_FILES in device _DUA2:
   Shelf slot 6B
   ```

4. After you locate the volume and place it on the device, OPCOM notifies the user that the volume is on the device and that the task is complete:

```
%MOUNT-I-MOUNTED,TEST_FILES mounted on _DUA2:
%MOUNT-I-RQSTDON, operator request canceled
-- mount completed successfully.
```

Instead of requesting a specific hardware device, such as DUA2:, for mounting a volume, users can make a generic MOUNT request. A generic MOUNT request specifies a type of device and lets you find an available device in that class. For example, to mount the volume CITIES on any tape drive whose name begins with MU, the user enters the following command:

```
$ MOUNT MU: CITIES/COMMENT="Slot 12c"
```

If the user has already allocated a drive whose name begins with MU, the Mount utility requests that you mount CITIES on that particular drive. If no device has been allocated, the Mount utility allocates the first available MU tape drive it finds and requests you to mount CITIES on that drive.

Sending Messages Back to Users

After you mount a disk or tape, follow these steps:

1. Use the Operator Communication Manager (OPCOM) to communicate with system users. OPCOM is a system process that receives input from a process that wants to inform an operator of a particular status or condition; OPCOM passes the message to the operator, and tracks the message.

   To use OPCOM, you must use a terminal that has been designated as an operator terminal. See Section 2.4.4 for instructions.

2. Enter the REPLY command in one of the following forms:

REPLY Command Qualifiers	Use
/ABORT=*identification-number ''message-text''*	Use to indicate that the user request is canceled. (The user's MOUNT command exits with an error status.)
/PENDING=*identification-number ''message-text''*	Use when the request has been put in a wait state until it can be completed. This command implies that the originating request was either a REQUEST/REPLY or a MOUNT command. The user cannot enter other commands until the operator fulfills or aborts the request.
/TO=*identification-number ''message-text''*	Use when the request is fulfilled. (Processing continues.)

If a user enters a MOUNT/ASSIST command and the desired device is unavailable, you can substitute another device. Whenever you must substitute a device, load the requested volume on the alternate device and prepare the device for connection before you enter the REPLY command. Use the following format:

REPLY/TO=identification-number "SUBSTITUTE device-name"

You can abbreviate the word SUBSTITUTE to ''S'' and use uppercase or lowercase characters. After a space, use the remainder of the message-text space to name the substituted device.

Examples

The following example shows how an operator redirects the mount operation from the DUB2: device to the DUA1: device:

```
$ REPLY/TO=24 "SUBSTITUTE DUA1: FOR DUB2:"
```

Following is an example of a user's request and the substitution information the user receives:

```
$ MOUNT/ASSIST  MKB500:  MYDATA
%MOUNT-I-OPRQST, Please mount volume MYDATA in device _MKB500:
%MOUNT-I-OPREPLY, Substitute MKA100:
11:44:28.71, request 1 was completed by operator _FTA8:
```

In this example, the MKA100: device has been substituted for the MKB500: device.

See the *OpenVMS DCL Dictionary* for a complete list of REPLY qualifiers and their functions. See Section 8.8 for instructions for entering REPLY commands after you mount a volume set with automatic switching disabled.

8.5.4 Mounting a Volume with Protected Subsystems

Security is usually based on control rights that are granted or denied to the user. In a protected subsystem, however, security is based on access controls assigned to the subsystem. The subsystem acts as a gatekeeper that grants or denies users access to objects belonging to the subsystem.

Unprivileged users can build and manage protected subsystems. You must be involved at two points in the process:

- To create the necessary identifiers for the subsystem. See the *OpenVMS Guide to System Security* for details.
- To mount the volume with the protected subsystem, which is explained in the next section.

Caution

Anyone who mounts a subsystem is responsible for knowing what is on the volume being mounted. Digital strongly recommends that you find out what is on a volume before you mount a subsystem. Without this knowledge, you might inadvertently subvert system security and jeopardize the privacy of users' data.

For example, a user with malicious intent who has privileges on one VMScluster node might place an application with a subsystem identifier on a volume and then request an unsuspecting operator or system manager to mount the volume on another node. Because the application has a subsystem identifier, the application appears to belong to a subsystem for which it is unauthorized.

Enabling Protected Subsystems on a Trusted Volume

The system enables protected subsystems by default only on the system disk. For other disks, you must enable subsystems every time you mount a volume. A person with the SECURITY privilege can enable subsystems on a volume by using the /SUBSYSTEM qualifier on the MOUNT command.

You can dynamically turn on and off the processing of Subsystem ACEs with the DCL command SET VOLUME/SUBSYSTEM. This command is especially useful for the system disk, which is not mounted using the MOUNT command.

Example

The command in the following example mounts the volume labeled DOC on the DUA0: device. Subsystems on the volume are accessible. The MOUNT command also assigns the logical name WORK8.

```
$ MOUNT/SUBSYSTEM/SYSTEM DUA0: DOC WORK8
```

8.5.5 Modifying Disk Volume Characteristics

Use the DCL command SET VOLUME to modify the characteristics of one or more mounted Files–11 disk volumes. To use this command, you must have write access to the index file on the volume. If you are not the owner of the volume, you must have either a system UIC or the user privilege SYSPRV. You must then specify the name of one or more mounted Files–11 volumes.

The following examples illustrate how you can use the SET VOLUME command.

Examples

1. `$ SET VOLUME/DATA_CHECK=(READ,WRITE) DKA100`

   This command requests that data checks be performed following all read and write operations to the DKA100: volumes.

2. `$ SET VOLUME/LABEL=LICENSES DKA100:`

   This command encodes the label LICENSES on the DKA100: volume. Note that, if characters in labels are entered in lowercase, the /LABEL qualifier changes them to uppercase.

8.5.6 Speeding Up Disk Mounting

The DISKMOUNT.C program can help to speed up disk mounts at system startup time. The program reduces the MOUNT image activation time by directly calling the $MOUNT system service.

Note

DISKMOUNT.C does *not* support mounting of disks connected to an Infoserver, disks served using DFS, or stripe sets.

This program requires a VAX C compiler. Perform the following steps:

1. Copy the files DISKMOUNT.H, DISKMOUNT.C, and DISKMOUNT_CHILD.C in SYS$EXAMPLES to a directory.
2. Define a logical name ‘‘SRC$’’ that points to this directory.
3. Assemble the DISKMOUNT.C and DISKMOUNT_CHILD.C files.
4. Link DISKMOUNT.OBJ and DISKMOUNT_CHILD.OBJ to produce the DISKMOUNT.EXE and DISKMOUNT_CHILD.EXE executable image files.
5. Copy these executable images to a directory, preferably SYS$MANAGER on the target system.

For additional information, see the comments in the DISKMOUNT.H file.

8.6 Setting Up Disk Volume Sets

The following sections discuss concepts related to disk volume sets and explain how to do the following:

Task	Section
Create a disk volume set from new volumes	Section 8.6.2
Create a shadowed disk volume set	Section 8.6.3
Create a disk volume set from an existing volume	Section 8.6.4
Add volumes to a disk volume set	Section 8.6.5

8.6.1 Understanding Disk Volume Sets

A **volume set** is a collection of disk volumes bound into a single entity by the DCL command MOUNT/BIND. To users, a volume set looks like a single, large volume. Volume sets have the following characteristics:

- Files are automatically located anywhere on the volume set that space is available.
- Disk quotas are enforced over the entire set.
- A single directory structure covers the whole volume set.

Use a volume set to provide a large, homogeneous public file space. You must use a volume set to create files that are larger than a single physical disk volume. (The file system attempts to balance the load on the volume sets, for example, by creating new files on the volume that is the least full at the time.)

If you want several distinct areas of file storage, with different types of users or different management policies, you must use a separate volume or volume set for each area. For example, you might want one volume for permanent user storage, with limited disk quotas and regular backups. You might want another volume for ‘‘scratch’’ use, which means that the volume has liberal or no quotas and is not backed up; also, its files are purged on a periodic basis. Each separate volume or volume set must contain a top-level user file directory for each user who keeps files on that volume.

An advantage of separate volumes is their modularity. If one of the drives holding a volume set is out of service, the whole volume set is unavailable because of its interconnected directory structure. When a drive holding a single volume is not functioning, only the files on that volume are not available.

A disadvantage of volume sets is the large size of an image backup of a multivolume set, which might affect your backup schedule. For example, if backing up each of five separate volumes takes 5 hours in the evening, backing up these same volumes in a volume set will take 25 hours, which cannot be done overnight, thus possibly causing a scheduling problem.

8.6.1.1 Guidelines for Creating Disk Volume Sets

When planning disk volume sets, keep in mind the following:

- You can turn any single volume into a volume set by binding it with a newly initialized volume. Likewise, you can add another newly initialized volume to an existing volume set.

- Do not include a system disk in a volume set.

Caution

Do not make the system disk part of a volume set because updates, upgrades, and optional product installations do not install correctly, and the operating system will no longer boot successfully.

- You cannot bind two existing separate volumes containing files into a volume set. (The MOUNT command appears to let you do this, but the result is not a coherent volume set.)
- Enter the MOUNT/BIND command only once to bind a volume set; thereafter, the volume set association is recorded on the volumes. (See Section 8.5 for details.)
- Once you bind two or more volumes into a volume set, you cannot separate them. The only way to separate a volume set is to use the Backup utility (BACKUP) to copy sets of directories selectively. (See Section 10.13 for more information.)

When you mount a disk volume set, the volume label specified in the list must correspond to a device name in the same position in the device name list.

You can bind two or more disk volumes into a volume set. The first volume in the set is called the **root volume**. Each volume in the set is identified by a volume number relative to the root volume, which is always relative to volume 1.

A disk volume set has a single directory structure. The master file directory (MFD) is on the first volume in the set.

When a disk volume set is on line and mounted, you can access all files and directories in the set by specifying either of the following:

- Device name of the device on which the root volume is mounted
- Logical name assigned to the volume set when it was mounted

8.6.1.2 Using the /BIND Qualifier

You use the /BIND qualifier with the MOUNT command to create a disk volume set in the following format:

```
MOUNT/BIND=volume-set-name
```

where:

volume-set-name Specifies a 1- through 12-alphanumeric-character name identifying the volume set.

The volume set name must be different from all volume labels within the set, and all labels in the set must be unique.

When you use /BIND, the qualifier identifies a volume set by assigning it a volume set name that applies to all volumes in the set. The qualifier also identifies the root volume and creates the directory structure for the volume.

When you create files on a volume set, the file system allocates space for the files anywhere on the set, wherever the most space exists. When existing files on any volume are extended, extension occurs on the same volume, unless the volume is physically full.

You can add new volumes to a volume set whenever additional space is needed. You can, for example, bind all disk volumes that are mounted into a volume set on a daily basis. Since this set contains all user file directories, users do not need to specify device names in file specifications to access files on any volume in the volume set. In fact, the physical location of a file is of no concern to users of the system.

Note

Do not bind your system disk into a volume set. System software updates and optional product installations do not support volume sets. If certain system files move or extend to other volumes in the set, the system might fail to boot.

You do not need special privileges to create volume sets. However, you must have write access to the index file on all volumes you are attempting to bind into a volume set; this usually means you also must have a system UIC, have the user privilege SYSPRV, or be the owner of the volumes.

You can create a disk volume set from newly initialized volumes, or you can create a volume set by extending an existing volume that already contains a directory structure and files. You can also add volumes to an existing volume set. The next three sections contain examples of how to create and mount each type of disk volume set.

8.6.2 Creating a Disk Volume Set from New Volumes

To create a disk volume set from new disk volumes, follow these steps:

1. Allocate the necessary devices and physically load the volumes.
2. Initialize each volume in the set.

   When you initialize volumes for a volume set, you can use qualifiers with the INITIALIZE command to define the volume ownership and protection. Protection and ownership information is obtained from the root (first) volume. The protection and ownership of the other volumes is ignored.
3. Enter the MOUNT/BIND command to create the volume set. The MOUNT /BIND command both creates the volume set and mounts the volumes. When this command completes successfully, all volumes in the set are ready for use; in other words, you can now create user file directories.
4. Use the /BIND qualifier only once to create the volume set. Subsequently, you can mount the volume set with a single MOUNT command.

Examples

1.
```
$ INITIALIZE DUA1: PAYVOL1
$ INITIALIZE DUA2: PAYVOL2
$ INITIALIZE DUA3: PAYVOL3
$ MOUNT/BIND=MASTER_PAY DUA1:, DUA2:, DUA3: PAYVOL1,PAYVOL2,PAYVOL3
```

   This example assumes that the volumes to be bound contain no files or data. The INITIALIZE command initializes each volume in the set. The MOUNT /BIND command defines the volume set name, MASTER_PAY, and defines the relative volume numbers of the volumes PAYVOL1, PAYVOL2, and PAYVOL3.

The order of the device names corresponds to the volume labels specified: PAYVOL1 must be physically loaded on DUA1:, PAYVOL2 on DUA2:, and PAYVOL3 on the DUA3: device.

PAYVOL1, which is listed first in the list of labels, becomes the root volume of the set. The master file directory (MFD) for PAYVOL1 contains the directory structure for the entire volume set.

2. `$ MOUNT DUA1:,DUA2:,DUA3: PAYVOL1,PAYVOL2,PAYVOL3`

   This example illustrates the use of one MOUNT command to mount a previously created volume set.

8.6.3 Creating a Shadowed Disk Volume Set

The following example illustrates one way to create a shadowed volume set.

```
$ MOUNT/BIND=TEST3013 DSA3011/SHADOW=($1$DUA402:,$1$DUA403:),
DSA3012/SHADOW=($1$DUA404:,$1$DUA405:) TEST3011,TEST3012 TEST3013
```

This command creates a volume set with the logical name TEST3013. The volume set TEST3013 is shadowed, and each element of the shadowset (TEST3011 and TEST3012) is itself a volume set.

8.6.4 Creating a Disk Volume Set from an Existing Volume

To create a disk volume set from an *existing* volume and a *new* volume, follow these steps:

1. Use the DISMOUNT command to dismount the *existing* volume. Use the /NOUNLOAD qualifier to logically dismount the volume from the drive. (The volume, however, remains physically loaded on the drive.)
2. Use the INITIALIZE command to initialize the *new* volume, specifying the device on which the volume is to be mounted and the volume label.
3. Use the MOUNT/BIND command to bind the *new* volume to the *existing* volume, specifying the volume set name, the devices on which the volumes are mounted, and the volume labels.

   Specify the volume label of the *existing* volume first; it becomes the root volume of the set. When you create a volume set from an *existing* volume, you must specify the label of the *existing* volume first because the file system must build on the *existing* directory structure.

Example

The following example shows how to create a disk volume set (called USERS) from an existing volume. The example assumes that the volume USERFILES already contains a directory structure and files and that the volume is currently located on the DUA1: device:

```
$ DISMOUNT/NOUNLOAD DUA1:
$ INITIALIZE DUA2: USERFILES2
$ MOUNT/BIND=USERS DUA1:, DUA2: USERFILES, USERFILES2
```

In the MOUNT/BIND command, you must specify the existing volume label USERFILES before the volume label USERFILES2. USERFILES will be the root volume of the set.

Caution

If you attempt to create a volume set from two or more volumes that already contain files and data, the file system does not issue an error message when you enter the MOUNT/BIND command. However, the volumes are unusable as a volume set because the directory structures are not properly bound.

8.6.5 Adding Volumes to an Existing Disk Volume Set

You can add volumes to an existing volume set at any time. The maximum number of volumes in a volume set is 255.

This section contains examples that show how to add volumes to an existing volume set.

Examples

1.
```
$ INITIALIZE DUA4: PAYVOL4
$ MOUNT/BIND=MASTER_PAY DUA4: PAYVOL4
```

   This example assumes that the volume set named MASTER_PAY is on line and mounted and has volumes named PAYVOL1, PAYVOL2, and PAYVOL3.

   The MOUNT/BIND command binds the volume PAYVOL4 with the existing volume set and makes the volume ready and available for use. Note that, if the volume set MASTER_PAY is mounted with the /SYSTEM, /GROUP, or /SHARE qualifier, the MOUNT/BIND command that adds a volume to the set must also specify the appropriate qualifier.

   When you add a volume to an existing set, the only volume in the set that you must mount is the root (first) volume, relative volume 1 (in this example, DUA4:). Mounting any of the other volumes is not necessary.

2.
```
$ INITIALIZE DUA4: PAYVOL4
$ MOUNT/BIND=MASTER_PAY DUA1:, DUA2:, DUA3:, DUA4: -
_$ PAYVOL1, PAYVOL2, PAYVOL3, PAYVOL4/SYSTEM
```

   This example assumes a set named MASTER_PAY already exists, with volumes named PAYVOL1, PAYVOL2, and PAYVOL3.

   You can add a volume to a set at the same time that you mount the volume set, as this example shows. Note that the first device/volume pair listed in the MOUNT/BIND command is the root (first) volume of the set, the DUA1: volume. When you add a volume to a set while mounting the set, you must list the root volume first.

8.7 Mounting ISO 9660 Volume Sets and Groups

To access an ISO 9660-formatted CD–ROM, you can mount disk volumes in two ways:

- Directly, using the Mount utility (MOUNT)
- Indirectly, using the DCL command MOUNT

The Mount utility (MOUNT) builds the I/O database structures that are needed to access ISO 9660 directories and files. MOUNT also verifies the presence of an appropriate ACP to perform $QIO functions specific to ISO 9660. Currently, you cannot mount ISO 9660 media as a system disk.

See the *OpenVMS System Management Utilities Reference Manual* for details.

8.7.1 Mounting ISO 9660 Volume Sets

ISO 9660 supports volume sets of up to 65,535 volume set members. At any one time, users can mount a 255-member subset of the total volume set of 65,535.

If your volume set is greater than the number of CD–ROM readers available to you, you can swap volume set members, for example, as you might when you have a single reader with multiple volume set members.

8.7.2 Mounting ISO 9660 Volume Groups

A volume group consists of one or more consecutively numbered volumes within a volume set. Affinity between the members of a volume group is established by the fact that the volumes are recorded together and are subject to the same *maximum-volume-set-size* parameter.

Each volume in a volume group contains information describing all the files and directories recorded on all of the volumes in the volume set, up to and including the members of its volume group. For example, assume that a volume set includes two volume groups:

- The first group includes Volumes 1 and 2, which were recorded together, prior to the second group.

  Volumes 1 and 2 each contain information only about their volume group; they have no information about the volumes in the second volume group.

- The second group includes Volumes 3, 4, and 5, which were recorded together at a later date.

  Volumes 3, 4, and 5 each contain information about all of the volumes in the volume set, including Volumes 1 and 2.

How to Perform This Task

When you mount a volume set, you must first mount a member of the highest-numbered volume group (the most recently recorded group—in the example, Volume 3, 4, or 5), because only a member of the highest-numbered group has the information needed to mount all members of the volume set.

If you do not follow this requirement, you must dismount all of the volumes and start again by specifying a member of the highest-numbered volume group as the first volume to be mounted.

8.7.3 Handling Partially Mounted ISO 9660 Volume Sets

OpenVMS systems support partially mounted ISO 9660 volume sets. Data is usually read from all mounted volumes in a manner that is transparent to the user program.

When a volume-set member is not mounted because the volume set is partially mounted, OPCOM sends a message to the OPERATOR class DISK requesting that the unmounted volume be mounted. If you do not honor the request within a specified time period, or if you do not enable the option to provide for dynamically mounting a volume, the I/O process fails, and an error message is issued.

8.7.4 Handling ISO 9660 Restrictions

Table 8–14 describes problems and restrictions that apply to OpenVMS support of the ISO 9660 standard and explains how to resolve them.

Table 8–14 ISO 9660 Restrictions

Media Affected	Description and Resolution
Volume Labels	These can contain from 1 to 32 characters. The first 12 characters are used to produce a unique volume identity. If the label is not unique within the first 12 characters, the volume will not mount and the following error message is displayed: `%SYSTEM-F-VOLALRMNT, another volume of the same label already mounted` **How to resolve this problem:** Mount the volume specifying a different volume label and use the OVERRIDE=IDENTIFICATION command qualifier. This will override the volume's label so as not to conflict with an already-mounted volume's label.

Volume Set Labels	These can be from 1 to 128 characters in length. The first 12 characters are used to produce a unique volume set identity. If the volume set label is not unique within the first 12 characters, the volume will not mount and one of the following error messages will be displayed:
	`%SYSTEM-F-VOLINSET, volume is already part of another volume set`
	`%MOUNT-F-DUPRVN, duplicate volume number already mounted`
	How to resolve this problem:
	Mount the volume specifying a new volume set label with the /BIND=*volume-set-label* command qualifier.

Volume Label and Volume Set Label Duplication	The first 12 characters of both the volume label and the volume set label are used to produce different lock manager resource names, which are then used to coordinate volume and volume set associations. If both the volume label and the volume set label are the same (within the first 12 characters, including null labels), a lock manager deadlock error occurs and the following error message is displayed: `%SYSTEM-F-DEADLOCK, deadlock detected` **How to resolve this problem:** Mount the volume specifying a different volume label and use the /OVERRIDE=IDENTIFICATION command qualifier. This will override the volume's label so as not to conflict with the volume set's label.
Undefined Record Format Errors	Many ISO 9660 CD–ROMs are mastered without a specified record format because the ISO 9660 media can be mastered from platforms that do not support the semantics of files containing predefined record formats. OpenVMS file system utilities (such as TYPE and COPY), language RTLs, and applications that use RMS for record access may report RMS errors, utility errors, and language errors when accessing files whose record format is undefined or appears illegally specified. **How to resolve this problem:** Use the following command syntax at mount time to force all files of type UNDEFINED to the STREAM record format having a maximum record length of 512 bytes: `MOUNT/MEDIA=CDROM/UNDEFINED=(STREAM:512) device label` For more information about RMS record formatting, see the *OpenVMS Record Management Utilities Reference Manual* and the *OpenVMS Record Management Services Reference Manual*.

8.8 Mounting Tape Volume Sets

The procedure for mounting a tape volume set is similar to the procedure for mounting a single tape volume, described in Section 8.5. The number of volume identifiers does not need to equal the number of device names you specify. In other words, when you mount a tape volume set, you can specify more volume identifiers than device names or more device names than volumes.

The number of devices you specify directly affects the action taken by the tape file system when processing continuation volumes in a volume set. For example, when the number of devices is greater than the number of volumes, the tape files system requests a continuation volume to be mounted on the first drive from the list that does not have a volume mounted.

When mounting a volume set, make sure that all the volumes in the set contain write rings if the user intends to write to any of the volumes in the set. (If even one of the volumes in the set does not contain a write ring at mount time, all volumes are write-locked; the system is unable to write to any of them.) Load the volumes on the drives that have been allocated and place the drives on line.

The following sections explain how to perform these tasks:

Task	Section
Create a tape volume set	Section 8.8.1
Mount continuation volumes in a volume set	Section 8.8.2
Mount volume sets with automatic switching disabled	Section 8.8.2.3

8.8.1 Creating a Tape Volume Set

If you do not create a volume set explicitly, the operating system creates one when necessary. If you have not mounted a volume set and a continuation volume is required, the tape file system requests that a continuation volume be mounted and implicitly creates a volume set. For example, if the tape file system encounters an EOT mark while writing a volume, it sends a message to the operator console requesting that another volume be mounted.

After you mount the next volume, the tape file system writes the volume and header labels and then reissues the pending write requests to the continuation volume. The file-set identifier in the first file-header label of all files written to the continuation volume is the file-set identifier of the first file on the first volume. The file-set identifier for volume sets is always that of the first file of the first volume that is mounted in the set.

How to Perform This Task

To explicitly create a volume set with three volumes, for example, follow these steps:

1. Allocate devices on which you will load the volumes.
2. Initialize the volumes. Specify the density and the access protection in addition to the device name and the volume identifier in the INITIALIZE commands.
3. Mount the volumes, including the device names and volume identifiers. Specifying a logical name for the volume set is optional. The system not only confirms which volumes have been mounted, but also indicates on which drive each volume has been mounted.

The system mounts and verifies only the volumes that are physically loaded on the devices at mount time. However, the volume identifiers of additional volumes that you specify are not verified until the volumes are accessed.

4. You can check the densities, volume labels, UICs, and relative volume numbers of the volumes that are mounted on devices. To do so, specify the SHOW DEVICES/FULL command. If you specify a generic device code for the tape drives, such as MU, information is displayed for all drives of that type configured in the system.

   To display information for a volume mounted on a specific drive, specify the physical device code, consisting of the generic device code, the controller designation, and the unit number followed by a colon.

   For more information on the SHOW DEVICES command, including examples of displays returned by the SHOW DEVICES/FULL command, see Section 7.2 or the *OpenVMS DCL Dictionary*.

Examples

1.
```
$ ALLOCATE MUA0:
%DCL-I-ALLOC, _MARS$MUA0: allocated

$ ALLOCATE MUA1:
%DCL-I-ALLOC, _MARS$MUA1: allocated

$ ALLOCATE MUA2:
%DCL-I-ALLOC, _MARS$MUA2: allocated
```

   The commands in this example allocate a drive on which you will load each volume.

2.
```
$ INITIALIZE/DENSITY=1600/PROTECTION=(G:RW) MUA0: TAPE1
$ INITIALIZE/DENSITY=1600/PROTECTION=(G:RW) MUA1: TAPE2
$ INITIALIZE/DENSITY=1600/PROTECTION=(G:RW) MUA2: TAPE3
```

   The commands in this example initialize the volumes. The commands specify the density and the access protection in addition to the device name and the volume identifier.

3.
```
$ MOUNT MUA0:,MUA1:,MUA2: TAPE1,TAPE2,TAPE3 TEST
%MOUNT-I-MOUNTED, TAPE1 mounted on _MUA0:
%MOUNT-I-MOUNTED, TAPE2 mounted on _MUA1:
%MOUNT-I-MOUNTED, TAPE3 mounted on _MUA2:
```

   The commands in this example mount the volumes. The commands include the device name and volume identifier.

8.8.2 Mounting Continuation Volumes in a Tape Volume Set

When mounting a tape volume set, follow the general procedures described in Section 8.8.1. Once you create the volume set, you do not need to initialize the volumes when you mount the volume set.

Allocating a drive for each volume in the volume set is not necessary. The tape file system requests that volumes be switched to appropriate drives when continuation volumes are required.

The operating system stores, but cannot verify, the identifiers of volumes you specify but do not physically mount on drives at mount time. The system later verifies the volume identifiers when the volumes are accessed.

The operating system supports the continuous processing of mounted volumes in a tape volume set through automatic volume switching and automatic volume labeling (AVL).

8.8.2.1 Creating Labels

Depending on the following conditions, the file system does or does not create a label:

- If the file system is writing to the volume set, it creates a label for the magnetic tape and initializes the tape with that label and the protection characteristics set for the first volume of the volume set.
- If the tape file system is reading the volume set, it tries to mount the next tape in the volume set with that label.
- If the drive has no tape loaded on it, or the wrong tape, the tape file system sends a message to the operator console notifying the operator either to mount a tape or to mount the correct tape.

Before processing continuation volumes, the tape file system processes the protection on that volume (as described in Section 8.4.2). If the file system determines that the user does not have access to the volume, it sends a message to the operator.

The label fills the six-character volume identifier field:

- Characters 1 to 4 of the field contain the first four characters of the label specified for the previous volume in the volume set. (If the label is less than four characters, the volume identifier field is padded with underscores; for example, if the volume identifier is XXX, the padded field is XXX_.)
- Characters 5 and 6 contain the relative volume number for that reel in the volume set.

Note that the system can generate only 99 unique labels for a given volume set.

With automatic volume switching enabled, the operator can load a tape on the next drive allocated to the tape volume set anytime before the volume being processed reaches the EOT mark. The tape file system mounts and initializes (if INITIALIZE was specified originally) the next tape in the volume set and then notifies the operator that the switch has occurred.

8.8.2.2 Enabling Automatic Volume Switching

To use automatic volume switching, you must allocate more than one tape drive to your volume set. After you do so, the tape file system switches volumes for you automatically by selecting the next tape drive allocated to the volume set. The tape file system expects you to load the next volume in the volume set on that drive.

Examples

1. `$ MOUNT MUA0:,MUA1:,MUA2: TAPE`

   In this example, the volume with the identifier TAPE is mounted on the MUA0: drive. Load continuation volumes for this set on the tape drives in the following order: MUA1:, MUA2:, MUA0:, MUA1:, MUA2:, and so forth.

2.
```
$ INITIALIZE MUA0: MAIN
$ MOUNT/OVERRIDE=IDENTIFICATION/INITIALIZE=CONTINUATION MUA0:,MUA1:
```

   This example shows the use of the /INITIALIZE=CONTINUATION qualifier for mounting volume sets. It also shows how the system creates volume identifiers for continuation volumes.

   The volume labeled MAIN is mounted on the MUA0: drive. The second volume in the set receives the volume identifier MAIN02 and is mounted on the MUA1: drive. The third volume in the set receives the volume identifier MAIN03 and is mounted on the MUA0: drive.

   To ensure that any volume added to the tape volume set is initialized prior to being written to, mount the volume with the /INITIALIZE=CONTINUATION qualifier. The default is /NOINITIALIZE.

3.
```
$ MOUNT MUA0:,MUA1: SUN
```

   In this example, the first volume in the set is labeled SUN and is mounted on the MUA0: drive. The second volume receives the identifier SUN_02 and is mounted on the MUA1: drive. The third volume receives the identifier SUN_03 and is mounted on the MUA0: drive.

4.
```
$ MOUNT MUA0:,MUA1: SUN,MOON
```

   In this example, a continuation volume with two volume identifiers, SUN and MOON, is mounted on MUA0: and MUA1:, respectively. If a third volume is added to the set, it is given the identifier MOON03 and is mounted on the MUA0: drive.

8.8.2.3 Disabling Automatic Switching

If your site prelabels volumes, you must disable automatic volume switching to avoid overwriting these labels. To explicitly override automatic volume switching, specify the /NOAUTOMATIC qualifier when mounting a tape volume. (The default is /AUTOMATIC.) Note that if you allocate only one drive to the tape volume set, automatic volume switching is implicitly disabled.

When a user is reading or writing to a magnetic tape and the tape reaches end-of-tape position, the system suspends processing and sends a request to mount the next tape in the volume set. For example:

```
%%%%%%%%%%  OPCOM, 28-MAY-1995 15:23:31.78  %%%%%%%%%%
request 3, from user PLAW
MOUNT new relative volume 2 (DW0QT2) on MUA1:
```

The user does not see this message and might not realize that another tape is needed to complete the read or write operation.

Example

```
$ MOUNT/NOAUTOMATIC MUA0: ABCD,EFGH
```

The command in this example tells MOUNT not to supply its own label for the second volume but, instead, to use the ones specified in the MOUNT command.

8.8.2.4 Sending Messages Back to Users

After loading the continuation volume on the drive specified in the mount request, mount the volume by entering the REPLY command with one of the three qualifiers shown in Table 8–15.

Table 8–15 REPLY Command Qualifiers for Continuation Volumes

Qualifier	Description

/BLANK_TAPE=*n*	Use with an unformatted volume for write operations. This qualifier initializes the volume and requires the VOLPRO and OPER privileges to avoid a runaway tape or timeout condition. (See Section 8.1.3.2.) Either of the following REPLY commands is valid:
	`$ REPLY/BLANK_TAPE=3`
	`$ REPLY/BLANK_TAPE=3 "DW0QT2"`
	The first command does not specify a volume identifier; the second does.

/INITIALIZE_ TAPE=*n*	Use with a formatted volume for write operations if the volume identifier on the continuation volume does not match the one specified in the mount request. The file system reinitializes the tape and mounts the volume with the new volume identifier. The tape file system then performs access checks and initializes the volume as if the INITIALIZE command had been specified. Any data on the tape prior to specifying the /INITIALIZE_TAPE command is lost. Either of the following REPLY commands is valid:
	`$ REPLY/INITIALIZE_TAPE=3`
	`$ REPLY/INITIALIZE_TAPE=3 "DW0QT2"`
	The first command does not specify a volume identifier; the second does.

/TO=*request_id [volume identifier]*	Use with a formatted volume for both read and write operations. During a write operation, use the /TO qualifier if you want the volume identifier that is specified in the mount request to be written on the continuation volume.
	For example, to respond to the mount request 3, mount volume DW0QT2 on drive MTA1: and enter one of the following commands:
	`$ REPLY/TO=3`
	`$ REPLY/TO=3 "DW0QT2"`
	The first REPLY command does not specify a volume identifier; the second does.

Specifying the Volume Identifier with the MOUNT Command

Specifying the volume identifier in the MOUNT command is essential during write operations because it ensures that the correct volume is mounted on the drive and links the continuation volume to the volume set.

Omitting the Volume Identifier with the REPLY/TO Command

To preserve the accessibility character on a volume, you must omit the volume identifier with the REPLY/TO command during a write operation. (When you read from tape, the volume identifier is optional.)

If you initialize and mount a volume set in which each volume has a unique accessibility character that you want to maintain, avoid using the volume identifier because it causes the accessibility character of the first volume in the set to overwrite the accessibility character on the continuation volume.

For example, to preserve the accessibility character, enter the following command in which 3 is the request identification number:

```
$ REPLY/TO=3
```

Once the tape file system receives the REPLY command, the system performs checks on the continuation volume to ensure that the volume is the correct one. If it is the correct volume with proper access codes, the system mounts the volume and reissues pending read or write requests to the continuation volume. If the volume fails any of these access checks, the system does not mount the volume (or initialize and mount it in the case of a blank tape).

8.8.3 Modifying Magnetic Tape Characteristics

Use the DCL command SET MAGTAPE to define the default characteristics associated with a specific tape device for subsequent file operations. Use the following format:

SET MAGTAPE device-name

where:

device-name Specifies the name of the tape device for which the characteristics are to be set. The device must not be currently allocated to any other user.

The following examples illustrate uses of the SET MAGTAPE command in conjunction with the MOUNT command.

Examples

1.
```
$ MOUNT MUB1:/FOREIGN
$ SET MAGTAPE MUB1:/DENSITY=800
```

   In this example, the MOUNT command mounts a foreign tape on the MUB1: drive. The SET MAGTAPE command defines the density at 800 bits per inch for writing to the magnetic tape. (The density is reset only if the tape has never been written before.)

2.
```
$ MOUNT MUA0:/USER_VOL
$ SET MAGTAPE MUA0:/SKIP=FILES:4
```

   In this example, the MOUNT command mounts a tape called USER_VOL on the MUA0: drive. The SET MAGTAPE command directs the I/O subsystem to position the tape to skip four files.

3.
```
$ MOUNT MUA1:/FOREIGN
$ SET MAGTAPE/REWIND MUA1:
```

   In this example, the MOUNT command mounts a foreign tape on the MUA1: drive. The SET MAGTAPE command rewinds the volume.

8.9 Dismounting Volumes and Volume Sets

When you finish processing the files or data on a disk or tape volume, you can use the DISMOUNT command to explicitly dismount a single volume or an entire volume set.

Use the following format when you enter the DISMOUNT command:

DISMOUNT device-name

where:

device-name — Name of the device containing the volume—either a logical name or a physical name. If you specify a physical name, the controller defaults to A and the unit defaults to 0.

If the volume currently mounted on the device is a member of a disk or tape volume set, all volumes in the set are dismounted unless you specify the /UNIT qualifier.

You can dismount a volume on a local node or on all the nodes throughout a cluster.

Before dismounting a volume or volume set, the DISMOUNT command checks for conditions that prevent the dismount from completing:

- Installed swap and page files
- Installed images
- Devices spooled to the volume
- Open user file (any files not falling into one of the first three groups)

If none of these conditions is found, the volume is marked for dismount. If any of these conditions exists, the DISMOUNT command does not mark the volume for dismount but, instead, displays error messages indicating the conditions that exist, the number of instances of each condition, and the fact that the volume cannot be dismounted.

If you attempt to dismount the system disk after it has been mounted shared, you may see a message such as the following, even if there are no user files open:

```
%DISM-W-CANNOTDMT, AXP27$DKA300: cannot be dismounted
%DISM-W-USERFILES, 1 user file open on volume
```

The message occurs because the file DISMOUNT.EXE is opened as a ‘‘user’’ file in the course of the dismount operation. To eliminate the error message, install the file DISMOUNT.EXE.

In some cases, you might want to mark a volume for dismount even though files are open on the volume. Marking the volume for dismount prevents users from opening any new files, thereby allowing activity to wind down. You can use the qualifier /OVERRIDE=CHECKS to mark the volume for dismount even if files are open.

Dismounting with Cached Information

As a performance enhancement, the system stores volume information in memory, including information about free space on a disk volume, file identifications, quota file entries, and file headers. This storing of information is called **caching**. Cached information can include blocks allocated but not yet in a file, or files created but not yet in a directory.

The system writes the information in the caches to the disk when you dismount the disk or shut down the system. If you remove a disk from a drive before the caches are written to disk, the information in the caches is lost. Therefore, you must follow these guidelines:

- Avoid write-locking a volume while it is mounted.
- Do not remove a volume from a drive before it has been dismounted.
- Do not halt the system without performing an orderly shutdown procedure (see Section 4.8.1).

You cannot dismount a volume if any known file lists associated with the volume contain entries. If a volume is referenced in a known file list, you must complete the following steps before you can dismount the volume:

1. Delete all known images associated with the volume using the Install utility delete command. For more information, see Section 16.9.14.
2. Wait for:
   a. All processes using those images to release the images.
   b. The system to write writable images back to their files.

      Use the DCL command SHOW DEVICES/FILES to determine the status of the files.

The following sections explain how to perform these tasks:

Task	Section
Dismount a single volume	Section 8.9.1
Dismount a volume set	Section 8.9.2
Dismount foreign volumes	Section 8.9.3
Dismount a volume in a cluster	Section 8.9.4

8.9.1 Dismounting a Single Volume

This section explains procedures to follow in dismounting a single volume and also describes some of the qualifiers you can use with the DISMOUNT command.

8.9.1.1 Dismounting Before Unloading a Volume

Always explicitly dismount a volume or volume set with the DISMOUNT command, or with a command procedure containing that command, before physically unloading that volume. Always wait for the drive to unload before you remove the volume. (You can verify that the dismount is complete by entering the DCL command SHOW DEVICES.)

A private volume is dismounted and unloaded automatically if you log out of the job from which you mounted the volume. If the system fails, however, the drive is not automatically dismounted.

Note that data loss can occur if you do not explicitly dismount a volume and the system fails. For tape volumes, data loss can occur if you unload a volume that contains an open file for which file-trailer labels have not been written. When you remount the volume and attempt to access the file without file-trailer labels, you receive the following error message:

```
%MTACP-magnetic tape position lost
```

You can access all the files that precede the file whose file-trailer labels have not been written. However, you cannot access the file that does not have file-trailer labels.

8.9.1.2 Dismounting Allocated Devices

If the device you are dismounting was allocated with an ALLOCATE command, it remains allocated after you dismount it with the DISMOUNT command. If the device was implicitly allocated with the MOUNT command, the DISMOUNT command deallocates it.

8.9.1.3 Using DISMOUNT Command Qualifiers

Following are explanations of the /UNIT and /NOUNLOAD qualifiers.

Qualifier	Description
/UNIT	Explicitly dismounts a single volume in the volume set without dismounting the entire set. (By default, the system dismounts all the volumes in the set when you explicitly dismount a single volume in a volume set.) Using this qualifier dismounts a volume but does not ''unbind'' the volume from the volume set; if you remount the volume, it becomes part of the volume set again.
/NOUNLOAD	Overrides the default automatic unloading of your volume from the drive. With this qualifier, your volume is logically dismounted from the drive; however, the volume remains physically loaded on the drive. If you use this qualifier to dismount a tape volume, the volume remains loaded on the tape drive and the tape reel is rewound to the BOT mark. Using this qualifier can save time and eliminate unnecessary handling of a volume if you plan to remount or reinitialize a volume you are dismounting.

Example

The following example shows how to use the DISMOUNT command. The example uses the /NOUNLOAD qualifier.

```
$ DISMOUNT/NOUNLOAD MUA1:
```

In this example, the tape volume is logically dismounted and remains loaded on the MUA1: device. Also, the tape reel is rewound to the beginning-of-tape mark. The operating system returns you to DCL level.

8.9.2 Dismounting a Volume Set

You use the DISMOUNT command to dismount an entire volume set. If you explicitly dismount any volume in a disk or tape volume set, the entire volume set is dismounted. For example, if you have a volume set that consists of DUA3: and DUA4: and you enter the following command, the entire volume set is dismounted:

```
$ DISMOUNT DUA3:
```

8.9.3 Dismounting Foreign Volumes

You also use the DISMOUNT command to dismount foreign volumes. The following command dismounts a volume that has been mounted with the /FOREIGN qualifier on the DUA0: device:

```
$ DISMOUNT DUA0:
```

In this example, the volume that had been mounted with the /FOREIGN qualifier on DUA0: is dismounted and automatically unloaded. The system returns you to DCL level.

8.9.4 Dismounting a Volume in a VMScluster System

You can use the DISMOUNT command to dismount a volume throughout a VMScluster system by using the /CLUSTER qualifier. The following command, which requires SYSNAM privilege, dismounts a volume in a VMScluster system:

```
$ DISMOUNT/CLUSTER $10$DJA100:
```

The DISMOUNT/CLUSTER command first checks for conditions that prevent the volume from dismounting on the local node. If none is found, the command then checks for such conditions on all the other nodes. If a condition is found on any node, the command sends error messages identifying the device, the node on which the error occurred, and the error.

For more information on the DISMOUNT command, see the *OpenVMS DCL Dictionary*.

8.10 Using Command Procedures for Media Setup

Many of the operations that you perform on disk and tape media are routine. It is probably worthwhile to identify those routine tasks and design command procedures to assist you in performing them. To become familiar with the syntax used to design and execute command procedures, see the *OpenVMS User's Manual*.

You might, for example, want to design command procedures to set up private disk and tape volumes. The command procedure examples in this section, although general in nature, can serve as guiding strategies for you. You can tailor these command procedures to meet the needs of your own setup tasks.

8.10.1 Sample Command Procedure for Setting Up Disk Volumes

The command procedure in this section allocates, initializes, and mounts a disk volume. Follow these steps:

1. Use a text editor to create a file named SETUP.COM.
2. Enter the following command procedure, which, when executed, allocates and mounts a disk:

```
$ ! Place a disk in the drive
$ IF P1 .EQS. "" THEN INQUIRE P1 "enter device name"
$ IF P2 .EQS. "" THEN INQUIRE P2 "enter volume label"
$ IF P3 .EQS. "" THEN INQUIRE P3 "enter logical name"
$ ALLOCATE 'P1'
$ MOUNT 'P1' 'P2' 'P3'
```

   This command procedure, although very simple, accomplishes the task of allocating and mounting a disk each time you execute it. It prompts you for the device name, volume label, and logical name of the disk device that you want to allocate and mount. By assigning logical names to your disks, you can use this command procedure to allocate and mount devices repeatedly.

   You can take further advantage of the power of a command procedure by including a few additional tasks as well. For example, you might design the SETUP.COM command procedure to deallocate and dismount the disk. The command procedure example used to set up a magnetic tape (described in the next section) takes advantage of some of these options.

3. To execute the SETUP.COM command procedure, enter the following command:

```
$ @SETUP
```

8.10.2 Sample Command Procedure for Setting Up Tape Volumes

The command procedure in this section, which is more complex and detailed than the previous example, is designed to set up a magnetic tape for processing. The ALLOCATE and MOUNT/FOREIGN commands are included in this command procedure. Using a text editor, construct the command procedure as shown in Example 8–1.

Example 8–1 Command Procedure to Set Up Tape Volumes

```
$ ! First mount the tape on the drive
$ ON CONTROL_Y THEN GOTO EXIT
$ ON ERROR THEN GOTO EXIT
$ WRITE SYS$OUTPUT "Welcome to FETCH."
$ WRITE SYS$OUTPUT " "
$ L1:  INQUIRE/NOPUNC PHYS "Have you placed the volume in the drive? "
$ IF .NOT. PHYS THEN GOTO L1
$ INQUIRE/NOPUNC DRIVE "Which drive is the volume mounted on? "
$ DRIVE = DRIVE - ":"
$ ALLOCATE 'DRIVE'
$ MOUNT/FOREIGN 'DRIVE'
$ ON ERROR THEN GOTO COMMAND_LOOP
$ !
$ COMMAND_LOOP:  INQUIRE/NOPUNC OPTION "FETCH> "
$ IF OPTION .EQS. "DIR" THEN GOTO DIR
$ IF OPTION .EQS. "EXIT" THEN GOTO EXIT
$ IF OPTION .EQS. "FETCH" THEN GOTO FETCH
$ IF OPTION .EQS. "HELP" THEN GOTO HELP
$ IF OPTION .EQS. "LIST" THEN GOTO LIST
$ GOTO COMMAND_LOOP
$ !
$ DIR:  INQUIRE SPEC "Filespec"
$ DIR 'SPEC'
$ GOTO COMMAND_LOOP
$ HELP:
$ WRITE SYS$OUTPUT "Enter any of the following commands at the prompt:"
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "DIR         (To search for a file)"
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "EXIT        (To exit this program)"
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "FETCH       (To perform a BACKUP RESTORE operation)"
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "HELP        (To read this text)"
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "LIST        (To perform a BACKUP LIST operation)"
$ GOTO COMMAND_LOOP
$ !
$ FETCH:  INQUIRE FILE "Filespec"
$ INQUIRE SAVESET "Save set name"
$ LINE := BACKUP/LOG 'DRIVE':'SAVESET'/SELECT='FILE'
$ INQUIRE EXCLUDE "Enter any filespecs you want excluded"
$ IF EXCLUDE .EQS. "" THEN GOTO L2
$ LINE := 'LINE'/EXCLUDE=('EXCLUDE')
$ !
$ L2:  INQUIRE/NOPUNC TO "Where do you want the file(s)? (RET for current directory)"
$ IF TO .EQS. "" THEN GOTO REPLACE
$ LINE := 'LINE' 'TO'
$ GOTO L3
$ REPLACE:  LINE := 'LINE' []
```

(continued on next page)

Example 8–1 (Cont.) Command Procedure to Set Up Tape Volumes

```
$ !
$ L3:  INQUIRE/NOPUNC NEW "Create a new version if file already exists? "
$ IF .NOT. NEW THEN GOTO NOT
$ LINE := 'LINE'/NEW_VERSION
$ !
$ NOT:  LINE := 'LINE'/OWNER_UIC=ORIGINAL
$ LINE
$ GOTO COMMAND_LOOP
$ !
$ LIST:  INQUIRE SPEC "Filespec"
$ INQUIRE SAVESET "Save set name"
$ INQUIRE/NOPUNC OUTPUT "What do you want to call the list file? (RET for SYS$OUTPUT )"
$ IF OUTPUT .EQS. "" THEN GOTO NOOUT
$ LINE := BACKUP/LIST='OUTPUT' 'DRIVE':'SAVESET'/SELECT=('SPEC')
$ GOTO L4
$ NOOUT:  LINE := BACKUP/LIST 'DRIVE':'SAVESET'/SELECT=('SPEC')
$ !
$ L4:  INQUIRE EXCLUDE "Enter any filespecs you want excluded"
$ IF EXCLUDE .EQS. "" THEN GOT L5
$ LINE := 'LINE'/EXCLUDE=('EXCLUDE')
$ !
$ L5:  LINE
$ GOTO COMMAND_LOOP
$ !
$ EXIT:
$ DISMOUNT 'DRIVE'
$ DEALLOCATE 'DRIVE'
```

Assuming this command procedure is in a file named FETCH.COM, execute the command procedure by entering the following command:

```
$ @FETCH
```

In addition to allocating and mounting functions, as in the previous example, FETCH.COM prompts you for input. For example, it specifically asks you if the tape is on the drive. Also note that FETCH.COM does a BACKUP restore operation. It prompts you for specific options on the restore operation. Finally, FETCH.COM explicitly dismounts your magnetic tape volume and deallocates the drive after your task completes.

8.11 Managing Disk Space

Disk space available for files is finite. As system manager, you share responsibility with your users for making the best use of disk space.

The following sections explain disk quotas and describe some methods you can use to conserve and monitor disk space:

Method	Section
Establish disk quotas	Section 8.11.2
Purge files	Section 8.11.3
Set version limits on files	Section 8.11.4
Set file expiration dates	Section 8.11.5
Analyze and repair error conditions	Section 8.12

8.11.1 Understanding Disk Quotas

A **disk quota** is a method for maintaining and enforcing limits on the amount of disk space available to users on a public volume. You limit the amount of space available to individual users on public volumes (or volume sets) by creating and maintaining a **quota file** on each volume. Individual users can similarly restrict usage on private volumes.

Quotas are maintained and enforced on a per-volume basis. Each volume or volume set has its own quota file. A volume on which quotas are not maintained has no quota file. On a volume set, volume 1 contains the quota file.

With OPER privilege, you (or the user maintaining the volume) supply identifiers and assign quotas and overdrafts with the System Management utility (SYSMAN). (During normal file activities, the system automatically maintains usage counts.)

If users run out of disk space during the creation of a file, they receive a system message. If they cannot obtain sufficient space by purging or deleting unnecessary files, they might contact you to increase their disk quota. If they attempt to write a file to a spooled printer, they must have write access and have sufficient quota on the disk associated with that printer.

Disk Quota File

A **quota file** records all users who are allowed to use the disk, and shows their current disk usage and their maximum disk allocation. A quota file, QUOTA.SYS, which is stored in directory [000000] with other system files, requires one block of disk storage for every 16 entries.

A quota file has the following format:

```
   UIC ❶          Usage ❷        Permanent Quota ❸  Overdraft Limit ❹
[0,0]             0               333333             3333
[TTD,DAVIS]       15590           333333             3333
[TTD,MORGAN]      1929            333333             3333
[MKT,MORSE]       7650            333333             3333
.
.
.
```

❶ User Identification Code (UIC) of each user entitled to maintain files on the volume. UIC [0,0] appears in all quota files; use it as a template to set default values for quotas and overdrafts.

❷ Number of disk blocks currently dedicated to a user's files. This number includes the blocks allocated (shown by the DCL command DIRECTORY /SIZE=ALL /BY_OWNER=*owner*), plus at least one block in the index file for every file owned by the user.

❸ Maximum number of blocks on the volume that a user's files can occupy. When the maximum number is exceeded, the system issues an error message when a file is created.

❹ Number of blocks by which a user can exceed the quota.

Each entry in a quota file includes the information shown in Table 8–16.

Table 8–16 Contents of a Quota File

Item	Description
General Identifier or UIC	Identification code of a user entitled to maintain files on the volume
Usage	Number of blocks on the volume taken up by the user's files
Quota	Maximum number of blocks on the volume that the user's files can take up before an error message is issued
Overdraft	Number of blocks over the quota that the user's files can take up

The maximum number of blocks permitted to a user on a volume is the sum of the quota and the overdraft.

A quota file is initialized with an entry for UIC [0,0]. The usage count for this UIC should not change from 0; in other words, UIC [0,0] should own no files. Its quota and overdraft, however, serve as defaults in certain situations; set them to values most likely to be assigned to other UICs as quotas and overdrafts.

How Quotas Are Maintained

During normal use of a volume with a quota file, the system automatically updates the usage counts as users create, delete, extend, and truncate files. Users without entries in the quota file are not allowed to create files or allocate space on the volume unless they have the EXQUOTA privilege.

To create new files, a user must have disk space usage below quota (not overdraft). If adding a new file or expanding a current file exceeds a user's quota, the system prohibits the operation and issues an error message.

A user with an overdraft might be able to extend an open file after exceeding the disk quota (for example, during an editing session). A user can extend an open file until usage exceeds the sum of the quota and the overdraft. At this point, the system prohibits further extensions to the file.

Quota restrictions are not enforced for users with the EXQUOTA privilege; however, their usage counts are maintained.

How the Rebuild Operation Ensures Quota File Accuracy

When you mount a volume that was not properly dismounted the last time it was used, the system performs an automatic REBUILD operation. If quotas are enforced on the volume, this action ensures that the quota file accurately reflects usage of the disk, under any of the following conditions:

- The system fails
- The volume is physically removed before being dismounted
- A user presses the WRITE PROTECT button

8.11.2 Establishing Disk Quotas

Disk quota operations are enabled by default. However, you can use SYSMAN DISKQUOTA commands to control disk usage. You can assign disk quotas to users and maintain an accurate record of disk use for ODS Level 2 disks. You create a quota file for each disk *except the system disk*. The quota file records the current usage and the maximum disk usage for all users.

SYSMAN allows you to access disks that are normally unavailable from your local node. With SYSMAN, you can obtain a display of all disks on the other nodes, including those that are mounted privately or used as system disks. You can run DISKQUOTA on any available disk without logging in to each node.

8.11.2.1 Creating a Quota File

The first step in allocating disk space is to create a quota file for each volume or each volume set. The absolute maximum number of blocks permitted a user on a volume is the sum of the quota and the overdraft. Only users with the EXQUOTA privilege can bypass disk quota restrictions.

How to Perform This Task

Creating a quota file requires SYSPRV, BYPASS, or GRPPRV privilege. To create a quota file on a disk, follow these steps:

1. Use the SYSMAN command DISKQUOTA CREATE and specify the target disk with the /DEVICE qualifier using the following format:

   DISKQUOTA CREATE/DEVICE=device-name

2. Use the DISKQUOTA MODIFY command to adjust [0,0] to an appropriate value for the device using the following format:

   DISKQUOTA MODIFY/DEVICE=device-name/PERMQUOTA=value

   Note

   If you create a quota file or enable disk quotas on a disk that has files on it, use the DISKQUOTA REBUILD command to update the disk quota entries with the current usage information.

3. Use the SYSMAN command DISKQUOTA SHOW to display the quota file using the following format:

   DISKQUOTA SHOW owner/DEVICE=device-name

Examples

1. ```
   $ MCR SYSMAN
   SYSMAN> DISKQUOTA CREATE/DEVICE=DUA12:
   ```

   The command in this example sets up the quota file, QUOTA.SYS, in directory [000000] on the DUA12: device.

   The quota file has one entry, UIC [0,0], that stores default values for quotas and overdrafts.

2. ```
   SYSMAN> DISKQUOTA MODIFY/DEVICE=DUA12: [0,0]/PERMQUOTA=3000
   ```

   The command in this example edits the entry for UIC [0,0] in the quota file on the DUA12: device, setting the default permanent quota to 3000 blocks.

3. ```
   SYSMAN> DISKQUOTA SHOW [0,0]/DEVICE=DUA12:
   ```

   The command in this example shows quotas, overdrafts, and usage counts for UIC [0,0] on the DUA12: device.
   ```

8.11.2.2 Monitoring Disk Quotas

You must monitor the amount of disk space users consume.

Command	Description
MOUNT/QUOTA	Use to enforce quotas on a specified disk volume. To use MOUNT/QUOTA, you must have the VOLPRO user privilege, or your UIC must match the UIC written on the volume.
SHOW QUOTA	Use to determine whether a quota exists for any specific user on a specific disk. The display that results from the SHOW QUOTA command gives the quotas used, authorized, and available. Enter the DCL command SHOW QUOTA using the following format: SHOW QUOTA/USER=uic (or identifier) The results of the SHOW QUOTA command depend on whether you have read access to the quota file: • If you have read access, the command shows how much disk space any user on the system has been allocated. • If you do not have read access, the command shows your own allotment.

Examples

1.
```
$ SHOW QUOTA
User [DOCUMENTATION,MALCOLM] has 2780 blocks used, 7220 available,
of 10000 authorized and permitted overdraft of 500 blocks on DISK$
```

   The SHOW QUOTA command displays the amount of disk space authorized, used, and still available on the current default disk for the present user. The permitted overdraft in this example is 500 blocks.

2.
```
$ SHOW QUOTA/USER=[DOCUMENTATION,JONES]/DISK=XXX1:
%SYSTEM-F-NODISKQUOTA, no disk quota entry for this UIC
```

   This SHOW QUOTA command shows that the user with UIC [DOCUMENTATION,JONES] has no disk quota allocation on the XXX1: device.

3.
```
$ SHOW QUOTA/USER=[DOCUMENTATION,ELAINE]
User [DOCUMENTATION,ELAINE] has 27305 blocks used, 2305 OVERDRAWN,
of 25000 authorized and permitted overdraft of 4000 blocks on DISK$
```

   This SHOW QUOTA command indicates that a user has an overdrawn quota.

8.11.2.3 Suspending Quota Operations

The SYSMAN command DISKQUOTA DISABLE (which requires SYSPRV, BYPASS, or GRPPRV privilege), suspends quota operations on a volume; the DISKQUOTA ENABLE command lifts the suspension. In addition, you can suspend quota operations on a volume at mount time by specifying the /NOQUOTA qualifier with the DCL command MOUNT. Disabling quotas requires privileges.

Whenever quotas are enabled on a volume—either implicitly with the MOUNT command or explicitly with the DISKQUOTA ENABLE command—you must update disk quota information using the command DISKQUOTA REBUILD. In updating the quota file, the system adds new UICs and corrects usage counts for each user. (See the *OpenVMS System Management Utilities Reference Manual* for more information.)

How to Perform This Task

To discontinue quota operations on a volume, follow these steps:

1. Log in to SYSMAN.
2. Use the following format to execute the DISKQUOTA DISABLE command on each member of the cluster that is mounting the volume:

   SYSMAN> DISKQUOTA DISABLE

3. Exit from SYSMAN.
4. Delete the QUOTA.SYS file in the top-level directory *disk:* [000000].

Note that the system does not suspend quotas across disk mounts if the QUOTA.SYS file is still present.

8.11.3 Purging Files

One of the best ways to conserve disk space is to purge the following:

- Old versions of files

  Before purging files, make sure that the most recent versions are the ones you want to preserve.

- System log files that the system generates automatically, such as the operator log and accounting log files
- Log files created by the PRINT and SUBMIT commands

Encourage individual users to purge files in their own areas and directories. If necessary, you can purge files from some or all directories. Following are examples of purge commands.

Examples

1.
```
$ PURGE/LOG $DISK1:[JONES...]
```

   The command in this example purges all files in the directory [JONES] and all the subdirectories below [JONES] on the $DISK1: device. It logs the files that are deleted (displaying their names on the terminal as they are deleted).

2.
```
$ PURGE/KEEP=3 $DISK1:[*...]
```

   This example uses wildcard characters to perform global purges and uses the /KEEP qualifier to retain only three versions of each file.

8.11.4 Setting Version Limits on Files

Another way to conserve disk space is to limit the number of file versions that users can create in a directory by using the /VERSION_LIMIT qualifier with the SET DIRECTORY or CREATE DIRECTORY command using the following format:

SET DIRECTORY/VERSION_LIMIT=value

Example

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

In the example, files in account [JONES] cannot exceed three versions. If a user in this directory attempts to exceed the three-version limit, the system purges the file, leaving only the three most recent versions.

Note

Be careful about setting a version limit on the master file directory (MFD). Because the system uses the version limit that you set on the MFD on any directory you create beneath the MFD, users might inadvertently lose important data.

8.11.5 Setting File Expiration Dates

Files–11 uses the expiration date of each file to track the use of the file. The expiration dates aid the disposal of seldom-used files when you use the DCL command BACKUP/DELETE command.

File expiration is a file system feature that is available only on Files–11 Structure Level 2 disks.

After you set an expiration date on a volume, the retention periods operate as follows:

- When a user creates a file, the expiration date of the file is the current time plus the maximum time set on the volume.
- Every time a user accesses a file (for either a read or write operation), the current time is added to the minimum time. If the total is greater than the expiration date, the new expiration date is used. (The new expiration date is calculated from the maximum retention period.)

The expiration date of a frequently accessed file fluctuates between the minimum and maximum period plus the current date. When you set a suitable interval between minimum and maximum retention periods, you can balance between accuracy and efficiency in maintaining expiration dates. Be careful about setting expiration dates; either be very specific, or set the expiration date in the simplest way.

Certain commands and utilities, such as the DIRECTORY command and the Backup utility, can selectively operate on files that are expired. For example, you can issue a command like the following one:

```
$ BACKUP/DELETE PUBLIC:[*...]/BEFORE=TODAY/EXPIRED  MUA0:ARCH20JUN
```

In this example, the BACKUP command copies to tape and then deletes all expired files. Users might not be aware of file expiration dates, so retain the tape for a substantial period of time.

For more information on the Backup utility, see Section 10.13.2.

How to Perform This Task

To enable the setting of expiration dates, enter the DCL command SET VOLUME in the following format:

SET VOLUME device-name[:][,...]/RETENTION=(min,max)

For *min* and *max*, specify the minimum and maximum retention periods for files on the volume, expressed as delta time values.

If you specify only a single value in the SET VOLUME/RETENTION command, the system uses the value of the minimum retention period; then the maximum retention period is set to twice the minimum or the minimum plus 7 days, whichever is less. For example, you might set the retention period as follows:

```
$ SET VOLUME PAYVOL1:/RETENTION=(3)
```

The system uses 3 as the minimum retention period. Twice the minimum is 6 days; the minimum plus 7 is 10. Because the system uses the smaller of the two numbers, the retention period is set to 6.

You can simulate the maintenance of ‘‘access dates,’’ which are available in some other operating systems, by setting the retention periods to very small values (for example, 1 hour). Note, however, that doing so substantially increases overhead in the file system.

This feature does not automatically remove unused files; instead, it maintains expiration dates to permit you to develop your own policy for handling files with little or no activity.

Note

If you start maintaining expiration dates on a previously existing volume, be aware that the expiration dates on existing files are 0 until the files are accessed. Files with expiration dates of 0 are considered expired.

Refer to the *OpenVMS DCL Dictionary* for details on the parameters and qualifiers of the SET VOLUME command.

Example

```
$ SET VOLUME DUA0:/RETENTION=(15-0:0,20-0:0)
```

In this example, the command sets the minimum retention period to 15 days and the maximum to 20 days.

8.12 Using the Analyze/Disk_Structure Utility to Check and Repair Disks

You can reclaim disk space by using the Analyze/Disk_Structure utility (ANALYZE/DISK_STRUCTURE) to identify and delete lost files and files marked for deletion. Use this utility on a regular basis to check disks for inconsistencies and errors, and to recover lost files.

This utility detects Files–11 On-Disk Structure (ODS) disk problems that have been caused by hardware errors, system errors, and user errors. ANALYZE /DISK_STRUCTURE does the following:

- Verifies the Files–11 structure on disk volumes
- Reports errors
- Repairs errors when you specify the /REPAIR qualifier

ANALYZE/DISK_STRUCTURE performs the verification of a volume or volume set in eight distinct stages. During these stages, the utility collects information used in reporting errors or performing repairs. However, the utility repairs volumes only when you specify the /REPAIR qualifier.

Digital recommends that you execute ANALYZE/DISK_STRUCTURE in two passes:

1. To report all errors
2. With the /REPAIR and /CONFIRM qualifiers to repair selected errors

Directing ANALYZE/DISK_STRUCTURE Output

By default, ANALYZE/DISK_STRUCTURE directs all output to your terminal. By using the /LIST qualifier, however, you can create a file containing the following information about each file on the disk:

- File identification (FID)
- File name
- Owner
- Errors associated with the file

The following sections explain ways to use ANALYZE/DISK_STRUCTURE:

Task	Section
To report errors (but not repair them)	Section 8.12.1
To both report and repair errors	Section 8.12.2
To recover lost files	Section 8.12.3
To create a disk usage file	Section 8.12.5

The *OpenVMS System Management Utilities Reference Manual* contains additional information about this utility.

8.12.1 Reporting Errors

By default, ANALYZE/DISK_STRUCTURE reports errors but does not make repairs. In this mode, ANALYZE/DISK_STRUCTURE runs through eight stages of data collection and then, by default, prints a list of all errors and lost files to your terminal.

One type of problem that ANALYZE/DISK_STRUCTURE locates is an invalid directory backlink. (A **backlink** is a pointer to the directory in which a file resides.) If your disk has a file with an invalid directory backlink, ANALYZE /DISK_STRUCTURE displays the following message and the file specification to which the error applies:

```
%VERIFY-I-BACKLINK, incorrect directory back link [SYSEXE]SYSBOOT.EXE;1
```

How to Perform This Task

Enter the ANALYZE/DISK_STRUCTURE command using the following format:

ANALYZE/DISK_STRUCTURE device-name:[qualifier]

Example

The following command reports all disk structure errors on the DUA1: device:

```
$ ANALYZE/DISK_STRUCTURE DUA1:
```

8.12.2 Reporting and Repairing Errors

To instruct ANALYZE/DISK_STRUCTURE to repair the errors that it detects, enter the /REPAIR qualifier using the following format:

ANALYZE/DISK_STRUCTURE device-name/REPAIR

To select which errors ANALYZE/DISK_STRUCTURE repairs, enter both the /REPAIR and /CONFIRM qualifiers using the following format:

ANALYZE/DISK_STRUCTURE device-name/REPAIR/CONFIRM

When you issue this command, ANALYZE/DISK_STRUCTURE displays a description of each error and prompts you for confirmation before making a repair.

Examples

1. `$ ANALYZE/DISK_STRUCTURE DUA1:/REPAIR`

   In this example, the command reports and repairs all errors on the DUA1: device.

2. `$ ANALYZE/DISK_STRUCTURE DUA1:/REPAIR/CONFIRM`

   The command in this example might produce the following messages and prompts:

   ```
   %VERIFY-I-BACKLINK, incorrect directory back link [SYS0]SYSMAINT.DIR;1

   Repair this error? (Y or N): Y

   %VERIFY-I-BACKLINK, incorrect directory back link [SYSEXE]SYSBOOT.EXE;1]

   Repair this error? (Y or N): N
   ```

For complete descriptions of all errors and recommended actions, refer to the *OpenVMS Command Definition, Librarian, and Message Utilities Manual*.

8.12.3 Recovering Lost Files

A **lost file** is a file that is not linked to a directory. Under normal circumstances, files are not lost. However, files occasionally lose their directory links because of disk corruption, hardware problems, or user error.

For example, in cleaning up files and directories, you might inadvertently delete directories that still point 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. Though lost, these files remain on the disk and consume space.

Use the Analyze/Disk_Structure utility periodically to check for disk structure errors such as lost files on the disk. When you run ANALYZE/DISK_ STRUCTURE specifying the /REPAIR qualifier, the utility places lost files in *disk:*[SYSLOST] and issues a message about each file, shown in the example that follows. (See the *OpenVMS System Management Utilities Reference Manual* for more information.)

Example

```
$ ANALYZE/DISK_STRUCTURE/REPAIR/CONFIRM DDA0:
```

The command in this example analyzes and repairs all errors and lost files on the DDA0: device.

If it discovers lost files on your disk, ANALYZE/DISK_STRUCTURE issues messages similar to the following:

```
%VERIFY-W-LOSTHEADER, file (16,1,1) []X.X;1
        not found in a directory
%VERIFY-W-LOSTHEADER, file (17,1,1) []Y.Y;1
        not found in a directory
%VERIFY-W-LOSTHEADER, file (18,1,1) []Z.Z;1
        not found in a directory
%VERIFY-W-LOSTHEADER, file (19,1,1) []X.X;2
        not found in a directory
%VERIFY-W-LOSTHEADER, file (20,1,1) []Y.Y;2
        not found in a directory
%VERIFY-W-LOSTHEADER, file (21,1,1) []Z.;1
        not found in a directory
%VERIFY-W-LOSTHEADER, file (22,1,1) []Z.;2
        not found in a directory
%VERIFY-W-LOSTHEADER, file (23,1,1) LOGIN.COM;163
        not found in a directory
%VERIFY-W-LOSTHEADER, file (24,1,1) MANYACL.COM;1
        not found in a directory
```

All lost files in this example are automatically moved to DDA0:[SYSLOST].

Another opportunity to check for lost files on your system is during a backup operation. See Section 10.13.3 for details.

8.12.4 Erasing Old Home Blocks

When you initialize a volume, the initialize operation might not erase old home blocks. These are blocks that were created by previous initialize operations. If a volume that has old home blocks is damaged, you may not be able to recover the volume without erasing the blocks.

You can erase old home blocks manually by using the /HOMEBLOCKS qualifier on the ANALYZE/DISK_STRUCTURE command as follows:

```
$ ANALYZE/DISK_STRUCTURE/REPAIR/HOMEBLOCKS
```

Note that this operation can take up to 30 minutes to complete.

8.12.5 Creating a Disk Usage File

You can create a disk usage file by using the /USAGE qualifier. The **identification record** in the file header contains a summary of disk and volume characteristics. Following the identification record is a series of summary records; one summary record is created for each file on the disk. A summary record contains the owner, size, and name of the file.

Example

```
$ ANALYZE/DISK_STRUCTURE/USAGE=[ACCOUNT]USAGE_DDA0.DAT DDA0:
```

In this example, the /USAGE qualifier creates a disk usage file, USAGE_ DDA0.DAT, and places it in the [ACCOUNT] directory.

8.13 Using Mount Verification for Recovery

Mount verification is a recovery mechanism for disk and tape operations. If a device goes off line or is write-locked while mount verification is enabled, you can correct the problem and continue the operation.

Without mount verification, a **write lock** or offline error causes a volume to be dismounted immediately. All outstanding I/O to the volume is canceled, and all open files on the volume are closed. Any data not yet written to the volume is lost.

8.13.1 Understanding Mount Verification

When the system or a user attempts to access a device after it has gone off line, mount verification is initiated. Usually a device goes off line as the result of a hardware or user error. Once a device is off line, the hardware (and for some disks, the software) marks the disk or tape as ''invalid,'' and I/O requests for that device fail.

As long as mount verification is enabled, the following occurs:

1. The software marks the volume to indicate that it is undergoing mount verification.
2. The software stalls all I/O operations to the disk or tape until the problem is corrected.
3. The operator communication manager (OPCOM) issues a message to operators enabled for DISKS and DEVICES or TAPES and DEVICES. The message announces the unavailability of the disk or tape in the following format:

```
%%%%%%%%%%% OPCOM, <dd-mmm-yyyy hh:mm:ss.cc> %%%%%%%%%%%
Device <device-name> is offline.
Mount verification in progress.
```

When a device goes off line or is write-locked, mount verification sends two messages:

- One message goes to OPCOM.
- The other message, distinguished by the prefix %SYSTEM-I-MOUNTVER, goes directly to the system console (OPA0:), bypassing OPCOM.

The second message is a form of insurance in cases in which OPCOM is unavailable. For example, if the system disk undergoes mount verification or if OPCOM is not present on a system, you at least receive the messages with the %SYSTEM-I-MOUNTVER prefix. Under normal circumstances, the operator terminal receives both messages, with the %SYSTEM-I-MOUNTVER message arriving first.

These messages notify you of the problem, and allow you to correct the problem and recover the operation. When a pending mount verification is canceled by timing out, OPCOM prints a message in the following format:

```
%%%%%%%%%%% OPCOM, <dd-mmm-yyyy hh:mm:ss.cc> %%%%%%%%%%%
Mount verification aborted for device <device-name>.
```

After a mount verification times out, all pending and future I/O requests to the volume fail. You must dismount and remount the disk before users can access it again.

Note

Mount verification caused by a write-lock error does not time out.

Mount Verification and Write-Locking

Suppose, for example, that a volume is mounted on a drive with write-lock off, and someone toggles the WRITE LOCK switch. If mount verification is enabled for the volume, the volume enters mount verification, and all I/O operations to the volume are suspended until you recover the operation, as explained in Section 8.13.2.4.

At mount time, if the system detects that the caches were not written back the last time the volume was used, the system automatically rebuilds the file information by scanning the contents of the volume. However, files being written at the time of the improper dismount might be partially or entirely lost. See Section 8.12 for details about analyzing and repairing these problems.

With the mount verification feature of disk and tape handling, users are generally unaware that a mounted disk or tape has gone off line and returned on line, or in some other way has become unreachable and then restored.

8.13.2 Using Mount Verification

The following sections explain how to perform these tasks:

Task	Section
Enable and disable mount verification	Section 8.13.2.1
Control timeout periods for mount verification	Section 8.13.2.2
Recover from offline errors	Section 8.13.2.3
Recover from write-lock errors	Section 8.13.2.4
Cancel mount verification	Section 8.13.2.5

8.13.2.1 Enabling Mount Verification

Mount verification is enabled by default when you mount a disk or tape. To disable mount verification, you must specify /NOMOUNT_VERIFICATION when you mount a disk or tape.

Note that this feature applies to standard mounted tapes, foreign mounted tapes, and Files–11 disks.

8.13.2.2 Controlling Timeout Periods for Mount Verification

You can control the amount of time (in seconds) that is allowed for a mount verification to complete before it is automatically canceled. The MVTIMEOUT system parameter for disks and the TAPE_MVTIMEOUT system parameter for tapes define the time (in seconds) that is allowed for a pending mount verification to complete before it is automatically canceled. (See the *OpenVMS System Management Utilities Reference Manual* for more information on system parameters.)

Always set either parameter to a reasonable value for the typical operations at your site. Note that resetting the value of the parameter does not affect a mount verification that is currently in progress. Ten minutes (600 seconds) is usually a good value for MVTIMEOUT and TAPE_MVTIMEOUT, whether or not an operator is usually present.

8.13.2.3 Recovering from Offline Errors

When a mounted disk or tape volume goes off line while mount verification is enabled, you can try to recover, or you can terminate the mount request. The following options are available:

- Try to put the device back on line by toggling the START or RUN button on disks or the LOAD button on tapes. If the device has failed, terminate mount verification.
- Take the disk or tape out of the offline and verification-pending state by shutting down mount verification with one of the three techniques described in Section 8.13.2.5. These techniques include canceling the mount request, dismounting the volume, and allowing mount verification to time out.

If you successfully put the device back on line, the mount verification software that polls the disk or tape drive begins verification in the following sequence of steps:

1. The system checks to see that the currently mounted disk or tape has the same identification as the previously mounted volume. In this way, mount verification confirms that this is the same disk or tape that was previously mounted and no switching has occurred.

   If the drive contains the wrong volume, OPCOM issues a message in this format:

   ```
   %%%%%%%%%%% OPCOM, <dd-mmm-yyyy hh:mm:ss.cc> %%%%%%%%%%%
   Device <device-name> contains the wrong volume.
   Mount verification in progress.
   ```

2. Once mount verification completes, the disk is marked as valid, and OPCOM issues a message in the following format:

   ```
   %%%%%%%%%%% OPCOM, <dd-mmm-yyyy hh:mm:ss.cc> %%%%%%%%%%%
   Mount verification completed for device <device-name>.
   ```

3. I/O operations to the disk or tape proceed, as shown in the following example:

   ```
   %%%%%%%%%%% OPCOM, 28-MAY-1995 11:54:54.12 %%%%%%%%%%%
   Device DUA0: is offline.
   Mount verification in progress.

   %%%%%%%%%%% OPCOM, 28-MAY-1995 11:57:34.22 %%%%%%%%%%%
   Mount verification completed for device DUA0:.
   ```

   In this example, the message from OPCOM informs the operator that device DUA0: went off line and mount verification was initiated. The operator finds that the drive was accidentally powered down and successfully powers it up again.

   The last message in the example indicates that mount verification is satisfied that the same volume is on the drive as before the error. All I/O operations to the volume resume.

8.13.2.4 Recovering from Write-Lock Errors

Devices become write-locked when a hardware or user error occurs while a disk or a tape volume is mounted for a write operation. For example, if a disk is write-locked or a tape is missing a write ring, the hardware generates an error. As soon as the software discovers that the disk or tape is write-locked (for example, when an I/O operation fails with a write-lock error), mount verification begins.

OPCOM issues a message in the following format to the operators enabled for DISKS and DEVICES or TAPES and DEVICES, announcing the unavailability of the disk or tape:

```
%%%%%%%%%%%% OPCOM, <dd-mmm-yyyy hh:mm:ss.cc> %%%%%%%%%%%
Device <device-name> has been write-locked.
Mount verification in progress.
```

You can either recover the operation or terminate mount verification. Your options include the following:

- Enable the drive for writing by toggling the hardware WRITE LOCK switch of the disk, or check to see that a tape volume has a write ring.
- If the disk or tape drive is faulty, but another functioning drive is available on the same controller, move the disk or tape to the functioning drive and swap the unit select plugs. (Note that switching to another drive causes the volume to undergo offline mount verification; once this completes, the write-lock mount verification continues.)
- Terminate the mount operation by shutting down mount verification with one of the techniques described in Section 8.13.2.5.

Once the mount verification software determines that the volume is in a write-enabled state, I/O operations to the tape or disk resume with no further messages.

8.13.2.5 Canceling Mount Verification

You can cancel a mount verification request in one of the following ways:

- Dismount the volume with the DCL command DISMOUNT from a process that is not hung.
- If the device is off line, allow mount verification to time out. The expiration value is 10 minutes by default; however, you can use the system parameter MVTIMEOUT (for disk) or TAPE_MVTIMEOUT (for tape) to set the value to whatever you want. When the time expires, the system automatically cancels the pending mount verification. Note that a mount verification initiated by a write-lock condition does not time out.
- Invoke a special canceling routine from the console terminal.

The following sections describe these methods in more detail.

Using the DISMOUNT Command

To dismount a volume, follow these steps:

1. Log in at another terminal, or use any logged-in terminal that has access to the volume. (It does not need to be an operator terminal.)
2. Enter the DISMOUNT/ABORT command for the volume. (To use the /ABORT qualifier with a volume that is not mounted group or system, you must have volume ownership or the user privilege VOLPRO.)

   If your system is in a VMScluster environment, also specify the /CLUSTER qualifier.

   When you cancel a pending mount verification by dismounting the volume, OPCOM issues a message in the following format:

   ```
   %%%%%%%%%%%% OPCOM, <dd-mmm-yyyy hh:mm:ss.cc> %%%%%%%%%%%
   Mount verification aborted for device <device-name>.
   ```

If you do not have access to the volume, you receive an error message. You can try again if you can find an appropriate process to use. If your process hangs, the system file ACP is hung, and you cannot use this technique to cancel mount verification.

3. When the cancellation is complete, remove the volume from the drive.

Using Cancellation Commands

To cancel mount verification, you run a special interrupt program called Interrupt Priority C (IPC). IPC issues a software interrupt to gain the attention of the operating system. You use IPC to cancel mount verification, enter the debugger, or recalculate the quorum on a VMScluster system. (The debugger in this case refers to the system-level debugger, XDELTA.)

The IPC program converts lowercase characters to uppercase, issues the terminal bell character whenever it receives illegal characters (such as most control characters), compresses multiple spaces, and ignores leading spaces.

How to Perform This Task

1. Invoke IPC by using commands specific to the hardware you are using. Display the console prompt and enter a sequence of commands from the console terminal.

   VAX

   On VAX systems, enter the following commands:

```
>>> HALT
>>> D/I 14 C
>>> CONT
IPC> ♦
```

   Alpha

   On Alpha systems, enter the following commands:

```
>>> HALT
>>> D SIRR C
>>> CONT
IPC> ♦
```

2. Enter one of the following commands in response to the IPC> prompt:

Command	Explanation
C *device-name*	Cancels any pending mount verification on the device specified. (A warning is given if no mount verification was in progress for that device.) For example: `IPC> C MUA1:`
X	Transfers control to the debugging tool XDELTA (provided it was loaded with the system by setting the appropriate value in the boot file). If XDELTA has not been loaded, the prompt IPC> is reissued. For example: `IPC> X`
Q	Recalculates the quorum in a VMScluster system.

   When a pending mount verification is canceled, OPCOM prints a message in the following format:

```
%%%%%%%%%%% OPCOM, <dd-mmm-yyyy hh:mm:ss.cc> %%%%%%%%%%%
Mount verification aborted for device <device-name>.
```

3. Exit from IPC by pressing Ctrl/Z.

4. After you successfully cancel a pending mount verification with this technique, you must dismount and then remount the volume before you can access it again.

Example

```
%%%%%%%%%%% OPCOM, 28-MAY-1995 10:54:54.12 %%%%%%%%%%%
        Device DUA0: is offline.
        Mount verification in progress.
```

VAX

On VAX systems, you might enter the following commands:

```
>>> HALT
>>> D/I 14 C
>>> CONT
IPC> C DUA0:
IPC> Ctrl/Z
%SYSTEM-I-MOUNTVER, _DUA0: has aborted mount verification.
%%%%%%%%%%% OPCOM, 28-MAY-1995 10:56:26.13 %%%%%%%%%%%
Mount verification aborted for device DUA0: ♦
```

Alpha

On Alpha systems, you might enter the following commands:

```
>>> HALT
>>> D SIRR C
>>> CONT
IPC> C DUA0:
IPC> Ctrl/Z
%SYSTEM-I-MOUNTVER, _DUA0: has aborted mount verification.
%%%%%%%%%%% OPCOM, 28-MAY-1995 10:56:26.13 %%%%%%%%%%%
Mount verification aborted for device DUA0: ♦
```

In both examples, device DUA0: is off line, but you are unable to spin the disk back up. No other drive is available on the controller, so you cannot switch the unit select plugs of the two drives.

Do not enter a DISMOUNT command for the disk, because it was mounted as a private volume, and you do not have access to it. The %SYSTEM-I-MOUNTVER message also appears because this is the console terminal.

8.14 Using the Bad Block Locator Utility to Detect Media Errors

The DCL command ANALYZE/MEDIA invokes the optional Bad Block Locator utility (BAD), which analyzes block-addressable media and records the location of blocks that cannot reliably store data.

Note

Many newer devices automatically check for bad blocks; therefore, BAD is more useful with older devices that do not check for bad blocks.

To test the blocks on a volume, ANALYZE/MEDIA does the following:

- Writes a test pattern to each block on the media
- Reads the contents of the block into a buffer
- Compares the data read back with the data written

If the data does not compare exactly, a block cannot reliably store data.

8.14 Using the Bad Block Locator Utility to Detect Media Errors

When the Bad Block Locator utility locates a bad block, it records the address of the block. Consecutive bad blocks are recorded as single entries for non-last-track devices. After it finishes testing the disk, BAD writes the addresses of the bad blocks into a file called the detected bad block file (DBBF).

Caution

Testing a volume for bad blocks destroys its contents. However, you can update the detected bad block file (DBBF) without erasing the contents of the volume by using the ANALYZE/MEDIA qualifiers /NOEXERCISE and /BAD_BLOCKS.

To use BAD, you perform the following steps:

1. Allocate the device with the DCL command ALLOCATE (to ensure that the device is not accessed by any other programs).
2. Enter the DCL command MOUNT/FOREIGN.

   When the device is mounted as foreign, the system does not recognize it as a Files–11 volume, and BAD can execute.
3. Enter the DCL command ANALYZE/MEDIA.

Refer to online help or to the optional *OpenVMS Bad Block Locator Utility Manual* for details on using the Bad Block Locator utility.

9
Using Files and Directories

This chapter discusses concepts and tasks related to file protection, file manipulation, and data transfer.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Getting file information	Section 9.2
Protecting disk files	Section 9.3.3
Protecting disk directories	Section 9.3.4
Protecting magnetic tape files	Section 9.3.5
Accessing disk files	Section 9.4
Accessing tape files	Section 9.5
Copying and transferring files	Section 9.6

This chapter explains the following concepts:

Concept	Section
DCL commands with files	Section 9.1
File protection	Section 9.3.1
Tape file names	Section 9.5.1

9.1 Using DCL Commands with Files

You use the DIGITAL Command Language (DCL) to perform a number of operations on files; among them are the following:

Operation	DCL Command
Retrieve disk and magnetic tape file information, such as device and protection characteristics, and display this information on your terminal screen	SHOW commands listed in Table 9–1
Modify disk file characteristics, such as protection or UIC information	SET commands listed in Table 9–3
Display the contents of a directory	DIRECTORY
Display the contents of a file	TYPE
Copy files to and from disk and magnetic tape volumes	COPY

Most DCL commands require file-structured devices. (The *OpenVMS DCL Dictionary* lists commands that do not require file-structured devices.)

In addition to manipulating files through DCL, you can write programs to assist you in routine file-manipulation tasks. You can write these programs in any of the languages supported by the operating system.

If you want to manipulate individual records within files (that is, to access files at the record level), write programs that include OpenVMS Record Management Services (RMS) facilities. Examples of the RMS facilities used to manipulate files at the record level are in the *OpenVMS Record Management Services Reference Manual*.

9.2 Getting File Information

With the DCL command DIRECTORY, you can retrieve information about disk and magnetic tape files in a directory, using the following format:

DIRECTORY [filespec[,...]]

When you include certain command qualifiers along with the DIRECTORY command, you can retrieve information in addition to a list of the names of the files in the directory. See the *OpenVMS DCL Dictionary* for a list of qualifiers that you can use with the DIRECTORY command and with other DCL commands.

The following examples illustrate three cases of retrieving information from the [MALCOLM] directory, which resides on a disk with the logical name DISK$DOCUMENT.

Examples

1.
```
$ DIRECTORY AVERAGE.*

Directory DISK$DOCUMENT:[MALCOLM]

AVERAGE.EXE;6      AVERAGE.FOR;6      AVERAGE.LIS;4      AVERAGE.OBJ;12

Total of 4 files.
```

   The DIRECTORY command in this example lists all file types of the AVERAGE file and the version number of each file. The command would also list all versions of these files; however, only one version of each file exists.

2.
```
$ DIRECTORY/SIZE/DATE/VERSIONS=1/PROTECTION  AVERAGE

Directory DISK$DOCUMENT:[MALCOLM]

AVERAGE.EXE;6       6        10-APR-1995 15:43 (RWED,RWED,RWED,RE)
AVERAGE.FOR;6       2         2-APR-1995 10:29 (RWED,RWED,RWED,RE)
AVERAGE.LIS;4       5         9-APR-1995 16:27 (RWED,RWED,RWED,RE)
AVERAGE.OBJ;6       2         9-APR-1995 16:27 (RWED,RWED,RWED,RE)

Total of 4 files, 15 blocks.
```

   The DIRECTORY command in this example displays all the file types of the AVERAGE file and the version number of each file. The /SIZE qualifier displays the size of each file in blocks used. The /DATE qualifier displays the creation date of the version of the file that is listed. The VERSIONS=1 qualifier limits the number of versions of the file displayed to one (the most recent) version. The /PROTECTION qualifier displays the file protection for each file.

3. `$ DIRECTORY/FULL/VERSIONS=1 [MALCOLM...]AVERAGE.EXE`

```
Directory DISK$DOCUMENT:[MALCOLM]

AVERAGE.EXE;6                File ID:  (4098,149,0)
Size:           36/36        Owner:    [DOCUMENTATION,MALCOLM]
Created:  27-MAY-1995 12:22:26.30
Revised:  27-MAY-1995 12:22:51.35 (2)
Expires:   <None specified>
Backup:    3-JUN-1995 22:03.09

Effective: <None specified>
Recording: <None specified>
File organization:  Sequential
Shelved state:      Online
File attributes:    Allocation: 36, Extend: 36, Global buffer count: 0
                    No version limit
Record format:      Variable length, maximum 255 bytes
Record attributes:  Carriage return carriage control
Journaling enabled: None
File protection:    System:RWED, Owner:RWED, Group:RE, World:
Access Cntrl List:  None

Total of 1 file, 36/36 blocks.


Directory DISK$DOCUMENT:[MALCOLM.TEST]

AVERAGE.EXE;1                File ID:  (7714,29,0)
Size:           36/36        Owner:    [DOCUMENTATION,MALCOLM]

Created:  15-APR-1995 10:12
Revised:  15-APR-1995 10:12 (1)
Expires:   <None specified>
Backup:   15-APR-1995 22:41

Effective: <None specified>
Recording: <None specified>
File organization:  Sequential
Shelved state:      Shelved
File attributes:    Allocation: 36, Extend: 36, Global buffer count: 0
                    No version limit
Record format:      Variable length, maximum 255 bytes
Record attributes:  Carriage return carriage control
Journaling Enabled : None

File protection:    System:RWED, Owner:RWED, Group:RE, World:
Access Cntrl List:  None

Total of 1 file, 36/36 blocks.
Grand total of 2 directories, 2 files, 72/72 blocks.
```

The DIRECTORY command in this example displays a full directory listing of one version of the AVERAGE.EXE file in the top-level directory [MALCOLM] and subdirectories under it.

9.3 Protecting Files

The following sections discuss file protection concepts and explain how to perform these tasks:

Task	Section
Display file ownership and protection	Section 9.3.2
Protect disk files	Section 9.3.3
Protect disk directories	Section 9.3.4
Protect magnetic tape files	Section 9.3.5

9.3.1 Understanding File Protection Concepts

You can protect data on disk and magnetic tape media at the following levels:

Level of Protection	Description; For More Information	
Device level	For information on setting device protection characteristics, see the descriptions of the DCL commands INITIALIZE, MOUNT, SET DEVICES, SET SECURITY/PROTECTION, and SET VOLUME in Chapter 8 and in the *OpenVMS DCL Dictionary*. Refer to Chapter 7 for additional information about peripheral devices.	
Volume level	The system provides protection for both disk and tape volumes. For more information about each of these, see the following sections:	
	Disk volume protection	Section 8.4.1
	Tape volume protection	Section 8.4.2
File level	The system provides protection for disk files and directory files. For more information, see the following sections:	
	Individual disk files	Section 9.3.3
	Directory files that reside on disk volumes	Section 9.3.4

You can protect data residing on disk and tape volumes by using one or more of the following:

Type of Protection	For More Information
UIC-based protection codes	Chapter 11
Access control lists (ACLs)	Chapter 11
ISO 9660-formatted media protection	Section 8.4.2
ANSI-standard accessibility protection (magnetic tape only)	Section 8.4.2

For the most part, file protection is transparent. Tools exist, however, to adjust the protection of a file. You can set the protection or modify the ACL of a file if at least one of the following is true:

- You own the file.
- You have control access to the file.
- You have SYSPRV privilege.
- The group part of your UIC is less than or equal to MAXSYSGROUP.
- You have GRPPRV and you have the same group UIC as the file.

9.3.2 Displaying File Ownership and Protection

You can display ownership and protection information with the commands and qualifiers shown in Table 9–1.

Table 9–1 DCL Commands to Display Ownership and Protection

Command	Use to Display
DIRECTORY/ACL *filespec*	ACL of file
DIRECTORY/OWNER_UIC *filespec*	UIC of owner of file
DIRECTORY/PROTECTION *filespec*	UIC-based protection of file
DIRECTORY/SECURITY	All of the above
DIRECTORY/FULL *filespec*	All of the above and other, non-security information
SHOW DEVICE/FULL *device-name*	Device UIC and protection
SHOW PROCESS	Process UIC
SHOW PROTECTION	Default file protection
SHOW SECURITY	All of the above

Directory structures do not apply to tape volumes. However, you can use the DIRECTORY command to search for files on tape volumes. Section 9.5 describes how to access tape files for read and write operations and also explains the use of the DIRECTORY command for tapes.

The DCL command SHOW PROTECTION displays the current process default protection. This protection is applied to files created during your terminal session or to batch jobs, where defaults from directories or previously existing versions are not available.

Note

To use the SHOW PROTECTION command to display the default protection of magnetic tapes, you must specify the /PROTECTION qualifier with the INITIALIZE command when you initialize the magnetic tape volume. Otherwise, the protection is not written to the magnetic tape volume. See the description of initializing magnetic tape volumes in Section 8.1.3.2.

The next example illustrates how you can use the SHOW PROTECTION command to display the default protection characteristics for disk files.

Example

```
$ SHOW PROTECTION
SYSTEM=RWED, OWNER=RWED, GROUP=RE, WORLD=NO ACCESS
```

In this example, the SHOW PROTECTION command requests a display of the current protection defaults.

9.3.3 Protecting Disk Files

Each file on a disk has its own protection code, which is distinct from the protection that applies to the disk volume itself. Files residing on disk volumes have the access types shown in Table 9–2.

Table 9–2 Access Types with Disk File Protection

Access Type	Description
Read	Gives you the right to read, print, or copy a disk file. Read access automatically includes execute access to a specified file or group of files on disk.
Write	Gives you the right to write to or change the contents of a file, but not delete it. Write access allows modification of the file characteristics that describe the contents of the file.
Execute	Gives you the right to execute a file that contains an executable program image or DCL command procedure.
Delete	Gives you the right to delete the file. To delete a file, you must have delete access to the file and write access to the directory that contains the file.
Control	Gives you the right to change file characteristics, including the protection code and ACL. Special restrictions apply to changing the owner of a file.

If you do not define a protection code for a file when you create it, the system applies default protection. If a version of the file already exists, protection is taken from the previous version.

For a new file, the system determines protection in two major ways:

- If the directory where the file is to be cataloged has an associated access control entry (ACE) that specifies the default protection, the system uses the specified protection.
- If the directory does not have the default protection ACE, the system uses the default process protection. You establish the default process protection explicitly with the SET PROTECTION/DEFAULT command, or by default when you log in.

For disk volumes, each file on the volume can have a different protection associated with it. The SET SECURITY/PROTECTION command and other file-manipulating commands allow you to define the protection for individual files.

Note

To protect a file completely, you must protect both the file itself and the directory that lists the file. To protect a file against unauthorized access, specify the proper protection both for the directory that lists the file and for the file itself. See Section 9.3.4 for instructions on protecting directories.

The following sections explain how to perform these tasks:

Task	Section
Set default disk file protection	Section 9.3.3.1
Set explicit disk file protection	Section 9.3.3.2
Modify disk file protection characteristics	Section 9.3.3.3

9.3.3.1 Setting Default Disk File Protection

A new file receives default UIC-based protection and the default access control entries (ACEs), if any, of its parent directory. A new version of an existing file receives the UIC-based protection and ACL of the previous version.

The protection of a renamed file is unchanged unless you use the RENAME /INHERIT command.

How to Change Default UIC Protection

The operating system provides each process with a default UIC-based protection of (S:RWED,O:RWED,G:RE,W). To change the default protection that is applied to files created by that process, enter the SET PROTECTION/DEFAULT command using the following format:

SET PROTECTION[=(code)]/DEFAULT

where:

code — Defines the protection to be applied to the specified files. If you omit the code, the access is set to the current default protection.

For example, if you place the following command in your login command procedure, you grant all processes read and execute access to any files that you subsequently create:

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

(Remember that you must execute the login command procedure for this command to take effect.)

9.3.3.2 Setting Explicit Disk File Protection

You can explicitly specify UIC-based protection for a new file with the /PROTECTION qualifier (valid with the BACKUP, COPY, RENAME, and CREATE commands), as shown in the following command line:

```
$ CREATE MAST12.TXT/PROTECTION=(S:RWED,O:RWED,G,W)
```

After a file is created and you have created an ACL for the file, you can modify the ACL and add as many ACEs to the ACL as you want. The protection specified by the ACL overrides the UIC protection of the file.

Following are examples showing how to check and specify protection codes.

Examples

1.
```
$ SHOW PROTECTION
SYSTEM=RWED, OWNER=RWED, GROUP=RE, WORLD=NO ACCESS
```

   The SHOW PROTECTION command displays the current default protection. In this example, the response shows the system default protection, which indicates that the system and owner have all types of access, group users have read and execute access, and world users have no access.

2. `$ SHOW SECURITY IMAGES.DIR`

```
DBA1:[SADAMS]IMAGES.DIR;1 object of class FILE
     Owner: [SAM,SADAMS]
     Protection: (System: RWE, Owner: RWE, Group: RE, World: E)
     Access Control List:
       (IDENTIFIER=[SAM,SADAMS],ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)
```

   In this example, the SHOW SECURITY command displays the current protection associated with the file IMAGES.DIR.

3. `$ DIRECTORY/SECURITY IMAGES.DIR`

```
Directory DBA1:[SADAMS]

IMAGES.DIR;1         [VMS,SADAMS]          (RWE,RWE,RE,E)
         (IDENTIFIER=[VMS,SADAMS],ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL)

Total of 1 file.
```

   In this example, the /SECURITY qualifier with the DIRECTORY command displays the current protection associated with the IMAGES.DIR file.

4. `$ COPY/PROTECTION=(SYSTEM:RW,OWNER:RWED,GROUP:RW,WORLD) ABC.DAT XYZ.DAT`

   In this example, the /PROTECTION qualifier specifies a protection code when the ABC.DAT file is copied to XYZ.DAT.

5. `$ SET SECURITY/PROTECTION=(SYSTEM:RWE,OWNER:RWED,GROUP:RE,WORLD) ABC.DAT`

   In this example, the SET SECURITY/PROTECTION command changes the protection for an existing file. The command gives the following instructions regarding the file ABC.DAT: system users have read, write, and execute access; the owner has read, write, execute, and delete access; group users have only read and execute access; world users have no access.

   Control access is implied and unchangeable for system and owner categories but not for group and world.

9.3.3.3 Modifying Disk File Protection Characteristics

Table 9–3 shows the DCL commands that you can use to establish and modify the protection characteristics of files.

Table 9–3 DCL Commands to Modify File Protection Characteristics

Command	Description	For More Information
SET DIRECTORY	Modifies the characteristics of one or more directories. The directory protection can override the protection of individual files within the directory.	See Section 9.3.4.
SET FILE	Modifies the characteristics of one or more files, including the version limits on files.	See Section 9.3.3.3.2.
SET PROTECTION /DEFAULT	Sets the default UIC protection on files.	See the *OpenVMS Guide to System Security*.

(continued on next page)

Table 9–3 (Cont.) DCL Commands to Modify File Protection Characteristics

Command	Description	For More Information
SET SECURITY	Modifies the security profile of an object. Such a profile contains the following characteristics: • An access control list (ACL). • A protection code, which defines access to objects based on the categories of system, owner, group, and world. • An owner. The system uses the owner characteristic to interpret the protection code.	See the *OpenVMS Guide to System Security* and the *OpenVMS DCL Dictionary*.
SET VOLUME	Changes the characteristics of one or more mounted Files–11 volumes. The /FILE_PROTECTION qualifier sets the default protection to be applied to all files on the specified disk volume.	See Section 8.4.1.2.

For a complete list of the command qualifiers and parameters applicable to each of these DCL commands, see the *OpenVMS DCL Dictionary*.

9.3.3.3.1 Changing File Protection Characteristics To change or reset the protection characteristics of one or more files, use the following format:

SET SECURITY/PROTECTION = code file-spec[,...]

where:

code	Defines the protection to be applied to the specified files. You cannot omit the code.
file-spec	Specifies one or more files for which the protection is to be changed. A file name and file type are required. If you omit a version number, the protection is changed only for the highest existing version of the file. Wildcard characters are allowed.

The following examples show ways to change file protection.

Examples

1.
```
$ DELETE INCOME.DAT;3
%DELETE-W-FILNOTDEL, error deleting DISK1:[SMITH]INCOME.DAT;3
-RMS-E-PRV, insufficient privilege or file protection violation
$ SET SECURITY/PROTECTION=OWNER:D INCOME.DAT;3
$ DELETE INCOME.DAT;3
```

In this example, the file INCOME.DAT;3 is protected against deletion. The SET SECURITY/PROTECTION command changes only the owner's delete access for the file INCOME.DAT;3. Now the owner can delete the file.

2.
```
$ SET SECURITY/PROTECTION=(SYSTEM:R,OWNER:RWED,GROUP:RW) PAYROLL.LIS
```

In this example, the SET SECURITY/PROTECTION command changes the protection codes applied to the PAYROLL.LIS file. To the file, the command gives the system read access; the owner has read, write, execute, and delete access; and users in the owner's group have read and write access.

9.3.3.3.2 Using the SET FILE Command

You can use the DCL command SET FILE to modify the characteristics of one or more files or to assign an additional name, or alias, to a file. The following examples illustrate ways you can use the SET FILE command.

Examples

1. `$ SET FILE/EXPIRATION_DATE=15-APR-1995:11:00 BATCH.COM;3`

   This SET FILE command requests that the expiration date of the file BATCH.COM;3 be set to 11:00 a.m., April 15, 1995.

2. `$ SET FILE/BEFORE=15-APR-95/ERASE_ON_DELETE PERSONNEL*.SAL`

   This SET FILE command erases disk locations for files that are deleted with commands such as DELETE or PURGE when applied to all files that match the file specification PERSONNEL*.SAL and are dated before April 15, 1995.

3. `$ SET FILE/OWNER_UIC=[DOCUMENTATION,GRAY]/VERSION_LIMIT=100 MYFILE.DAT`

   This SET FILE command modifies the characteristics of the file MYFILE.DAT, changing the owner UIC and assigning a file version limit of 100. Note that the /OWNER_UIC qualifier requires SYSPRV or GRPPRV privilege for changing the ownership at the system or group level.

4. `$ SET FILE OLD_FILENAME.DAT/ENTER=NEW_FILENAME.DAT`

   This SET FILE command assigns an additional name, or alias (NEW_ FILENAME.DAT), to the file OLD_FILENAME.DAT. Both the original name and the alias refer to the same file. For this reason, be careful when you delete files with aliases. To keep the file, but to remove one of its names, use the /REMOVE qualifier with the SET FILE command. You cannot use wildcards in the file name. (Refer to the *OpenVMS DCL Dictionary* for details.)

9.3.4 Protecting Disk Directories

Each directory has a protection associated with it. Directory protection can override the protection of individual files within the directory. For example, if a directory denies world access, world users cannot look up files in that directory even though the files permit world access.

For directory protection, you can use the access types shown in Table 9–4.

Table 9–4 Access Types for Directory Protection

Access Type	Description
Read	Gives you the right to examine, print, or copy a file. If you have read access to a directory, you can display the contents of the directory with the DIRECTORY command. For example, if you have read access to the directory [JONES], you can enter the following: `$ DIRECTORY [JONES]` This command displays the files contained in the [JONES] directory. With read access, you can access any file listed in the directory, unless the protection on that file denies you access. If the protection applied to the whole directory denies you read access, then you cannot access even those files in the directory that permit access to users in your group.
Write	Gives you the right to modify or write to a directory. However, you must have both read and write access to a directory to create files in the directory, to rename files in the directory, or to perform any file operation that involves changes to the directory file.

Execute	When applied to directories, execute access allows you to access files by name but does not allow you to list all the entries in a directory (that is, to use specific or implied wildcards). For example, assume that you have execute access to the [JONES] directory, and you enter the following command:
	`$ DIRECTORY [JONES]`
	The system responds with an error message of ''insufficient privilege or file protection violation'' and does not list the files in the [JONES] directory. However, if you know that the file DATAFILE.DAT resides in the [JONES] directory, you can enter the following command:
	`$ TYPE [JONES]DATAFILE.DAT`
	The system displays the contents of the file. Thus, with execute access, you can perform some, but not all, of the operations that you can with read access. (Access to individual files is still controlled by their file protection.)
	As another example, to display the contents of the EXPENSES.DAT file, you must have read or execute access to each directory in the directory tree, that is, to the JONES, REPORTS, and JUNE directories:
	`$ TYPE [JONES.REPORTS.JUNE] EXPENSES.DAT`

Delete	Gives you the right to delete a directory file. You must remove all entries from a directory before you can delete the directory file. When you create a directory with the CREATE /DIRECTORY command, you do not, by default, get delete access. If you want to be able to delete a directory file, you must use the DCL command SET SECURITY/PROTECTION to explicitly assign delete access to the owner category.
Control	Gives you the right to change the characteristics of a directory.

Using UIC Directory Protection

You cannot completely protect a file without applying at least the same protection to the directory in which the file resides. For example, if you deny a user all access to a file but allow that user read access to the file's directory, the user cannot access the contents of the file but can see that it exists. Conversely, a user allowed access to a file and denied access to the file's directory (or one of the parent directories) cannot see that the file exists.

Note

To protect sensitive files, the directory protection alone is not adequate. You must also protect each individual file contained within the directory. Section 9.3.3 contains instructions for protecting disk files.

By default, top-level directories receive UIC-based protection (S:RWE,O:RWE,G:RE,W:E) and no ACL. A newly created subdirectory receives the same protection as its parent directory, but delete access is removed from all categories.

Following are guidelines for specifying UIC-based protection on a directory:

- To specify UIC-based protection explicitly when creating a directory, use the /PROTECTION qualifier or the CREATE/DIRECTORY command. You cannot specify an ACL for the directory before you create the directory.
- To change the UIC-based protection of an existing directory, use the SET SECURITY/PROTECTION command for the directory file.
- You can limit but not prohibit directory access by specifying execute access but not read access. Execute access on a directory permits you to examine and read files that you know are in the directory (that is, you know the file specifications) but prevents you from displaying a list of the files in the directory.

The following sections explain how to change directory protection characteristics and default ACL protection.

9.3.4.1 Changing Directory UIC Protection Characteristics

The DCL command SET DIRECTORY modifies the characteristics of one or more directories.

Example

```
$ SET DIRECTORY/OWNER_UIC=[360,020] [DAVIS],[USERS]
```

The SET DIRECTORY command in this example modifies both the [DAVIS] and [USERS] directories, changing their owner UICs. Using the OWNER_UIC qualifier requires SYSPRV (system privilege).

9.3.4.2 Changing Default ACL Protection

You can override default UIC protection for specified directories or subdirectories by placing a default protection ACE in the ACL of the appropriate directory file. The default protection specified in the ACE is applied to any new file created in the specified directory or in any subdirectory of the directory.

Example

The following ACE, which must be in the ACL of a directory file, specifies that the default protection (for files created in the directory and its subdirectories) will allow system and owner processes full access, group processes read and execute access, and world users no access:

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

9.3.5 Protecting Magnetic Tape Files

Because tapes are single-user devices, tape protection is only at the volume level. The protection codes for magnetic tape volumes are usually assigned with the INITIALIZE command.

You cannot use DCL commands to change protection characteristics on magnetic tape volumes. See Section 8.5.1 for more information.

9.4 Accessing Disk Files

This section describes how to use DCL commands to access files at the file level, not at the record level. This applies to reading files on disks, which is explained in this section, as well as to copying tape files, which is explained in Section 9.6.1.

Although DCL does allow you to manipulate files at the record level, for reasons of performance, you probably want to use a conventional programming language instead. Digital recommends that you write programs using the OpenVMS Record Management Services (RMS) facilities, which are specifically designed to access files at the record level. You can write these programs in any higher-level language that the operating system supports.

To access disk files at the file level, you can use DCL commands. You cannot, however, use DCL commands to read or write files that are not in the standard formats supported by the operating system. If the file formats are not standard, you must mount the volumes on which they reside with the /FOREIGN qualifier to have read and write access.

Although the examples used in this section show how to access disk files on RA90 disk packs, they also apply to other devices.

To read the contents of a disk file, use the DCL command TYPE, which displays the contents of a file on your terminal. To find the exact location of the disk file you want to read, use the DCL command DIRECTORY.

How to Perform This Task

If, for example, you want to read the contents of a file named HISFILE, which is located somewhere in the directory [CHARLES] on a disk device whose logical name is DISK$DOCUMENT, follow these steps:

1. Look for the exact location of HISFILE by entering the following command:

   ```
   $ DIRECTORY DISK$DOCUMENT:[CHARLES...]HISFILE.*
   ```

   This command instructs the operating system to search the entire [CHARLES] directory, including all the subdirectories, for all file types and version numbers of HISFILE.

   The following information is displayed on your terminal:

   ```
   Directory DISK$DOCUMENT:[CHARLES.MEMO]

   HISFILE.UPD;1

   Total of 1 file.
   ```

   This display informs you that only one version of HISFILE exists, that its file type is UPD, and that it resides in the [CHARLES.MEMO] directory.

2. To read the contents of this file, enter the following command:

   ```
   $ TYPE [CHARLES.MEMO]HISFILE.UPD
   ```

   The contents of HISFILE are displayed on your terminal.

9.5 Accessing Tape Files

This section describes file-level access for tapes. When you request access to a standard-labeled volume or a file, the operating system checks at the volume and file level to ensure that your process can access the volume or file. The level at which the system checks access depends on the operation you request and the type of access the operation requires.

When you access a volume or a file, the operating system software reads the volume- and file-header labels to determine whether access to the volume or file is restricted. Which label is read depends on the operation requested. For example, if you want to mount a volume, your process must have access to it.

The protection set on a file determines your access to the file. The expiration date field in the header can prevent you from overwriting or appending to a file immediately preceding the one in question. If the expiration date field has not been reached, a file has not expired.

To overwrite an unexpired file, you must specify the /OVERRIDE=EXPIRATION qualifier when you mount the volume. Performing this operation requires that you have read or write access.

Following a section that explains tape file names are sections that tell how to perform these tasks:

Task	Section
Locate standard-labeled tape files	Section 9.5.2
Use wildcards with tape files	Section 9.5.3
Read files on tape volumes	Section 9.5.4
Write files to tape volumes	Section 9.5.5

9.5.1 Understanding Tape File Names

OpenVMS systems accept two types of file names for magnetic tapes:

- OpenVMS extended names

  Use OpenVMS extended names if your files are to remain on media using only the OpenVMS operating system.

- Standard names

  Use standard names if you want to move files to media on operating systems other than OpenVMS.

Table 9–5 compares characteristics of OpenVMS extended names and standard names.

Table 9–5 Comparison of OpenVMS Extended Names and Standard Names

Characteristic	OpenVMS Extended Names	Standard Names
Valid with...	Tape and disk volumes	Tape volumes
Format	filename.type;version	filename.;version (*Version* is optional.)
Length	39.39;	17.;

(continued on next page)

Table 9–5 (Cont.) Comparison of OpenVMS Extended Names and Standard Names

Characteristic	OpenVMS Extended Names	Standard Names
Valid Characters	A through Z; 0 through 9; ampersand (&), hyphen (-), underscore (_), and dollar sign ($); wildcard characters asterisk (*) and percent sign (%)	ASCII ''a''[1] characters enclosed in quotation marks (" "). Note that within a file name, DCL interprets a double set of quotation marks ("") as a single set ("). If a name has fewer than 17 characters, the system pads the name on the right with spaces to arrive at the 17-character maximum length.
Examples	OPENVMS_FILENAME.DAT;23	"GENLABEL#123";2

[1]The ASCII ''a'' character set is defined in Clause 7.4.1 of the ISO 9660 Standard.

9.5.2 Locating Standard-Labeled Tape Files

Before accessing a particular file for a read or write operation, you might want to search the magnetic tape volume for that file. You use the DCL command DIRECTORY to locate a file or group of files on a tape volume.

When you specify a file name for a file residing on tape, the tape file system does the following:

1. Compares the file name with the file header labels of each file until it finds a match in the file identifier field of the file header labels.
   - If you supply a version number in the file name, the magnetic tape file system compares it with the generation number and generation version-number fields in the first file header label.
   - If you do not specify a version number, the system neither defaults a version number nor checks the generation number and generation version-number fields.
2. The system selects the first file on the magnetic tape whose file name in the file identifier field matches the specified file name.

   The operating system supports neither the directory nor the latest version number concept for magnetic tape volumes. The system does not search for or list the latest version of a specified file. The magnetic tape file system cannot increment version numbers of files written to tape; therefore, two or more files in the same volume set can have the same file name and version number.

   Because the tape file system selects the first matching file name and version number (if specified), the position of the magnetic tape within the volume set determines which file is returned on a search operation. A search operation begins at the current position, so you might want to rewind the volume set before accessing a file.
3. The search for a matching file and version number (if specified) continues at the beginning of the header-label set of the next file. The search ends when the magnetic tape is positioned at the file where the search began.

   If the system does not find the requested file on the current volume, it searches the remaining volumes in the volume set sequentially and then searches from the beginning of the first volume of the volume set. If the system still does not find the file name, it reports an error.

9.5.3 Using Wildcard Characters with Tape Volumes

The OpenVMS operating system supports a limited use of wildcard characters in file specifications for tape volumes.

Table 9–6 explains the use of wildcard characters with OpenVMS extended names and with standard names.

Table 9–6 Wildcard Character Support with Tape Volumes

Wildcard Character	OpenVMS Extended Names	Standard Names	Description
Asterisk (*)	X	X	In OpenVMS extended names, you can use an asterisk anywhere in the file name and file type field to match a field or portion of a field. You can also use the asterisk in the version number field. In standard names, you can use only a single asterisk in a field.
Percent sign (%)	X		In OpenVMS extended names, you can use a percent sign in a file specification only to match character positions within a field. You cannot use the percent sign in the version number field.

Unlike OpenVMS extended names, which can consist of up to 39 characters each for the file name and file type, standard names can have a maximum of 17 characters.

The following examples show how to use wildcard characters in file specifications to search for files on tape volumes. These examples also show how you can use the DIRECTORY command with tapes. Note that the DIRECTORY command does not work the same with tape files as with disk files.

Examples

1.
```
$ DIRECTORY MFA1:*.*;*
```

   This command instructs the system to search a volume set. Because asterisks are used in the file specification and the asterisk is a valid wildcard character for both standard and OpenVMS extended names, the system returns both OpenVMS extended names and standard names. Note that the system returns tape file names within quotation marks.

2.
```
$ DIRECTORY MTA1:%*.*;*
$ DIRECTORY MTA0:*.%*;*
```

   In these two commands, the search can match only with OpenVMS extended names because the percent sign is not valid for standard names. In the second command, the file type field must contain at least one character. Files with no file type are not returned.

3.
```
$ DIRECTORY MTA0:*.;*
```

   In this example, the DIRECTORY command instructs the system to search for files with both standard names and OpenVMS extended names that do not have a file type.

9.5.4 Reading Files on Tape Volumes

When you access a tape file for a read operation, the tape is positioned at the beginning of the file section after the file header labels. When you access a file residing on a tape volume only to read the attributes in the header labels (rather than the data in the file section), the tape file system returns the RMS attributes to your process. For example, when you specify the DIRECTORY/FULL command for a volume, file, or list of files, the tape file system does the following:

1. Selects the file identifiers from the header labels
2. Returns the file attributes to your process
3. Positions the tape after the header labels of the last file accessed

A tape file opened for read access is closed in either of the following ways:

Method	Description
Implicitly	The file is closed implicitly when the drive encounters a tape mark while the system reads a file. The tape file system then reads the trailer labels, closes the file, and positions the tape at the next file.
Explicitly	The file is closed explicitly when you finish accessing the file before all the data in the file is read. The tape file system then closes the file without reading the trailer labels, and the tape remains at the current position.

Example

You can use the DCL command TYPE to read a file or group of files on the tape volume and to display the contents of the file on your terminal. For example, if you want to read the contents of a file named TESTFILE.DOC;1 (which you know from your directory searches is an OpenVMS file residing on the tape volume MTA1:), enter the following command:

```
$ TYPE MTA1:TEST*.%*;*
```

You then receive the following display on your terminal:

```
MTA1:TESTFILE.DOC;1

This is a test file.
```

9.5.5 Writing Files to Tape Volumes

When you write files to a tape volume, the tape file system performs access checks, writes labels, and, if necessary, switches volumes.

9.5.5.1 Writing New Files That Overwrite Existing Files

If a new file will overwrite an existing file, the tape file system does the following:

1. Checks the expiration date and accessibility fields of the existing file.
2. If overwriting is allowed, the tape file system does the following:
   a. Overwrites the header label set of the existing file
   b. Creates the file section
   c. Writes the trailer labels
   d. Writes two tape marks to denote the logical end-of-volume (EOV)

All files following the newly created file are lost.

To close a tape file that was opened for write access, the tape file system issues commands to the driver to write the labels, followed by a double tape mark that indicates the logical EOV.

9.5.5.2 Appending or Updating Files

When you use DCL to access an existing file for a write operation, either an append or an update operation is actually performed. Following is a description of each operation.

Access Method	Description
Append	When you access a file for an append operation, the tape is positioned *before* the tape mark that precedes the *trailer* labels. After the file is appended and closed, all files beyond the appended file are lost. When the positioning is complete, the processing is handled as if the file had been created.
Update	When you access a file for an update operation, the tape is positioned *after* the tape mark that follows the *header* labels. After the file is written to and closed, all files beyond the updated file are lost. The processing is handled as if the file had been created.

Note that you can update or append tape files only when the header label contains a value of 0 for the buffer offset length. For more information on how to update and append tape files, see Section 9.6.

If you do not specify the /OVERRIDE=EXPIRATION qualifier when you *update* a file, the tape file system checks the expiration date field on the file before it allows you to write to that file.

In addition, before you *append* a file, the tape file system checks the expiration dates of both the file being appended and the file immediately following. If the expiration date of either file has not been reached, the magnetic tape file system does not allow you to append the file.

Example

You can use the CREATE command to access a volume for a write operation. The following CREATE command writes a new file to the tape volume:

```
$ CREATE MTA0:MYFILE
```

After issuing a command like the one in this example, follow these steps:

1. Enter the contents of the file.
2. Press Ctrl/Z to close the file and write it to the tape volume without leaving the DCL command level.

9.6 Copying and Transferring Files

With the OpenVMS operating system, you can copy files on disks and tapes both within the system and across other operating systems. The OpenVMS operating system provides a number of facilities to assist you in both types of information transfer.

Table 9–7 summarizes the methods you can use to transfer information.

Table 9–7 Methods of Transferring Information

Method	Description
DCL command COPY	Most frequently used method for transferring information.
Convert utility (CONVERT)	On a local system, allows you to change the organization of a file from sequential to indexed, for example.
Exchange utility (EXCHANGE)	On a local system, allows you to access disk and tape volumes that are formatted for operating systems other than OpenVMS. You can use EXCHANGE to transfer files between foreign volumes and standard Files–11 volumes.
DCL command EXCHANGE /NETWORK	Allows you to transfer files via the network between OpenVMS and other operating systems. The command is useful for transferring files between nodes that use OpenVMS and those that do not. The file is copied in such a way that it is meaningful on OpenVMS and other operating systems as well.
Backup utility (BACKUP)	With tapes, the only means of copying entire directory trees or files that are not sequentially structured. See Section 10.13.2 for information on using BACKUP to copy files.

The COPY command, the Exchange utility, and the DCL command EXCHANGE /NETWORK are explained in the following sections.

In many cases, you can copy information without physically transporting media. Perhaps you want to copy files between systems that are not connected by a communications link. If so, you must be able to move your files physically from one location to another. A convenient way is to copy your files to a portable volume, such as a tape reel, tape cartridge, or disk pack, and then carry that volume to the location of the other system.

The following sections describe how to perform these tasks:

Task	Section
Copy files to disk volumes	Section 9.6.1
Copy files to tape volumes	Section 9.6.2
Continue to copy at the end of a tape	Section 9.6.3
Use the Exchange utility to copy files	Section 9.6.4
Use the DCL command EXCHANGE/NETWORK to transfer files over a network	Section 9.6.5

9.6.1 Copying Files to Disk Volumes

Before you can copy files to a disk volume, you must do the following:

1. Prepare the volume, as explained in Section 8.1.3.2.
2. Because disks are random-access devices, and because files must be listed in directories, you must create a directory to contain your files on the disk volume after you initialize the volume.

Copying from Disks

The default format for files on disk volumes is Files–11 Structure Level 2. You can also initialize disks in the Files–11 Structure Level 1 format, which is the format used by other Digital operating systems, including RSX–11M, RSX–11M–PLUS, RSX–11D, and IAS.

When you copy files from disks to standard-labeled disk volumes, the following items are not preserved:

- Directory specifications
- Individual file protections
- User identification codes (UICs)
- Creation times (but the dates are preserved)
- Revision and backup dates and times

You can use the COPY command to copy the highest version of all the files in your default directory to another directory on that volume.

Copying from Tapes

The default format for files on tapes is the standard-labeled volume. The OpenVMS system supports sequential, relative, and indexed files on disks, but you can copy only sequential files to standard-labeled disk volumes. The only valid record formats are variable-length and fixed-length.

When you copy files with tape file names from magnetic tape to disk, specify a standard OpenVMS file name for the output file name specification. If you do not specify an OpenVMS file name on output, your process receives the following error message:

```
RMS-F-FNM, error in file name
```

This message indicates that the tape file name is not a valid OpenVMS file name.

If you enter the COPY command with the /LOG qualifier, the system sends a message to the current SYS$OUTPUT device after each file has been copied. To verify that the files were successfully copied, use the DIRECTORY command.

Examples

1.
```
$ CREATE/DIRECTORY DMA3:[PUBS]
$ DEFINE P DMA3:[PUBS]
$ COPY *.* P
$ COPY [PRIMER]*.* P
$ COPY [COMMANDS]*.* P
```

   The CREATE/DIRECTORY command in this example creates a disk directory file named [PUBS] on DMA3:, and the DEFINE command defines the logical name P as DMA3:[PUBS]. The COPY command copies the highest version of each file in the current default directory and in the directories [PRIMER] and [COMMANDS] to the newly created directory.

2.
```
$ COPY *.* DMA5:[PRIVATE]
```

   For this example, assume that the disk device DMA5: has been allocated to your process and that a disk volume has been initialized and mounted on that device. Also assume that you have a directory called PRIVATE already created on that volume.

3.
```
$ COPY/LOG MTA1:"%&*?!SKI! """ SEASON.DAT
%COPY-S-COPIED, MTA1:[]"%&*?!SKI! """.;1
copied to WRKD:[MANUAL]SEASON.DAT;1 (120 records)
```

   The COPY/LOG command in this example copies the tape file %&*?!SKI!#" (# means space) to the file SEASON.DAT on the default disk and directory, WRKD:[MANUAL]. To copy the file to disk, you must you specify a new file name. (The OpenVMS software provides defaults for segments of the file specification that are not specified.)

   Because this example specifies the /LOG qualifier, the system returns a message that confirms the file was copied from the MTA1: tape volume; the message also tells how many records were copied.

4.
```
$ COPY/LOG MTA0:*.* *
%COPY-S-COPIED, MTA0:[]TASTETEST.DAT;1
copied to WRKD:[FOOD]TASTETEST.DAT;1 (249 records)
%COPY-S-COPIED, MTA0:[]ALLAT;1 copied to WRKD:[FOOD]ALALL;1 (48 records)
%COPY-S-NEWFILES, 2 files created
```

   In this example, the COPY/LOG command specifies wildcard characters for the file name and file type. Therefore, the system copies the only two files on the tape volume to the disk volume.

5.
```
$ COPY/LOG MTA1:*.* [EX]
%COPY-S-COPIED, MTA1:[].DAT;1 copied to WRKD:[EX]TEST.DAT21 records
%COPY-E-OPENOUT, error opening WRKD:[EX]"%&*()!SKI! """.;1 as output
-RMS-F-FNM, error in file name
%COPY-W-NOTCOPIED, MTA1:[]"%&*()!SKI! """.;1 not copied
%COPY-E-OPENOUT, error opening WRKD:[EX]"SANFRAN%%%""".;1 as output
-RMS-F-FNM, error in file name
%COPY-W-NOTCOPIED, MTA1:[]"SANFRAN%%%""".;1 not copied
%COPY-S-COPIED, MTA1:[]OPENVMS_LONG$FILE_NAME.LONG_EXT;1
copied to WRKD$:[EX]OPENVMS_LONG$FILE_NAME.LONG_EXT;1 (80 records)
%COPY-S-COPIED, MTA1:[]C6.JOU;1 copied to WRKD:[EX]C6.JOU;1 (4 records)
%COPY-S-NEWFILES, 2 files created
```

   The COPY/LOG command string specifies that all files on the volume mounted on tape volume MTA1: are to be copied to the current default disk and directory WRKD:[EX]. However, the system does not copy files with tape file names, but, instead, returns an error message.

9.6.2 Copying Files to Tape Volumes

You can use the COPY command to copy files from a disk volume to a tape volume. The procedures are similar to those for copying files from one disk volume to another. One difference, however, is that magnetic tapes are sequential-access devices and do not have directories. You must set up (initialize and mount) a tape device before copying disk files to a tape volume. (The characteristics of tape files are described in Section 9.5.)

The entire set of Files–11 file names is supported for magnetic tapes. You can copy a disk file with the following file name to a magnetic tape volume without having to modify the file name:

```
THIS_IS$AN_OPENVMSLONG_FILE.LONG_TYPE
```

Note

Most system that are not OpenVMS do not use file names longer than 17 characters.

Although the OpenVMS system supports stream and variable with fixed-length control (VFC) records, it encodes these records in a variable-length format on standard-labeled volumes. Systems that are not OpenVMS do not distinguish stream records or VFC records from variable-length records; instead, they interpret both as variable-length records. Therefore, do not create either stream or VFC records on volumes that will be used for information interchange to a system that is not OpenVMS.

The following steps show how to use DCL commands to copy files from a default directory on a disk volume to a standard-labeled magnetic tape volume. Included in the steps are examples showing how to allocate, initialize, and use a magnetic tape to copy a set of your disk files.

How to Perform This Task

To copy files from a default directory on a disk volume to a standard-labeled tape volume, follow these steps:

1. First, allocate a drive as follows:

   ```
   $ ALLOCATE MT: TAPE_DEVICE
   %DCL-I-ALLOC _MARS$MTA2: allocated
   ```

   This ALLOCATE command requests the allocation of a tape drive whose name begins with MT. The logical name TAPE_DEVICE in this case refers to the MARS$MTA2: drive.

   The system response indicates that unit 2 on controller A was available and is now allocated to you. You can now physically load the tape on the drive. Be sure the write ring on the tape is in place; if it is not, you cannot write to the tape.

2. Initialize the tape by entering a command like the following:

   ```
   $ INITIALIZE TAPE_DEVICE: GMB001/PROTECTION=(GROUP:R,WORLD)
   ```

   The INITIALIZE command specifies the logical name for the volume (TAPE_ DEVICE, which in this case refers to MTA2:) and the volume label for the tape volume (GMB001). The label can be no longer than six characters. The /PROTECTION qualifier defines a protection code restricting group access to read and allowing no world access.

3. Enter the MOUNT command to mount the volume and write files to it, as in the following example:

   ```
   $ MOUNT TAPE_DEVICE: GMB001
   %MOUNT-I-MOUNTED, GMB001 mounted on _MTA2:
   $ COPY *.* TAPE_DEVICE:
   ```

   The MOUNT command specifies the device name and volume label of the volume on the device. The COPY command copies the highest version of each file in your default directory onto the tape. The file names, file types, and version numbers of the output files default to the same file names, file types, and version numbers as the input files.

   If you enter the COPY command with the /LOG qualifier, the system sends a message to the current SYS$OUTPUT device after it copies each file.

4. You can also use the DIRECTORY command to verify that the files were copied successfully.

   ```
   $ DIRECTORY TAPE_DEVICE:
   ```

This DIRECTORY command lists the file names and file types of all files on the tape.

5. When you finish using the magnetic tape, dismount and deallocate it as follows:

```
$ DISMOUNT TAPE_DEVICE:
$ DEALLOCATE TAPE_DEVICE:
```

If you do not dismount and deallocate the magnetic tape, the system does so automatically when you log out.

The following examples illustrate ways of copying files to tape volumes.

Examples

1.
```
$ COPY *.* MTA2:
```

For this example, assume that MTA2: has been allocated to your process and that a tape volume has been initialized and mounted on that device. The COPY command writes files to the MTA2: tape volume.

The highest versions of all files in your default disk directory are copied to the tape volume. The file names, file types, and version numbers of the output files default to the same file names, file types, and version numbers as the input files.

2.
```
$ COPY/LOG FORTAP.DAT MTA1:"%&*?!SKI! "" "
%COPY-S-COPIED, WRKD:[MANUAL]FORTAP.DAT;1
copied to MTA1:[]"%&*?!SKI! """.;0 (120 records)
```

In this command for copying from disk to tape, a tape file name is specified as the output file specification. Note that the trailing space in the file name %&*?!SKI!#"# (where # means space) is not present because trailing spaces are not significant in tape names.

3.
```
$ COPY/LOG OPENVMS_LONG$FILE_NAME.LONG_EXT MTA1:
%COPY-S-COPIED, WRKD:[MANUAL]OPENVMS_LONG$FILE_NAME_EXT;1
copied to MTA1:OPENVMS_LONG$FILE_NAME.LONG_EXT;1 (80 records)
```

In this example, a long file name with a long file type is copied to the tape volume MTA1: with the same file name and type as on the disk volume.

4.
```
$ COPY/LOG %%.JOU;* MTA1:*.*
%COPY-S-COPIED, WRKD:[MANUAL]C6.JOU;1 copied to MTA1:[]C6.JOU;1 (4 records)
```

In this example, all files with a two-character file name and a file type of .JOU are copied to the tape volume MTA1: with the same file name and type as on the disk volume. Version numbers are preserved.

9.6.3 Continuing to Copy at the End of a Tape

When you are copying to or from a tape and that tape reaches the end, the system suspends processing and sends a request to mount the next tape in the volume set. An operator communication manager (OPCOM) message similar to the following is displayed at the terminal:

```
%%%%%%%%%%  OPCOM, 14-MAY-1995 15:23:31.78  %%%%%%%%%%
request 3, from user PLAW
MOUNT new relative volume 2 (DW0QT2) on MTA1:
```

Note

Because messages are sent only to the operator's terminal that is enabled for tape messages, you do not usually see this message and might not realize that another tape is needed to complete the read or write operation.

See Section 18.6 for more information on OPCOM messages.

If automatic volume switching is disabled or if the tape file system cannot mount a given volume, you might need to mount a continuation volume in a volume set. See Section 8.8.2 for information on mounting continuation volumes.

9.6.4 Using the Exchange Utility (EXCHANGE)

The Exchange utility (EXCHANGE) converts the format of files, as appropriate, when copying files between volumes of different structures. EXCHANGE recognizes all Files–11 and RT–11 disk volumes on OpenVMS devices, as well as all DOS–11 and RT–11 formatted volumes on 9-track tape devices.

For more information on how to use EXCHANGE and for a description of all EXCHANGE commands, qualifiers, and parameters, see the *OpenVMS Exchange Utility Manual*.

9.6.5 Using the EXCHANGE/NETWORK Command

You can use the DCL command EXCHANGE/NETWORK to transfer files to and from operating systems that do not support OpenVMS file organizations. This transfer occurs over a DECnet communications link that connects nodes that are both OpenVMS and not OpenVMS operating systems.

Using the EXCHANGE/NETWORK command, you can perform the following operations:

- Transfer files between an OpenVMS node and a node that is not OpenVMS.
- Transfer a group of input files to a group of output files.
- Transfer files between two nodes that are not OpenVMS, provided those nodes share DECnet connections with the OpenVMS node that issues the EXCHANGE/NETWORK command.

For details on using the EXCHANGE/NETWORK command, refer to online help or the *OpenVMS DCL Dictionary*.

How to Perform This Task

To issue the EXCHANGE/NETWORK command, use the following format:

EXCHANGE/NETWORK input=filespec[,...] output-filespec

where:

input-filespec	Specifies the name of an existing file to be transferred. (Wildcard characters are allowed.)
output-filespec	Specifies the name of the output file into which the input is to be transferred.

Example

```
$ EXCHANGE/NETWORK MYSYS_FILE.DAT FOO::FOREIGN_SYS.DAT
```

The command in this example transfers the file MYSYS_FILE.DAT, which is located in the current default device and directory, to the file FOREIGN_SYS.DAT on node FOO, which is not an OpenVMS node. By default, the command automatically determines whether the transfer method should be block or record I/O.

10
Using BACKUP

You can guard against data loss or corruption by using the OpenVMS Backup utility (BACKUP) to create copies of your files, directories, and disks. In case of a problem—for example, a disk drive failure—you can restore the backup copy and continue your work with minimal disruption.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Formulating a BACKUP strategy	Section 10.3
Setting up a BACKUP account	Section 10.7
Initializing, mounting, and dismounting disks and tapes	Section 10.8
Listing the contents of a BACKUP save set	Section 10.10
Backing up user disks and volume shadow sets	Section 10.15
Restoring user disks and volume shadow sets	Section 10.16
Backing up and restoring the system disk	Section 10.17
Building and booting standalone BACKUP	Section 10.17.2
Ensuring data integrity	Section 10.18
Troubleshooting	Section 10.19

This chapter explains the following concepts:

Concept	Section
Types of backups	Section 10.2
The BACKUP command line	Section 10.4.1
The Backup Manager	Section 10.4.2
Save sets	Section 10.5
BACKUP file formats	Section 10.6
Volume initialization	Section 10.8.1
OPCOM and volumes	Section 10.9
Multivolume BACKUP operations	Section 10.11
BACKUP tape label processing	Section 10.12
Standalone BACKUP	Section 10.17.2

10.1 Overview of BACKUP Tasks

For BACKUP effectively guard against data loss, you must back up important data on a regular basis and be familiar with how to restore the data when necessary.

Besides backing up your own files, directories, and disks, you should back up your system disk. If you have a standalone workstation, backing up the system disk is probably your responsibility. If your system is part of a large clustered computer system, an operator or system manager is probably responsible for backing up the system disk.

The two ways to back up your system disk are:

- Use the menu-driven procedure described in Section 10.17.1.
- Use a special version of the OpenVMS Backup utility called standalone BACKUP, described in Section 10.17.2. Use standalone BACKUP if you do not have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc.

Note

Standalone BACKUP is not supported on OpenVMS Alpha systems beginning with Version 6.1; you must use the menu-driven procedure.

Performing an image backup using BACKUP also eliminates disk fragmentation. Fragmentation can occur as you create and extend files on a disk. If the file system cannot store files in contiguous blocks, it stores them in noncontiguous pieces. Eventually, the disk can become severely fragmented and system performance suffers.

To eliminate fragmentation, perform an image backup of the disk and restore the backup copy. When you restore the image backup, BACKUP places the files on the disk contiguously. Alternatively, you can perform a disk-to-disk image backup without using the /SAVE_SET qualifier. This creates a functionally equivalent copy of the entire system disk, on which files are stored contiguously.

Note

Some layered products have their own special backup procedures. For more information, see the layered product documentation.

10.2 Understanding Types of Backups

The following table lists the types of backup operations.

Operation	Description
File operation	Processes individual files or directories. Section 10.13 describes file operations.
Selective operation	Processes files or volumes selectively, according to criteria such as version number, file type, UIC, date and time of creation, expiration date, or modification date. Perform selective save operations by using wildcard characters and input file-selection qualifiers (for example, /BACKUP, /BEFORE, /BY_OWNER [/OWNER_UIC], /CREATED, /EXCLUDE, /EXPIRED, /MODIFIED, and /SINCE). Section 10.13 describes selective operations.
Physical operation	Copies, saves, restores, or compares an entire volume in terms of logical blocks, ignoring any file structure.
Image operation	Processes all files on the input disk. The types of image operations are: – An **image backup** (also called a full backup) saves a copy of all the files on a disk (or volume) to a special file called a **save set**. The first backup that you do on a disk must be an image backup; you cannot perform an incremental backup first. – An **image restore** initializes the output disk and restores an entire volume. – An **image copy** operation initializes the output disk and copies an entire volume; the image backup is a logical duplicate of the contents of the disk. – An **image compare** operation compares the contents of entire volumes. Because an image copy or backup operation processes all files on the input volume, you cannot specify file-selection qualifiers for these operations. You can, however, restore files and directories selectively from an image save set.
Incremental operation	The two types of incremental operations are: – An **incremental backup** saves only those files that have been created or modified since the most recent backup that was performed using the /RECORD qualifier. (The /RECORD qualifier records the date and time that the files are backed up.) – An **incremental restore** operation restores an incremental save set. Specify the command qualifier /INCREMENTAL in an incremental restore operation. Section 10.16.2 describes incremental restore operations.

10.3 Formulating a Backup Strategy

When formulating a backup strategy, keep in mind the specific requirements of your site and the advantages and disadvantages of the different types of backups. Your backup strategy also depends on the following factors:

- The resources you can devote to backups
- The importance of the data
- The volatility of the data

For example, if you have a standalone workstation, you might conclude that a nightly image backup is your best approach.

Under other circumstances, you may want to choose some combination of image and incremental backups. For example, daily image backups might be inconvenient if your system always has interactive users logged in (see Section 10.15.1). You could choose to perform a weekly image backup and nightly incremental backups.

Table 10–1 compares image and incremental backups.

Table 10–1 Comparison of Image and Incremental Backups

Backup Type	Advantages	Disadvantages
Image	Faster to restore than incremental backups. Backs up entire disk.	Uses more space and time than incremental backups. Requires that no interactive users are logged in because of the effect on system performance and because of open file considerations (see Section 10.15.1).
Incremental	Takes less time and media storage space.	More difficult to restore files. Still requires periodic image backups.

You do not have to change tapes or disks during a backup if any of the following are true:

- All of the files fit on a single piece of storage media.
- Your site uses a tape loader.
- You have several disk or tape drives available.

In these cases, the backup can be performed by a batch job that runs late at night or at some other time when interactive use of the system is likely to be at a minimum. Section 10.15.6 contains some sample command procedures that you can run in a batch job to back up your disks.

10.4 Understanding the Backup Interfaces

Two interfaces are available to the OpenVMS Backup utility:

- The BACKUP command, which is a command in the DCL command line interface
- The Backup Manager, which is an interactive screen-oriented interface

10.4.1 The BACKUP Command Line

To back up files, you must specify what you want to back up (the input) and where you want BACKUP to place the resultant save set or file (the output). You can also use BACKUP qualifiers to perform different functions depending upon their position on the command line:

```
BACKUP/qualifiers input_specifier/qualifiers output_specifier/qualifiers
```

Table 10–2 lists the types of BACKUP command qualifiers.

Table 10–2 BACKUP Command Qualifier Types

Type	Position	Effect
Command qualifier	Anywhere on the command line	Affects both input and output specifiers.
Input specifier qualifier	Directly after the input specifier	Affects only the input specifier.
Output specifier qualifier	Directly after the output specifier	Affects only the output specifiers.

When you use BACKUP, make sure you place BACKUP qualifiers in their correct positions on the command line. For more information about the BACKUP command line, see *OpenVMS System Management Utilities Reference Manual*.

10.4.2 The Backup Manager

Backup Manager is a screen-oriented interface to the OpenVMS Backup utility (BACKUP) that presents BACKUP's capabilities in an intuitive, task-oriented, self-documenting manner. Backup Manager can ease backup tasks by guiding you through the backup process. No real performance differences exist between using the Backup Manager and using the BACKUP command line.

Backup Manager runs on:

- Any Digital Equipment Corporation VT*xxx*-series video display terminal or equivalent terminal emulator
- VMS VAX Version 5.4 or higher systems, and OpenVMS Alpha Version 1.5 or higher systems

The Backup Manager interface is based on the OpenVMS Screen Management Run-Time Library (RTL) routines.

10.4.2.1 Backup Manager Features

Backup Manager can perform the following backup operations:

- Save an entire volume to a save set
- Save selected files from a volume to a save set
- Restore an entire volume from a save set
- Restore selected files from a save set
- List a save set

Three types of online assistance are available with Backup Manager:

- Context-sensitive help

You can press PF2 or the Help key to get help about the object at which the display cursor is currently located.

- Context-sensitive hints

  You are prompted for input by a one-line ‘‘hint’’ about the field where the display cursor is currently located.

- Pull-down help

  You can choose the pull-down Help menu bar item for more extensive help on a variety of Backup Manager topics.

10.4.2.2 Getting Started with Backup Manager

To start Backup Manager, enter the following command at the DCL prompt:

```
$ RUN SYS$SYSTEM:BACKUP$MANAGER
```

Output from the Backup utility is automatically displayed when an operation starts. You can suspend output at any time (Ctrl/P) and scroll through it. You can also use Ctrl/T to display status or Ctrl/C to stop the current BACKUP operation.

10.5 Understanding Save Sets

When you enter a BACKUP command to save files to a tape, BACKUP writes the files to a special file called a save set. You can also create a save set on a disk using the /SAVE_SET qualifier. Save sets are classified according to the media on which they reside. Table 10–3 lists the types of media that you write a save set to.

Table 10–3 Save-Set Types

Media Type	For More Information
Magnetic tape	Section 10.5.1
Files–11 disk	Section 10.5.2
Files–11 disk on a remote node (network save set)	Section 10.5.3
Sequential disk	Section 10.5.4

10.5.1 Magnetic-Tape Save Sets

Magnetic tape is the most commonly used media for storing BACKUP save sets. It is less expensive than disk media, and its compact size makes it easy to store. You can use more than one tape device at a time to save or restore data; this allows processing to continue on another tape while the one most recently used is rewinding.

BACKUP treats all magnetic tape files as BACKUP save sets. Because you cannot use save-set specifications as both the input and output specifiers in a BACKUP command line, you cannot perform a BACKUP operation from one magnetic tape to another.

Digital recommends that you copy magnetic-tape save sets to disk with the BACKUP command; however, you can use the DCL command COPY on magnetic-tape save sets that were created with the /INTERCHANGE qualifier.

Save-set specifications on magnetic tape are limited to 17 characters, including the period delimiter (.) and file type. The following is a valid save-set specification:

WKLY27JAN1995.BCK

When restoring data from tape, if you do not include a save-set name with an input magnetic tape, BACKUP reads the next save set it encounters on the tape. (If you specify the input save-set qualifier /REWIND, BACKUP rewinds the tape and reads the first save set on the tape.)

10.5.2 Files–11 Disk Save Sets

To write save sets on a Files–11 disk, you must include the output save-set qualifier /SAVE_SET. The /SAVE_SET qualifier indicates to BACKUP that you want to create a save set, rather than a copy of the selected files, on the output volume. The disk must be mounted as a Files–11 volume; all volumes in a volume set must be mounted.

BACKUP can read a Files–11 save set as a Files–11 save set or as a sequential-disk save set:

- When BACKUP reads a save set as a Files–11 save set, all volumes of the save set must be mounted. To read a save set that is not located in your process default directory, you must specify the directory in which the save set is located.
- When BACKUP reads a Files–11 save set as a sequential-disk save set, you can mount the volumes one at a time. You must specify the master file directory [000000] in the save-set specification when reading a Files–11 save set as a sequential-disk save set.

A save set stored on a Files–11 disk is a standard file, however, and can be copied, renamed, deleted, or backed up.

10.5.3 Network Save Sets

You can create or read a network save set on a Files–11 disk attached to a remote node by specifying the node name of a remote node in the save-set specification. A **remote node** is a node that is accessible to the node you are working on (the host node) over a network. The network save set must be located on a publicly accessible disk (a disk mounted from the remote node with the /SYSTEM, /GROUP, or /CLUSTER qualifier) on the remote node.

Depending on the volume and file protection at the remote node, you may need to specify an access control string in the network save-set specification. An access control string includes the user name and password, and has the following format:

remote_nodename"username password"::device_name:[directory]

Example

The following example creates a network save set on the remote node DOUBLE:

```
$ BACKUP
_FROM: [MY_DIR]
_TO: DOUBLE"username password"::DBA0:SAVEIT.BCK/SAVE_SET
```

Omit the access control string if it is not required to gain access to the remote node, such as in the case of proxy network access. See the *DECnet for OpenVMS Networking Manual* for more information about access control strings and proxy network access.

10.5.4 Sequential-Disk Save Sets

Sequential-disk save sets allow you to treat a Files–11 disk volume sequentially, (like a magnetic tape volume). The primary advantage of using sequential-disk save sets is that you can mount multivolume save sets one volume at a time. This is particularly useful on systems without tape drives that have a large fixed-media disk and a small removable disk.

When one sequential disk is full, BACKUP prompts you to mount another disk. You can use more than one disk device at a time to save or restore data; this allows processing to continue on another disk while the one most recently used is spinning down.

You must have the privilege LOG_IO or PHY_IO to read or write a multivolume sequential-disk save set.

Before creating a sequential-disk save set, mount the first volume of the sequential-disk save set using the DCL command MOUNT/FOREIGN. Although the disk is mounted with the /FOREIGN qualifier, BACKUP manages the disk using Files–11 structure.

When you perform a save operation to a sequential disk, you must use the output save-set qualifier /SAVE_SET. When you perform a restore operation from a sequential disk, you must specify the input save-set qualifier /SAVE_SET. If you do not specify the /SAVE_SET qualifier, BACKUP displays the following error message:

```
%BACKUP-F-IMGFILSPE, /IMAGE specification must only have device name
```

Do not specify a directory name for the save set; sequential-disk save sets are always entered in the master file directory [000000]. Even if you specify a directory other than the master file directory in a save operation, the save set is entered in the master file directory. If you specify a directory other than the master file directory in a restore or list operation, BACKUP returns an error message indicating that it cannot locate the file.

BACKUP does not initialize the first sequential-disk volume because the default is /NOINITIALIZE; however, continuation volumes are initialized. Unless you specify the command qualifier /INITIALIZE, the following restrictions apply to the first sequential-disk volume:

- The disk must be Files–11 Structure Level 2.
- The disk must not be part of a volume set.
- The cluster factor of the disk must be 1.
- The free space on the disk cannot be fragmented into more than 100 contiguous extents.
- The index file cannot be extended.
- The master file directory cannot be extended.

Volumes you use for sequential-disk save sets should contain only save sets. You must initialize a volume that has been used for general file processing before using it as a sequential-disk volume. You can place a maximum of 12 save sets on a single sequential disk. Use Files–11 disk save sets if you want to create more than 12 save sets on a single disk.

BACKUP can read a sequential-disk save set either as a sequential-disk save set or as a Files–11 save set:

- When BACKUP reads a save set as a sequential-disk save set, the save set can be mounted one volume at a time. The default directory for the save set file specification is the master file directory [000000] on the disk.
- When BACKUP reads a save set as a Files–11 save set, all volumes of the save set must be mounted. The default directory is your process default directory. Therefore, you must specify the master file directory [000000] in order to read a sequential-disk save set as a Files–11 save set.

10.6 Understanding BACKUP File Formats

VAX

On VAX systems, BACKUP saves files and directories from Files–11 Structure Level 1 and 2 disks to disks or magnetic tapes. If necessary, you can use BACKUP to restore the saved files and directories to Files–11 Structure Level 1 and 2 disks.

If a VAX system performs image backup of an Alpha system disk, a restore operation causes the Alpha system to reboot successfully. ♦

Alpha

On Alpha systems, BACKUP saves files and directories from Files–11 Structure Level 2 disks to disks or magnetic tapes. If necessary, you can use BACKUP to restore the saved files and directories to Files–11 Structure Level 2 disks.

Note

The OpenVMS Alpha operating system does not support the Files–11 Structure Level 1 format.♦

You cannot back up files on ISO 9660-formatted media, but you can restore save sets stored on ISO 9660-formatted media.

For more information about the Files–11 disk structure, see Section 8.1.1.2. For more information about ISO 9660 devices, see Section 7.2.2.

10.7 Setting Process Quotas for Efficient Backups

You can optimize the efficiency of backups on your system by properly setting the process quotas for the process from which backups will be made (the process from which you enter the BACKUP command or submit your backup command procedure). This is especially important if you are using a streaming tape drive.

How to Perform This Task

To set process quotas for efficient backups, do the following:

1. Determine which account you want to use for backups. You can use an existing account, or create one solely for backups. For information about creating accounts, see Section 6.6.
2. Use the Authorize utility (AUTHORIZE) to determine the current quota values for for the account you will use for backups. For example, if you are using the SYSTEM account for backups, enter the following commands:

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

3. Using the System Management utility (SYSMAN), determine the value of the system parameters WSMAX and CHANNELCNT, as follows:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAMETERS SHOW WSMAX
%SYSMAN-I-USEACTNOD, a USE ACTIVE has been defaulted on node DIEM
Node DIEM:   Parameters in use: ACTIVE
Parameter Name          Current   Default   Minimum   Maximum Unit  Dynamic
--------------          -------   -------   -------   ------- ----  -------
WSMAX                      2600      1024        60    100000 Pages
SYSMAN> PARAMETERS SHOW CHANNELCNT

Parameter Name          Current   Default   Minimum   Maximum Unit  Dynamic
--------------          -------   -------   -------   ------- ----  -------
CHANNELCNT                 127       127        21     2047 Channels

SYSMAN> EXIT
$
```

   In this case, the values for WSMAX and CHANNELCNT, as shown in the column marked Current, are 2600 and 127, respectively. Use these values to help set the correct values for the process quotas.

4. Use AUTHORIZE to compare the values for the process quotas to the recommended values for efficient backups, as shown in Table 10–4.

Table 10–4 Recommended Process Quotas for Efficient Backups

Process Quota	Recommended Setting
WSQUOTA	Equal to system parameter WSMAX
WSEXTENT	Equal to WSQUOTA quota
PGFLQUOTA	Equal to or greater than WSEXTENT quota
FILLM	Less than system parameter CHANNELCNT
DIOLM	Either 4096 or three times the value of FILLM quota, whichever is *greater*
ASTLM	Either 4096, at least 100 greater than DIOLM quota, or three times the value of FILLM quota, whichever is *greater*
BIOLM	Equal to FILLM
BYTLM	Equal to or greater than the following value: $(256 * FILLM) + (6 * DIOLM)$
ENQLM	Greater than FILLM quota

5. If necessary, change process quotas using the AUTHORIZE command MODIFY. If you change process quotas, you must log out and log in again for the changes to take effect. For information on changing these process quotas, see Section 6.7.2.

   Table 10–5 lists a set of process quota values that are appropriate for many configurations. If your disks are highly fragmented or if your backups will be performed during periods of heavy system use, you should reduce the values shown for WSQUOTA and FILLM.

Table 10–5 Sample Process Quotas for Efficient Backups

Process Quota	Suggested Value
WSQUOTA	16384
WSEXTENT	Greater than or equal to WSQUOTA
PGFLQUOTA	32768
FILLM	128
DIOLM	4096
ASTLM	4096
BIOLM	128
BYTLM	65536
ENQLM	256

Example

The following steps show the commands that you would use to run the Authorize utility and set process quotas for the SYSTEM account (if you plan to run backups from a different account, determine the process quotas for that account):

1. Determine the current quota values:

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

Username: SYSTEM                          Owner:  SYSTEM MANAGER
Account:  SYSTEM                          UIC:    [1,4] ([SYSTEM])
CLI:      DCL                             Tables: DCLTABLES
Default:  SYS$SYSROOT:[SYSMGR]
                                    .
                                    .
                                    .
Maxjobs:         0  Fillm:        40  Bytlm:        32768
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:           0
Maxdetach:       0  BIOlm:        18  JTquota:       1024
Prclm:          10  DIOlm:        18  WSdef:          256
Prio:            4  ASTlm:        24  WSquo:          512
Queprio:         0  TQElm:        20  WSextent:      2048
CPU:        (none)  Enqlm:       200  Pgflquo:      20480
                                    .
                                    .
                                    .
UAF> EXIT

%UAF-I-NOMODS, no modifications made to system authorization file
%UAF-I-NAFNOMODS, no modifications made to network authorization file
%UAF-I-RDBNOMODS, no modifications made to rights database
$
```

In this example, SYSTEM has the following quotas:

WSQUOTA	512
WSEXTENT	2048
PGFLQUOTA	20480
FILLM	40
DIOLM	18

ASTLM	24
BIOLM	18
BYTLM	32768
ENQLM	200

2. Using the System Management utility (SYSMAN), determine the value of the system parameters WSMAX and CHANNELCNT, as follows:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAMETERS SHOW WSMAX
%SYSMAN-I-USEACTNOD, a USE ACTIVE has been defaulted on node DIEM
Node DIEM:   Parameters in use: ACTIVE
Parameter Name          Current   Default   Minimum   Maximum Unit  Dynamic
--------------          -------   -------   -------   ------- ----  -------
WSMAX                      2600      1024        60    100000 Pages
SYSMAN> PARAMETERS SHOW CHANNELCNT

Parameter Name          Current   Default   Minimum   Maximum Unit  Dynamic
--------------          -------   -------   -------   ------- ----  -------
CHANNELCNT                 127       127        21     2047 Channels

SYSMAN> EXIT
$
```

   In this case, the values for WSMAX and CHANNELCNT, as shown in the column marked Current, are 2600 and 127, respectively.

3. Compare the values for SYSTEM to the values in Table 10–4, and set the appropriate values:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> MODIFY SYSTEM/WSQUOTA=2600
UAF> MODIFY SYSTEM/WSEXTENT=2600
UAF> MODIFY SYSTEM/DIOLM=4096
UAF> MODIFY SYSTEM/ASTLM=4096
UAF> MODIFY SYSTEM/BIOLM=40
UAF> MODIFY SYSTEM/BYTLM=34816
UAF> EXIT
```

4. Log out and then log in again so that these process quotas take effect.

10.8 Using Disks and Tapes

During the course of your backup operations, you will use both disk and tape volumes. The four steps you normally perform before using a volume in a backup operation are:

1. Determine the device name.
2. Allocate the device.
3. Initialize the volume (optional).
4. Mount the device (for disks only; BACKUP mounts tapes automatically).

These tasks are described in Chapter 8. This chapter describes specifically how these tasks relate to BACKUP. Note that all disk operations in this chapter also apply to diskettes.

10.8.1 Understanding Volume Initialization

Initializing a volume does the following:

- Formats it in the OpenVMS Files–11 format
- Assigns it an ANSI label
- Removes links to any existing files (effectively erasing them)
- Writes a tape expiration date and volume protection data to the volume header record of the tape

Caution

Initializing a volume removes links to existing files on the volume, effectively erasing the files. Do not initialize a volume that contains data you want to keep.

10.8.1.1 When to Initialize Volumes

You must initialize a volume for use with BACKUP if any of the following conditions exist:

- The volume is new and has not been formatted in the Files–11 format.
- You want to remove access to data stored on the volume.
- You want to change the volume label, expiration date, or volume protection data.
- The volume contains a non-ANSI or non-ISO label.

Table 10–6 show the three ways to initialize a volume.

Table 10–6 Methods of Volume Initialization

Method	For More Information
Before a backup operation with the DCL command INITIALIZE	Section 8.3
On the BACKUP command line with the /REWIND qualifier (for tapes only)	Section 10.8.1.2
On the BACKUP command line with the /INITIALIZE qualifier (for disks only)	Section 10.8.1.3

10.8.1.2 Initializing Tapes

Instead of using the INITIALIZE command and then performing a backup operation, you can initialize a tape and perform a backup operation by entering one BACKUP command.

How to Perform This Task

To initialize a tape volume on the BACKUP command line, add the /REWIND and /LABEL qualifiers to the output specifier. The /REWIND qualifier rewinds and initializes the volume. The /LABEL qualifier allows you to specify the volume label.

Magnetic tape volume labels can contain a maximum of six characters. You can use any ANSI ‘‘a’’ character in a magnetic tape volume label. The ANSI ‘‘a’’ characters include numbers, uppercase letters, and any of the following nonalphanumeric characters:

! " % ' ( ) * + , _ . / : ; < = > ?

If you use any of the nonalphanumeric characters, you must enclose the volume label with quotation marks.

Label your magnetic tapes according to the data contained on the tapes. The following table presents some suggestions for labeling tapes:

Label	Type of Backup	Expiration Date
DLY101	Daily, group 1, volume number 1	Expires in 7 days
DLY102	Daily, group 1, volume number 2	Expires in 7 days
WKY101	Weekly, group 1, volume number 1	Expires in 4 weeks
WKY201	Weekly, group 2, volume number 1	Expires in 4 weeks
MTH101	Monthly, group 1, volume number 1	Expires in 12 months
YRY101	Yearly, group 1, volume number 1	Expires in 5 years

Note that:

- If the tape volume has already been initialized with a label that is different from the label you specify on the BACKUP command line, BACKUP displays an error message about the label mismatch. For more information, see Section 10.12.
- If the tape is not expired, BACKUP displays the following error message:

  ```
  %INIT-F-FILNOTEXP, file is not expired
  ```

  If you have either VOLPRO privilege or write access to the volume, or you are the owner of the volume, you can use the DCL command INITIALIZE /OVERRIDE=EXPIRATION to initialize the magnetic tape.

  You can also reenter the BACKUP command line using the /IGNORE=LABEL_PROCESSING qualifier (see the *OpenVMS System Management Utilities Reference Manual*).
- If the volume was previously initialized with the output save-set qualifiers /REWIND and /PROTECTION, you must either own the volume (your UIC matches the owner UIC of the volume) or have VOLPRO privilege.

Example

```
$ BACKUP [ACCOUNTS.JUNE] MUA0:JUNE.BCK/REWIND/LABEL=MTH101
```

10.8.1.3 Initializing Disks

Instead of using the INITIALIZE command and then performing a backup operation, you can initialize a disk and perform a backup operation by entering one BACKUP command.

How to Perform This Task

The two ways to initialize a disk during a backup operation are:

- When you perform an image copy to disk, BACKUP automatically initializes the output disk, effectively erasing any existing files and volume-initialization data on the disk. To preserve volume-initialization data on the output disk, use the /NOINITIALIZE qualifier.
- When you create a sequential disk save set, BACKUP does not initialize the output volume (by default). You can, however, instruct BACKUP to initialize the output volume using the /INITIALIZE qualifier.

Examples

1. The following command shows how to initialize a disk on the BACKUP command line:

   ```
   $ BACKUP/IMAGE DUA1: DUA2:
   ```

   This command initializes DUA2: using the volume-initialization data from DUA1. BACKUP then copies the contents of DUA1: to DUA2:, effectively erasing any existing files on DUA2. Note that the files on DUA2: are stored contiguously, eliminating disk fragmentation.

2. The following command shows how to preserve volume-initialization data on the output disk during an image copy:

   ```
   $ BACKUP/IMAGE DUA1: DUA2:/NOINITIALIZE
   ```

   This command causes BACKUP to initialize DUA2:, preserving the initialization data on that volume. BACKUP then copies the contents of DUA1: to DUA2:, effectively erasing any existing files on DUA2.

3. These commands cause BACKUP to initialize DJA2:, effectively erasing any existing files:

   ```
   $ MOUNT/FOREIGN DJA2:
   %MOUNT-I-MOUNTED, USER1 mounted on _DJA2:
   $ BACKUP/IMAGE DUA1: DJA2:DAILY.SAV/INITIALIZE
   ```

   BACKUP then creates an image backup of DUA1: in the sequential disk save set DUA2:[000000]DAILY.SAV. If the save set exceeds the available disk space, BACKUP prompts for another volume. BACKUP initializes the new volume and extends the save set in the master file directory ([000000]) of the new volume. (For more information about save sets, see Section 10.5. For more information about the /INITIALIZE qualifier, see the *OpenVMS System Management Utilities Reference Manual*.)

10.8.2 Mounting a Volume

Mounting a volume makes it available to the system. BACKUP automatically mounts tapes when you use them for a backup operation. Most disks on your system are mounted at system startup. This section describes how to explicitly mount volumes.

If you are planning to write a save set to a disk, decide whether the save set will be written in standard Files–11 format or in sequential-disk format:

- If the save set will be written in standard Files–11 format, the target disk must be mounted as a Files–11 disk.

- If a save set will be written in sequential-disk format (for example, if the save set occupies more than one disk), the target disk must be mounted as a foreign device by specifying the command qualifier /FOREIGN to the DCL command MOUNT.

How to Perform This Task

1. Enter the SHOW DEVICE command in the following format to check whether the device is already mounted:

   SHOW DEVICE device_name

2. Enter the MOUNT command in the following format:

   MOUNT [/FOREIGN] device-name [volume-label] [logical-name]

   where:

device-name	is the name of the drive that holds the volume you want to mount.
volume-label	is the alphanumeric identification you assigned to the volume with the INITIALIZE command. For disk volumes, labels can have a maximum of 12 characters; for magnetic tape volumes, labels can have a maximum of 6 characters. You do not need to add this parameter if you are mounting the volume with the /FOREIGN qualifier.
logical-name	is an optional 1- to 255-character alphanumeric specification that you want to associate with the volume.

Example

```
$ SHOW DEVICE MU

Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
DAD$MUA6:               Online               0
MOM$MUA6:               Online               0
FRED$MUA6:              Online               0
$ MOUNT/FOREIGN FRED$MUA6: TEST DRIVE1
%MOUNT-I-MOUNTED, TEST mounted on _FRED$MUA6:
```

This command mounts the tape in FRED$MUA6: and assigns it the logical name DRIVE1.

10.8.3 Dismounting a Volume

BACKUP does not dismount the last volume of a backup operation (unless you use the /RELEASE_TAPE qualifier). When you finish using a volume, you should dismount it.

How to Perform This Task

Enter the DISMOUNT command in the following format:

DISMOUNT device-name

Example

The following command dismounts a tape in drive MUB6:

```
$ DISMOUNT MUB6:
```

This command dismounts and unloads the tape in MUB6. After you dismount and unload the volume, you can remove it from the drive. If you want to dismount the tape but not unload it, enter the following command:

```
$ DISMOUNT/NOUNLOAD MUB6:
```

10.9 Understanding OPCOM and Volumes

If you have a standalone workstation or easy access to disk and tape drives at your facility, you probably can mount and initialize your own volumes. At some sites, however, an operator performs these tasks. Using the services of an operator might be necessary because the drive you want to use is located remotely or because you do not have the necessary privileges to manipulate a volume.

To communicate with the operator at your site, consult the operator about site-specific procedures. Depending on how your system is customized, using the operator communication manager (OPCOM) might be necessary. The OPCOM system process allows you to request assistance from the operator and allows the operator to respond to your requests.

Section 2.4 explains OPCOM.

10.9.1 Requesting Operator Assistance

Note

Please consult your operator about your site-specific procedures. Your site may not use OPCOM or may use it differently than the examples in this section demonstrate.

If you want the operator to mount a tape for you use OPCOM to ask the operator to mount the tape.

How to Perform This Task

Enter either the REQUEST/REPLY or the REQUEST/TO command:

- The /REPLY qualifier assigns your request a unique number to which the operator can respond.
- If your facility is very large, several operators might each have specific tasks. If this is the case, use the REQUEST/TO command, which allows you to send a message to a specific operator (identified by a keyword).

If you request operator assistance and an operator is not available, you receive the following message:

```
%MOUNT-I-NOOPR, no operator available to service request
```

This indicates that the operator has disabled the operator's terminal. To abort your request, press Ctrl/Z.

You can also use the /[NO]ASSIST qualifier with either the BACKUP or the MOUNT command:

- If a mount request fails and you specified /ASSIST, mount failure messages appear on the operator terminal (if OPCOM is enabled). The /ASSIST qualifier is the default for both the BACKUP and MOUNT commands.
- If you specified /NOASSIST, mount failure messages appear on your terminal instead of on the operator terminal.
- If you are on a workstation but forget to specify /NOASSIST, OPCOM (if OPCOM is running) requests that the operator load the next volume.

If you have the OPER privilege, you can respond to the request by using another terminal window to enter the following commands:

```
$ REPLY/ENABLE=TAPES
$ REPLY/TO=identification-number "message text"
```

Examples

1. To request the operator to mount a tape, enter a command similar to the following :

   ```
   $ REQUEST/REPLY "Is anyone using drive MUA12?"
   %OPCOM-S-OPRNOTIF, operator notified, waiting...12:21:12.46
   %OPCOM-S-OPREPLY, PLEASE DIRECT YOUR REQUEST TO THE TAPE OPERATOR
   2-APR-1995 12:26:13.12. request 2 completed by operator OPA0
   $
   ```

   The /REPLY qualifier assigns your request a unique number (in this case, 2) to which the operator can respond. Note that you cannot enter any additional commands until the operator responds.

2. The following example shows you how to direct your request to a specific operator using the /TO qualifier:

   ```
   $ REQUEST/TO=TAPES "Is anyone using drive MUA12?"
   %OPCOM-S-OPRNOTIF, operator notified, waiting...12:40:11.32
   %OPCOM-S-OPREPLY, I'M DONE GO AHEAD
   2-APR-1995 12:45:26.18. request 5 completed by operator OPA0
   $
   ```

10.10 Listing the Contents of a BACKUP Save Set

BACKUP allows you to obtain information about save sets and the files in a save set. You can display this information at your terminal or send it to an output file.

Because BACKUP writes save sets in a format that only BACKUP can interpret, a list operation is the only way to determine the contents of a save set without restoring the save set. You can perform a list operation in conjunction with any other BACKUP operation.

By default, a save-set listing supplies information about files in the save set similar to the information supplied by the DCL command DIRECTORY/DATE /SIZE, including the actual number of blocks used for each file.

You can also perform a BACKUP list operation to list the contents of a BACKUP journal file. BACKUP journal files, which are created during a save operation by using the command qualifier /JOURNAL[=*file-spec*], contain on-disk records of BACKUP save operations and the file specifications of the files saved during each operation. Section 10.13.4 contains more information about creating and listing BACKUP journal files.

How to Perform This Task

To list the contents of a BACKUP save set, do the following:

1. Insert the media containing the save set into the drive.
2. If the volume is a disk, mount the disk as described in Section 10.8.2 (BACKUP mounts tapes automatically).

3. Enter the BACKUP/LIST command in the format specified in the *OpenVMS System Management Utilities Reference Manual*. The /REWIND qualifier rewinds the tape to the beginning before searching for the save set. To list all the save sets on a volume, include the asterisk wildcard character (*) with the device specification.

   To list the contents of save sets does not require you to know the names of save sets on magnetic tape. Enter the device specification of the drive in which the tape is inserted with the BACKUP/LIST command. BACKUP reads the next save set it encounters on the magnetic tape and stops processing when it reaches the end of that save set. BACKUP does not automatically rewind to the beginning-of-tape marker unless you include the /REWIND qualifier in your command. Therefore, you can list the next save set (if one exists) by repeating the BACKUP/LIST command. If no more save sets exist on the tape, BACKUP issues the following error messages:

   ```
   %BACKUP-F-OPENIN, error opening MUA0:[000000].; as input
   -SYSTEM-W-NOSUCHFILE, no such file
   ```

Examples

1. To obtain save-set information about a magnetic tape save set named 2MAR1555.BCK in the drive MIA0:, enter the following command:

```
$ BACKUP/LIST MIA0:2MAR1555.BCK/REWIND
Listing of save set(s)

Save set:          2MAR1555.BCK
Written by:        POLYANNA
UIC:               [000200,000207]
Date:              21-MAY-1995 09:36:14.68
Command:           BACKUP/LOG [USER.SAVE] MIA0:2MAR555.BCK/REWIND/LABEL=WKY201

Operating system:  VMS version 7.0

BACKUP version:    7.0
CPU ID register:   08000000
Node name:         _SUZI::
Written on:        _MIA0:
Block size:        8192
Group size:        10
Buffer count:      3

[USER.SAVE]ANOTHER.DAT;1                1  18-MAY-1995 14:10
[USER.SAVE]LAST.DAT;1                   1  18-MAY-1995 14:11
[USER.SAVE]THAT.DAT;1                   7  18-MAY-1995 14:10
[USER.SAVE]THIS.DAT;2                   1  18-MAY-1995 13:44

Total of 4 files, 10 blocks
End of save set
```

2. The following command rewinds the tape to the beginning and lists all save sets on the volume MIA0:

   ```
   $ BACKUP/LIST MIA0:*.*/REWIND
   ```

3. The following command combines a list operation with a save operation to magnetic tape:

   ```
   $ BACKUP/LIST=MYBACK.DAT [PRAMS] MTA0:2MAR1555.BCK/LABEL=DLY201
   ```

   BACKUP verifies that the volume label is DLY201 and copies the contents of the directory [PRAMS] to a save set named 2MAR1555.BCK. The command qualifier LIST causes BACKUP to write save-set information to the file MYBACK.DAT as the save operation proceeds.

10.11 Understanding Multivolume BACKUP Operations

When you save data with BACKUP, the save set often spans more than one volume, creating a multivolume save set. When this occurs, BACKUP fits as much data as it can on the first volume, then dismounts it. Depending on whether you specified more than one drive in the BACKUP command line or if you are using a tape loader, BACKUP then does the following:

- If you specified only one drive in the BACKUP command line and you are not using a tape loader or operator assistance, BACKUP prompts you to remove the tape that is in the drive and insert another one:

  ```
  %BACKUP-I-RESUME, resuming operation on volume 2
  %BACKUP-I-READYWRITE, mount volume DAILY02 on MUA0: for writing
  Respond with YES when ready:
  ```

  Note

  If you are using OPCOM and the /ASSIST qualifier (the default), the following message appears on your terminal:

  ```
  %BACKUP-I-RESUME, resuming operation on volume 2
  %MOUNT-I-OPRQST, Please mount volume DAILY02 in device MUA0:
  BACKUP requests: Saveset DAILY.SAV, Volume number 02, write ENABLED
  ```

  After you insert and load the second volume (or an operator fulfills the mount request), BACKUP continues writing data to the second volume.

- If you specified multiple drives on the command line, BACKUP continues writing data to the second volume, assuming the drive is loaded, is on line, and has the correct volume label. BACKUP unloads the first volume and displays the following message:

  ```
  %BACKUP-I-RESUME, resuming operation on volume 2
  ```

- If you are using a tape loader, BACKUP continues writing data to the tape in the next slot, assuming the tape loader has an adequate supply of correctly labeled tapes. BACKUP rewinds and unloads the first tape and displays the following message:

  ```
  %BACKUP-I-RESUME, resuming operation on volume 2
      .
      .
      .
  ```

10.11.1 Multivolume Tape Labeling

In a multivolume save-set operation, BACKUP does not initialize the first volume (unless you use the /REWIND qualifier). BACKUP *does* initialize subsequent volumes. BACKUP determines the volume labels for subsequent volumes as follows:

- If you did not specify a label on the command line, BACKUP uses the first six characters of the save-set name to create a label for the first volume (unless you use the /EXACT_ORDER qualifier, in which case BACKUP preserves the volume label on the tape). For subsequent volumes, BACKUP uses the first four characters from the label of the first volume plus the number of the volume in the sequence. For example, suppose you are saving files that require three tapes and the save-set name is BACKUP. If you do not specify

a label, the first tape is labeled BACKUP, the second BACK02, and the third BACK03.

- If you specified a single label on the command line using the /LABEL qualifier and it matches the label of the first volume, BACKUP labels subsequent volumes with the first four characters of the label from the first volume plus the number of the volume in the sequence. For example, suppose you are saving files that require three tapes and the first tape is labeled TAPE. The second tape gets the label TAPE02, and the third tape gets the label TAPE03.
- If you specified multiple labels on the command line using the /LABEL qualifier (without the /EXACT_ORDER qualifier), BACKUP uses the labels you specify. If the operation requires more labels than you specified, BACKUP uses the first four characters of the last volume label and the volume number of the tape.
- You can use the /EXACT_ORDER qualifier in conjunction with the /LABEL qualifier to specify the order in which you want BACKUP to use the labels. BACKUP continues the operation as long as the label of the tape in the drive matches the corresponding label on the command line. If you do not specify enough labels on the command line to complete the operation, BACKUP prompts you to enter a label for the tape in the drive.

As a safeguard against initializing or writing the wrong tape, BACKUP compares the label that you specify on the command line to the label of the tape in the drive. Section 10.12 describes how BACKUP processes tape labels and handles a label mismatch.

10.12 Understanding BACKUP Tape Label Processing

After mounting a tape, BACKUP processes information stored in the volume header record of the tape before writing to it. Specifically, BACKUP does the following:

- Checks the volume protection information to ensure that you have the right to access the volume in the manner you requested.
- Checks the tape expiration date to prevent you from initializing a magnetic tape that has not yet expired.
- Compares the volume label specified in the BACKUP command line (either explicitly with the /LABEL qualifier or implicitly through the save-set name) to the volume label of the tape to prevent you from creating a save set on the wrong magnetic tape. BACKUP uses the following guidelines when processing tape labels:
  - If you specify a label that is longer than six characters, BACKUP truncates the label to six characters.
  - If the volume label is less than six characters long, BACKUP pads the volume label with the blank character to six characters.
  - The first four characters of the volume label either must match the first four characters of the label specified in the BACKUP command line exactly, or must end with one or more underscore characters. If the first four characters of the volume label end with one or more underscore characters, and the label specified in the command line matches the part of the volume label that appears before the underscore characters, BACKUP accepts the match. (For example, the volume label ABN_

matches the command line label ABN but does not match the command line label ABNE.)

- If the fifth and sixth characters of the volume label are numbers between 0 and 9, BACKUP does not compare these characters with corresponding characters in the label specified in the BACKUP command line. Otherwise, the fifth and sixth characters in the volume label must exactly match the corresponding characters in the label specified in the BACKUP command line.

If the labels match, you have the proper protection, and the tape is expired, BACKUP performs the designated operation.

If you specify more than one label with the /LABEL qualifier and you do not specify the /EXACT_ORDER qualifier, the BACKUP operation succeeds if any of the labels you specify match the tape's volume label. For example, if the tape's volume label is MA1686, the BACKUP operation will succeed if you specify the following list of labels with the /LABEL qualifier:

```
/LABEL=(MA1684,MA1685,MA1686)
```

If the volume labels do not match, BACKUP displays the following error message:

```
%MOUNT-I-MOUNTED, DKA0 mounted on _SODAK$MUA0:
%BACKUP-W-MOUNTERR, volume 1 on _SODAK$MUA0 was not mounted because
 its label does not match the one requested
%BACKUP-W-EXLABEER, volume label processing failed because
 volume MB1684 is out of order, Volume label MA1684 was expected
 specify option (QUIT, NEW tape, OVERWRITE tape, USE loaded tape)
BACKUP>
```

Depending on the option you specify, you can quit the backup operation (QUIT), dismount the old tape and mount a new one (NEW), overwrite the data on the tape (OVERWRITE), or USE the loaded tape.

If you specify more than one label with the /LABEL qualifier and you also specify the /EXACT_ORDER qualifier, BACKUP compares the label of the loaded tape with the first label that you specified with the /LABEL qualifier. If the labels match, BACKUP begins the operation. If the labels do not match, BACKUP prompts you with the previous message.

Assuming the volume labels of the tapes you use match the corresponding labels on the command line, BACKUP continues processing until it completes the operation or runs out of volume labels. If you do not specify enough labels on the command line to complete the operation or if the tape loaded does not have an ANSI label, BACKUP prompts you to enter a label for the tape in the drive.

If you use blank tapes or tapes that you intend to overwrite, use the /IGNORE=LABEL_PROCESSING qualifier. This suppresses the previous BACKUP message, which normally occurs if BACKUP encounters a non-ANSI-labeled tape during a save operation.

For more information about the /EXACT_ORDER, /IGNORE, and /LABEL qualifiers, see the *OpenVMS System Management Utilities Reference Manual*.

10.13 Backing Up Files and Directories

This section explains copying files, backing up files and directories, comparing files, and creating and listing BACKUP journal files.

10.13.1 Copying Files to Other Files

You can copy files using BACKUP. The copy function of the BACKUP command differs from the DCL command COPY because it preserves certain file information such as the version number, creation dates, revision dates, and protection codes (although, by default, the owner UIC of the copies is the UIC of the current process). Also, unlike the DCL command COPY, you can use BACKUP to copy entire directory trees, maintaining the directory structure.

How to Perform This Task

To make identical disk-to-disk copies of files, use the following format:

BACKUP input_specifier output_specifier

Examples

1. The following command copies the file EMPLOYEES.DAT in the current directory to the directory [BATES.TEST]:

   ```
   $ BACKUP EMPLOYEES.DAT USER1:[BATES.TEST]EMPLOYEES.DAT
   ```

2. You can also create copies of entire directory trees. For example:

   ```
   $ BACKUP USER1:[BATES...] USER2:[BATES...]
   ```

   This command re-creates the directory structure of user BATES on the disk named USER2:

3. The following command copies all files in the directory tree [LYKINS...] to the directory tree [OWLCR...] on the same disk:

   ```
   $ BACKUP [LYKINS...]*.*;* [OWLCR...]*.*;*
   ```

10.13.2 Backing Up Files and Directories to a Save Set

One of the most common BACKUP operations is to save files to a save set. There are several types of save sets. For more information on save sets, see Section 10.5.

How to Perform This Task

To back up files or directories, use the BACKUP command in the following format:

BACKUP input_specifier output_specifier [/SAVE_SET] [/LABEL=label]

The *input_specifier* specifies the file you want to back up, and the *output_specifier* specifies the device and save-set name.

When you save data to disk, use the output save-set qualifier /SAVE_SET. If you do not specify /SAVE_SET, BACKUP copies files in standard file format rather than creating a BACKUP save set. When you save data to tape, you do not need to specify /SAVE_SET; BACKUP treats all magnetic tape files as save sets. Use the /LABEL qualifier to specify the label of the tape you are using.

Examples

1. The following commands back up the file EMPLOYEES.DAT to a save set:

   ```
   $ ALLOCATE MUA0: TAPE1 ❶
   %DCL-I-ALLOC, MUA0: allocated
   $ INITIALIZE TAPE1 DLY101 ❷
   $ BACKUP/LOG EMPLOYEES.DAT MUA0:EMPL_MAY91.BCK/LABEL=DLY101 ❸
   %MOUNT-I-MOUNTED, BACKUP mounted on _MUA0:
   BACKUP-S-COPIED, copied DUA0:[SCHULT]EMPLOYEES.DAT;32
   $
   ```

In this example, the individual commands do the following:

❶ Allocate the tape drive MUA0: and assign it the logical name TAPE1.

❷ Initialize the tape in the drive and assign it the label DLY101.

❸ Save the file EMPLOYEES.DAT to a save set on the tape in MUA0. The /LOG qualifier causes BACKUP to display the file specification of the file that BACKUP copies. The /LABEL qualifier indicates the volume label that you assigned with the INITIALIZE command.

2. To create a magnetic-tape save set named NOV13SAVE.BCK that contains all files and subdirectories of a directory tree named [LYKINS...], enter the following command:

```
$ BACKUP [LYKINS...] TAPE:NOV13SAVE.BCK/LABEL=NOV13
```

3. You can also specify a list of files that you want to back up:

```
$ BACKUP
_From: DUA0:[MGR]EMPLOYEES.DAT,USER1:[RECORDS]DOOHAN.DAT,EVANS.DAT
_To: MUA1:MONTHLY_AUG.BCK/LABEL=TAPE1
```

4. If you are backing up large amounts of data, you can also specify more than one output device:

```
$ BACKUP
_From: DUA0:[000000]*.*
_To: MTA1:BACKUP.BCK,MTA2:
```

In this example, if BACKUP uses all of the space on the tape in MTA1:, it continues writing the save set on the tape in MTA2: (assuming MTA2: contains a tape that has never been initialized or one that has been initialized with the label BACK02).

5. As shown in the following example, you can create a Files–11 save set that consists of a single file; DUA1: is already mounted:

```
$ BACKUP STRATCOL1.DAT DUA1:STRATDAT1.BCK/SAVE_SET
```

6. To create a network save set, add the node, user name, and password to the output specifier in the following format:

remote_nodename"username password"::device_name:[directory]

For example:

```
$ BACKUP
From: STRATCOL1.DAT
To: NIMBL"ROGERS SANFRANCISCO"::WORK1:[ROGERS]STRATDAT1.BCK/SAVE_SET
```

7. To create a sequential-disk save set on DUA0: named NOV12SAVE.BCK that consists of all files in the current default directory, enter the following commands:

```
$ MOUNT/FOREIGN DUA0:
$ BACKUP [] DUA0:NOV12SAVE.BCK/SAVE_SET
```

8. The following example backs up the directory tree [REPORTS...] to a save set:

```
$ BACKUP [REPORTS...] MIA11:REPORT.BCK/REWIND/IGNORE=LABEL_PROCESSING
```

The /REWIND qualifier in this command line rewinds the tape and initializes it. The /IGNORE=LABEL_PROCESSING qualifier causes BACKUP to ignore any existing label information on the tape. Because the command does not include the /LABEL qualifier, BACKUP uses the first six characters of the save-set name (REPORT) as the label.

9. You can also back up a directory to a disk that is mounted in the Files–11 format. For example:

```
$ MOUNT DUA1: PAYROLL
%MOUNT-I-MOUNTED, PAYROLL mounted on _DUA1:
$ MOUNT DUA21: DISK21
%MOUNT-I-MOUNTED, DISK21 mounted on _DUA21:
$ BACKUP
From: DUA1:[PAYROLL]
To: DUA21:[PAYROLL_BACKUPS]PAY22MAY1995.SAV/SAVE_SET
```

   If the contents of the [PAYROLL] directory exceed the capacity of the disk DUA21:, the backup operation fails.

10. If you are backing up more data than the output volume can contain, mount the output volume using the /FOREIGN qualifier and create a sequential disk save set. For example:

```
$ MOUNT DUA1: PAYROLL
%MOUNT-I-MOUNTED, PAYROLL mounted on _DUA1:
$ MOUNT/FOREIGN DJA21:
%MOUNT-I-MOUNTED, WEEKLY mounted on _DJA21:
$ BACKUP
From: DUA1:[PAYROLL]
To: DJA21:[PAYROLL_BACKUPS]PAY22MAY1995.SAV/SAVE_SET
```

    In this example, if the contents of the [PAYROLL] directory exceed the capacity of the disk DJA21:, BACKUP prompts you to remove the volume in the drive and insert another one. For more information about Files–11 and sequential disk save sets, see Section 10.5.

10.13.3 Comparing Files

A BACKUP compare operation compares a save set with disk files or compares disk files with other disk files. Perform a compare operation to check the integrity of a file or volume after a copy, save, or restore operation. For example, you can use the compare operation to compare a save set with original files or to compare files or volumes copied using BACKUP with original files.

Note

Because BACKUP processes files by blocks, comparing files not produced by BACKUP is likely to cause mismatch errors in files that are apparently identical.

How to Perform This Task

The two ways to perform a compare operation are:

- You can perform a compare operation in conjunction with a save, restore, copy, or list operation by specifying the command qualifier /VERIFY. When you specify /VERIFY, BACKUP first performs the save, restore, copy, or list operation and then compares the output with the input. When you use /VERIFY in a copy or list operation, BACKUP displays no message when it begins the compare operation. When you use /VERIFY in a save or restore

operation, BACKUP displays the following message when it begins the compare operation:

```
%BACKUP-I-STARTVERIFY, starting verification pass
```

- You can perform a compare operation independently of other BACKUP operations by specifying the command qualifier /COMPARE. In addition, you can use the /COMPARE and /IMAGE qualifiers to instruct BACKUP to perform an image compare operation. This operation compares files on two different disks by using the file identifications (FIDs).

  An image compare operation may not work correctly when you create two disks with identical files by incrementally backing up and restoring the files from one disk to the other disk. This is because BACKUP does not ensure that the incrementally restored files have the same FIDs as the incrementally saved files. This is true regardless of whether the /OVERLAY, /NEW_VERSION, or /REPLACE qualifiers are used in the restore command.

Examples

1. The following example compares a save set on tape with files on disk. The command directs BACKUP to compare the contents of the save set 2MAR1555.BCK with the directory [LYKINS].

   ```
   $ BACKUP/COMPARE MTA0:2MAR1555.BCK [LYKINS]
   ```

2. The following example compares files on disk; note the inconsistency in block 16 between UPLIFT.EXE;4 and UPLIFT.EXE;3:

   ```
   $ BACKUP/COMPARE UPLIFT.EXE;3 UPLIFT.EXE;4
   %BACKUP-E-VERIFYERR, verification error for block 16 of
    WRKD$:[LYKINS]UPLIFT.EXE;4
   ```

3. To compare two entire Files–11 volumes, use an image compare operation, as follows:

   ```
   $ BACKUP/IMAGE/COMPARE DBA1: DBA2:
   ```

4. To compare a physical save set with a Files–11 volume, use a physical compare operation, as follows. All disks in a physical compare operation must be mounted as foreign volumes.

   ```
   $ MOUNT/FOREIGN DBA2:
   $ BACKUP/PHYSICAL/COMPARE MIA0:PHYSBACK.BCK DBA2:
   ```

5. The following example combines a compare operation with a copy operation:

   ```
   $ BACKUP/VERIFY/LOG FRED.DAT [FRIENDS]OLDFRED.DAT
   %BACKUP-S-CREATED, created DISK$:[FRIENDS]OLDFRED.DAT;3
   %BACKUP-S-COMPARED, compared DISK$:[FRIENDS]OLDFRED.DAT;3
   ```

10.13.4 Creating and Listing BACKUP Journal Files

If you want to keep a record of BACKUP operations, create a journal file. A BACKUP journal file contains records of BACKUP save operations and the file specifications of the files saved during each operation.

How to Perform This Task

To create a journal file, use the command qualifier /JOURNAL=[*file-spec*] in a BACKUP save operation.

To list the contents of a BACKUP journal file, enter a command in the following format:

BACKUP/LIST[=file-spec]/JOURNAL[=file-spec]

You cannot specify an input or output specifier with a BACKUP/LIST/JOURNAL command. If you omit the file specification from the command qualifier /LIST, BACKUP directs the output to your terminal; if you omit the file specification from the command qualifier /JOURNAL, the journal file receives the default BACKUP journal file name (SYS$DISK:[ ]BACKUP.BJL).

For more information about creating and listing BACKUP journal files, see the description of the /JOURNAL qualifier in the *OpenVMS System Management Utilities Reference Manual*.

Example

This example shows how to create a BACKUP journal file and list the contents of the BACKUP journal file:

```
$ BACKUP/JOURNAL/LOG/IMAGE  DRA2: MIA0:3OCT.FUL
%BACKUP-S-COPIED, copied DRA2:[COLLINS]ALPHA.DAT;4
%BACKUP-S-COPIED, copied DRA2:[COLLINS]EDTINI.EDT;5
.
.
.
%BACKUP-I-RESUME, resuming operation on volume 2
%BACKUP-I-READYWRITE, mount volume 2 on _MIA0: for writing
Press return when ready: Return
%BACKUP-S-COPIED, copied DRA2:[LANE]MAIL.MAI;1
%BACKUP-S-COPIED, copied DRA2:[LANE]MEMO.RNO;5
.
.
.
$ BACKUP/JOURNAL/LIST
Listing of BACKUP journal
Journal file _DB2:[SYSMGR]BACKUP.BJL;1 on 3-OCT-1995 00:40:56.36
Save set 3OCT.FUL created on 3-OCT-1995 00:40:56.36
Volume number 1, volume label 3OCT01

         [COLLINS]ALPHA.DAT;4
         [COLLINS]EDTINI.EDT;5
         [COLLINS]LOGIN.COM;46
         [COLLINS]LOGIN.COM;45
         [COLLINS]MAIL.MAI;1
         [COLLINS]MAR.DIR;1
         [COLLINS.MAR]GETJPI.EXE;9
         [COLLINS.MAR]GETJPI.LIS;14
                   .
                   .
         [LANE]LES.MAI;1
                   .
                   .
Save set 3OCT.FUL created on 3-OCT-1995 00:40:56.36
Volume number 2, volume label 3OCT02

          [LANE]MAIL.MAI;1
          [LANE]MEMO.RNO;5
          [LANE]MEMO.RNO;4
                   .
                   .
          [WALTERS.VI]KD.RNO;52

End of BACKUP journal
```

10.14 Restoring Files and Directories

A BACKUP restore operation takes a save set and restores it to its original condition. Often a restore operation is the result of a crisis (you have deleted an important file or a disk has become corrupted, for example). When you restore files, BACKUP places the contents of the save set in the location that you specify.

If you want to restore an entire disk, see Section 10.16.

How to Perform This Task

To restore files, use the BACKUP command in the following format:

BACKUP save_set_specifier [/SAVE_SET] /SELECT=[dir...] output_specifier:[dir...]

Use the /SAVE_SET qualifier if the save set is on a disk or diskette. The /SELECT qualifier lets you specify the exact file you want to restore.

If your save set is stored on more than one magnetic tape or sequential disk volume, it is possible to begin restore and compare operations with any volume of the save set. However, if you are restoring a save set with the command qualifier /IMAGE, processing must begin with the first volume. (An image restore operation restores all files to a volume or volume set.) If you attempt an image restore or compare operation and specify a tape that is not the first volume of the save set, you receive the following message:

```
%BACKUP-W-NOT1STVOL, tape 'name' is not the start of a save set
```

You can use the command qualifier /LOG to monitor the files as they are restored. If you want to restore only a small number of files from a large save set, press Ctrl/Y to terminate processing once the files you need have been restored.

Examples

1. If you mistakenly delete the file USER1:[WORK.SEPT]INVOICES.DAT but it has been backed up to a save set named NIGHTLY.BCK, you could restore it using the following command:

   ```
   $ BACKUP
   _From: MUA0:NIGHTLY.BCK/SELECT=[WORK.SEPT]INVOICES.DAT
   _To: USER1:[WORK.SEPT]INVOICES.DAT
   ```

2. You can also use wildcard characters to restore more than one file. For example:

   ```
   $ BACKUP/LOG
   _From: MUA0:NIGHTLY.BCK/SELECT=[WORK.SEPT]INVOICES*.*
   _To: USER1:[WORK.SEPT]INVOICES*.*
   %BACKUP-S-CREATED, created USER1:[WORK.SEPT]INVOICES_01.TXT;1
   %BACKUP-S-CREATED, created USER1:[WORK.SEPT]INVOICES_02.TXT;1
   %BACKUP-S-CREATED, created USER1:[WORK.SEPT]INVOICES_03.TXT;1
   %BACKUP-S-CREATED, created USER1:[WORK.SEPT]INVOICES_04.TXT;1
   %BACKUP-S-CREATED, created USER1:[WORK.SEPT]INVOICES_05.TXT;1
   %BACKUP-S-CREATED, created USER1:[WORK.SEPT]INVOICES_06.TXT;1
      .
      .
      .
   ```

   The /LOG qualifier displays the file specification of the files that you restored.

3. The following example restores files from the magnetic tape save set NOV12SAVE.BCK to subdirectories of the directory [LYKINS]:

   ```
   $ BACKUP TAPE:NOV12SAVE.BCK [LYKINS...]
   ```

4. To restore a specific file from a save set, use the input save-set qualifier /SELECT. In the following example, the file STRAT1.DAT in the directory [LYKINS.GLENDO] was deleted accidentally. The user, who previously saved the file to a save set named NOV2SAVE.BCK, uses BACKUP to restore the file to the directory. Next, the user enters the DIRECTORY command to confirm that the file has been restored to the subdirectory [LYKINS.GLENDO].

```
$ BACKUP
_From: MIA0:NOV2SAVE.BCK/SELECT=[LYKINS.GLENDO]STRAT1.DAT;5
_To: STRAT1.DAT;5
$ DIRECTORY STRAT1.DAT
Directory [LYKINS.GLENDO]

STRAT1.DAT;5

Total of 1 file.
$
```

5. Suppose you deleted the entire [REPORTS] directory, which previously contained the following subdirectories:

```
$ SET DEFAULT [REPORTS]
$ DIRECTORY *.DIR

Directory USER3:[REPORTS]

INTERNAL.DIR             2
PUBLIC.DIR               5
SUMMARIES.DIR            1
TEST.DIR                 3
WEEKLY.DIR               2

Total of 5 files, 13 blocks.
$
```

   If you made a backup save set of the directory and subdirectories, you could restore them. For example:

```
$ BACKUP MUA0:MAY-10.BCK/SELECT=[REPORTS...] USER3:[REPORTS...]
```

   This command restores all the files in the [REPORTS] directory and the subdirectories ([.INTERNAL], [.PUBLIC], [.SUMMARIES], [.TEST], and [.WEEKLY]).

6. To restore all files from a magnetic-tape save set named NOV12SAVE.BCK to the directory tree from which they were saved, enter the following command:

```
$ BACKUP TAPE:NOV12SAVE.BCK/REWIND [*...]
```

   The /REWIND qualifier directs BACKUP to rewind the tape to the beginning-of-tape before beginning the restore operation. This ensures that the save set will be restored even if it is located before the current tape position.

10.14.1 Accessing Files in Deep Directory Structures

BACKUP can access a file in a directory structure that is a maximum of 32 levels deep. BACKUP can also select a file from within a BACKUP save-set file that was previously in a deep directory (one that is greater than 8 levels deep). On an ODS-2 disk, however, you can restore a file from a directory that is a maximum of 8 levels deep. The following example restores a deep directory structure that is 12 levels deep:

```
$ BACKUP MTA1:T.BCK/SAV/SELECT=[A.B.C.D.E.F.G.H.I.J.K.L]*.* DISK:[DIR]*.*;*
```

10.15 Backing Up User Disks

This section explains performing incremental and image backups to disk and tape.

10.15.1 Preparing to Back Up User Disks

Digital recommends that you back up your disks with no interactive users logged in and with no applications running. This is because if BACKUP encounters an open file during a save operation, it issues an error message and *does not* copy the file. Also, because of the way BACKUP scans directories, *any* activity in a directory (such as creating or deleting files) can cause files to be excluded from the backup.

You can instruct BACKUP to save open files by using the /IGNORE=INTERLOCK qualifier on the BACKUP command, as described in Section 10.18.3. However, open files saved by BACKUP might contain inconsistent data, depending on the applications that are writing to the open files. BACKUP reports a message if either:

- The file was modified while BACKUP was reading the file
- The file is accessed for writing on the local node when BACKUP finishes reading the file

However, if the file is accessed for writing from a remote node when BACKUP finishes reading the file, no message is displayed because BACKUP cannot detect the access.

If a file with the specified version already exists, BACKUP reports the following error message:

```
RMS-E-FEX, file already exists, not superseded
```

How to Perform This Task

If several users are on your system, notify them that a disk backup is about to take place. If you have the OPER privilege, you can notify users with the REPLY /ALL command, as follows:

```
$ REPLY/ALL "System Backup About to Begin -- Open Files Will Not Be Backed Up"
```

When you enter this command, each interactive terminal on the system displays the following message:

```
Reply received on MYNODE from user SYSTEM at VTA28:   23:35:11
System Backup About to Begin -- Open Files Will Not Be Backed Up
```

Guidelines for Backup

If this is the first time a disk is being backed up, perform an image backup using the BACKUP/IMAGE/RECORD command before performing regular incremental backups. The image backup saves a copy of the entire disk and marks each file as having been saved. Regularly performed subsequent incremental backups assume an image backup was already performed and therefore will save new or modified files. If an image backup was not performed first, the incremental backups will save more files than may be necessary, ensuring that an incremental restore will be successful.

10.15.2 Performing Image Backups to Tape

As described in Section 10.2, an image backup of a disk provides you with an exact logical copy of all the files on the disk. You should perform image backups with no interactive users on the system because of open file considerations (described in Section 10.15.1). Also, system performance can be affected during the backup process, so it is best to schedule the backup during the least busy times for your system. You can optimize the speed of the backup procedure by ensuring that certain process and system parameters are set properly (as described in Section 10.7).

How to Perform This Task

To perform an image backup, use the BACKUP command in the following format:

BACKUP/IMAGE [/RECORD] input_device output_specifier [/LABEL=label] [/REWIND]

The /IMAGE qualifier identifies the backup operation as an image backup. The /RECORD qualifier is optional and records the current date and time in the file header record of each file that is backed up. You must use the /RECORD qualifier if you are planning to perform future incremental backups. Specify the name of the disk you are backing up as the *input_device*; do not specify individual files. The /REWIND qualifier is optional depending on whether you want to initialize the tape. The /LABEL qualifier identifies the label of the tape.

Examples

1. The following example shows how to create an image backup of a disk on your workstation. If the disk is named DKA100:, and the tape cartridge drive is named MKB100:, you could perform the image backup by entering the following commands:

```
$ INITIALIZE MKB100: WKLY ❶
$ MOUNT DKA100: DISK$1 ❷
%MOUNT-I-MOUNTED, DISK$1 mounted on _DKA100:
$ BACKUP/IMAGE/RECORD/VERIFY
_From: DKA100:
_To: MKB100:FULL02.SAV/LABEL=WKLY ❸
%BACKUP-I-STARTVERIFY, starting verification pass
```

   In this example, the individual commands do the following:

   ❶ Initialize the tape in MKB100: with the label WKLY.

   ❷ Mount the disk DKA100: (BACKUP will mount the tape drive).

   ❸ Back up the disk DKA100: to the save set FULL02.SAV on MKB100. The /IMAGE qualifier indicates that this is an image backup. The /RECORD qualifier records the current date and time of the backup in the file header record of each file that is backed up. The /VERIFY qualifier causes BACKUP to check the contents of the output specifier against the input specifier after the files are written to the volume. The /LABEL qualifier indicates the label of the tape.

2. If you are backing up a large disk, you may want to use several tape drives for the backup. For example:

```
$ ALLOCATE MUA0:,MUA1:,MUA2: ❶
%DCL-I-ALLOC, MUA0: allocated
%DCL-I-ALLOC, MUA1: allocated
%DCL-I-ALLOC, MUA2: allocated
$ BACKUP/IMAGE/RECORD/NOASSIST/RELEASE_TAPE
_From: DKA100:
_To: MUA0:FULL02.SAV,MUA1,MUA2/LABEL=MNTH ❷
%MOUNT-I-MOUNTED, MNTH mounted on _MUA0:
%BACKUP-I-RESUME, resuming operation on volume 2
%MOUNT-I-MOUNTED, MNTH02 mounted on _MUA1:
%BACKUP-I-RESUME, resuming operation on volume 3
%MOUNT-I-MOUNTED, MNTH03 mounted on _MUA2:
$
```

In this example, the individual commands do the following:

❶ Allocate the tape drives that will be used in the backup.

❷ Back up DKA100: to a save set. The /IMAGE qualifier indicates this operation is an image backup. BACKUP begins writing data to a save set on the tape in MUA0. If the tape in MUA0: becomes full, BACKUP initializes the tape in MUA1: and continues writing the save set. The tape in MUA1: gets the label MNTH02. If necessary, BACKUP also uses the tape in MUA2.

The /RELEASE_TAPE qualifier dismounts and unloads an output tape device after BACKUP writes the save set. The /RECORD qualifier records the current date and time in the file header record of each file that is backed up.

10.15.3 Performing Image Backups to Disk

As described in Section 10.2, an image backup of a disk provides you with an exact logical copy of all the files on the disk. You should perform image backups with no interactive users on the system because of open file considerations (described in Section 10.15.1). Also, system performance can be affected during the backup process, so it is best to schedule the backup during the least busy times for your system. You can optimize the speed of the backup procedure by ensuring that certain process and system parameters are set properly (as described in Section 10.7).

How to Perform This Task

To perform an image backup to a disk, use the BACKUP command in the following format:

BACKUP/IMAGE/RECORD input_device output_specifier/SAVE_SET

The /IMAGE qualifier identifies the backup operation as an image backup. The /RECORD qualifier records the current date and time in the file header record of each file that is backed up. This information is essential for future incremental backups. The /SAVE_SET qualifier indicates that you are creating a save set on a disk.

Examples

1. For example, if you want to create an image backup save set of the disk named DUA1: on a disk named DUA2:, you could enter the following commands:

```
$ MOUNT DUA1: USER1
%MOUNT-I-MOUNTED, USER1 mounted on _DUA1:
$ MOUNT DUA2: USER2
%MOUNT-I-MOUNTED, USER2 mounted on _DUA2:
$ BACKUP/IMAGE/RECORD
_From: DUA1:
_To: DUA2:[USER.BACKUPS]USER1.SAV/SAVE_SET
```

2. You can also specify multiple disk drives as the output specifier in the BACKUP command line. For example:

```
$ BACKUP/IMAGE/RECORD
_From: DUA0:
_To: DUB24:[USER.BACKUPS]USER1.SAV,DUB25/SAVE_SET
```

10.15.4 Performing Incremental Backups to Tape

As described in Section 10.2, an incremental backup of a disk provides you with an exact copy of only those files that have been created or modified since the last image or incremental backup in which the /RECORD qualifier was used.

How to Perform This Task

To perform an incremental backup to tape, do the following:

1. Perform an image backup using the /RECORD qualifier (see Section 10.15.2).
2. To determine the date of the last backup that used the /RECORD qualifier, enter the DIRECTORY/FULL command and the file name. For example:

```
$ DIRECTORY/FULL LOGIN.COM

Directory WORK204:[HIGGINS]

LOGIN.COM;31                File ID:  (23788,1,0)
Size:            7/9        Owner:    [ACC,HIGGINS]
Created:  30-APR-1995 14:37:33.98
Revised:  30-APR-1995 14:37:34.44 (1)
Expires:   <None specified>
Backup:   30-APR-1995 20:20:57.37
File organization:  Sequential
File attributes:    Allocation: 9, Extend: 0, Global buffer count: 0, No version limit
Record format:      Variable length, maximum 94 bytes
Record attributes:  Carriage return carriage control
RMS attributes:     None
Journaling enabled: None
File protection:    System:RWED, Owner:RWED, Group:RE, World:
Access Cntrl List:  None

Total of 1 file, 7/9 blocks.

$
```

The date of the last /RECORD backup is indicated in the Backup field of the display. In this example, a /RECORD backup was performed on 30-APR-1995 20:20:57.37.

Note

If you used the /IGNORE=INTERLOCK qualifier to back up open files during your last image backup or incremental backup in which the /RECORD qualifier was used, see Section 10.18.3. If the files remain open, they will not be included in the incremental backup because their backup date fields are not as recent as the last image backup or incremental backup in which the /RECORD qualifier was used.

3. Enter the BACKUP command in the following format:

   BACKUP/RECORD/SINCE=BACKUP input_specifier output_specifier[/LABEL=label] [/REWIND]

   The /RECORD qualifier records the current date and time in the file header record of each file that is backed up. This information is essential for future incremental backups. The /SINCE=BACKUP qualifier backs up files dated later than the last /RECORD backup. The /REWIND qualifier is optional depending on whether you want to initialize the tape. The /LABEL qualifier identifies the label of the tape.

Examples

The following command is an example of an incremental backup in which BACKUP saves all files on DRA1: that were modified since the previous BACKUP /RECORD command and stores them in a save set named 20APR1995.SAV:

```
$ BACKUP/RECORD/SINCE=BACKUP/RELEASE_TAPE
From: DRA1:[000000...]
To: MIA0:20APR1995.SAV/LABEL=20JUNE
```

The /LABEL qualifier identifies the volume label of the tape. Also, because BACKUP is performing an incremental rather than an image backup, it is necessary to explicitly use the notation DRA1:[000000 . . .] to specify all the files on DRA1. The /SINCE=BACKUP qualifier saves all files created or modified since the last /RECORD backup. The /RELEASE_TAPE qualifier dismounts and unloads an output tape device after BACKUP writes the save set and before it performs the action of the /RECORD command.

10.15.5 Performing Incremental Backups to Disk

As described in Section 10.2, an incremental backup of a disk provides you with an exact copy of only those files that have been created or modified since the last image or incremental backup in which the /RECORD qualifier was used.

How to Perform This Task

To perform an incremental backup to disk, do the following:

1. To perform an incremental backup, you must first perform an image backup using the /RECORD qualifier (see Section 10.15.2).
2. To determine the date of the last backup that used the /RECORD qualifier, enter the DIRECTORY/FULL command and the file name. For example:

```
$ DIRECTORY/FULL LOGIN.COM

Directory WORK204:[HIGGINS]

LOGIN.COM;31                File ID:  (23788,1,0)
Size:            7/9        Owner:    [ACC,HIGGINS]
Created:  30-APR-1995 14:37:33.98
Revised:  30-APR-1995 14:37:34.44 (1)
Expires:   <None specified>
Backup:   30-APR-1995 20:20:57.37
File organization:  Sequential
File attributes:    Allocation: 9, Extend: 0, Global buffer count: 0, No version limit
Record format:      Variable length, maximum 94 bytes
Record attributes:  Carriage return carriage control
RMS attributes:     None
Journaling enabled: None
File protection:    System:RWED, Owner:RWED, Group:RE, World:
Access Cntrl List:  None

Total of 1 file, 7/9 blocks.
```

```
$
```

The date of the last /RECORD backup is indicated in the Backup field of the display. In this example, a /RECORD backup was performed on 30-APR-1995 20:20:57.37.

Note

If you used the /IGNORE=INTERLOCK qualifier to back up open files during your last image backup or incremental backup in which the /RECORD qualifier was used, see Section 10.18.3. If the files remain open, they will not be included in the incremental backup because their backup date fields are not as recent as the last image backup or incremental backup in which the /RECORD qualifier was used.

3. Enter the BACKUP command in the following format:

   BACKUP/RECORD/SINCE=BACKUP input_specifier output_specifier/SAVE_SET

   The /RECORD qualifier records the current date and time in the file header record of each file that is backed up. The first step in an incremental backup is an image backup (see Section 10.15.2). If you plan to perform incremental backups, you must use the /RECORD qualifier when you perform image backups. The /SINCE=BACKUP qualifier backs up files dated later than the last /RECORD backup. The /SAVE_SET qualifier indicates that you are creating a save set on a disk.

Examples

1. If you want to create an incremental backup of a disk named DUA55: on a sequential disk save set on a disk named DJC12:, you could enter the following commands:

```
$ MOUNT DUA55: DISK1
%MOUNT-I-MOUNTED, DISK1 mounted on _DUA55:
$ MOUNT/FOREIGN DJC12:
%MOUNT-I-MOUNTED, DISK2 mounted on _DJC12:
$ BACKUP/RECORD/SINCE=BACKUP
_From: DUA55:[000000...]
_To: DJC12:USER1.SAV/SAVE_SET
```

2. You can also specify multiple disk drives as the output device in the BACKUP command line. For example:

```
$ MOUNT DUA0: USER1
%MOUNT-I-MOUNTED, USER1 mounted on _DUA0:
$ MOUNT/FOREIGN DUB24:
%MOUNT-I-MOUNTED, DISK2 mounted on _DUB24:
$ MOUNT/FOREIGN DUB25:
%MOUNT-I-MOUNTED, DISK3 mounted on _DUB25:
$ BACKUP/RECORD/SINCE=BACKUP
_From: DUA0:[000000...]
_To: DUB24:USER1.SAV,DUB25/SAVE_SET
```

10.15.6 Backing Up Your Workstation Disk

On a standalone workstation, you are probably responsible for backing up files on your user disks. Section 10.15.6.1, Section 10.15.6.2, and Section 10.15.6.3 contain command procedures for making image, incremental, and interactive backups of user disks on your workstation.

Digital also provides two template command procedures in the SYS$EXAMPLES directory for you to use in designing BACKUP command procedures. These command procedures are called BACKUSER.COM and RESTUSER.COM.

If you are not familiar with using command procedures, see the *OpenVMS User's Manual*.

10.15.6.1 Using a Command Procedure for Nightly Image Backups

The following command procedure performs nightly image backups, backing up all the files on disk DUA2: to a tape in MUA0. The files are copied to a magnetic tape save set named FULL_BACKUP.SAV. This procedure is particularly useful for backing up files on a MicroVAX system or workstation.

How to Perform This Task

To use the command procedure, do the following:

1. Ensure that you have a batch queue available on your system. (See Section 13.7 for information about setting up a batch queues.) You submit the command procedure only once, and it will execute daily at 2:00 A.M. The command procedure automatically resubmits itself at 2:00 each morning; however, you must physically load a tape each day or the backup procedure will fail. Even if the backup procedure fails, however, the command procedure will continue to resubmit itself.
2. From the SYS$MANAGER directory, create the command procedure as shown and call it SYSTEM_BACKUP.COM.

```
$!
$! Resubmit this procedure --
$ SUBMIT/AFTER="TOMORROW+2:0" SYS$MANAGER:SYSTEM_BACKUP
$!
$  ON ERROR THEN GOTO FAILURE
$  SET PROCESS/PRIVILEGES=ALL
$!
$  REPLY/ALL -
    "Full Backup About to Begin.  Open Files Will Not Be Saved"
$!
$  BACKUP /IMAGE   DUA2:   MUA0:FULL_BACKUP.SAV /REWIND /IGNORE=LABEL_PROCESSING
$  DISMOUNT MUA0:
$  EXIT
$!
$FAILURE:
$  WRITE SYS$OUTPUT "---> Backup failed"
$  WRITE SYS$OUTPUT ""
$  DISMOUNT MUA0:
$  EXIT
```

3. Edit the command procedure to reflect:
   - The name of the disk that you want to back up. If you want to back up more than one disk, list each of the devices in the BACKUP command line. For example, you could substitute the following lines for the BACKUP command line in the preceding example:

```
    .
    .
    .
$!
$ BACKUP/IMAGE WORK_DISK MIA0:WORK_BACK.SAV/REWIND
$ BACKUP/IMAGE PAYROLL_DISK MIA0:PAYROLL_BACK.SAV
$!
    .
    .
    .
```

If you plan to perform any incremental backups later, include the /RECORD qualifier in the BACKUP command line.

- The name of the tape drive that you will use.
- The name that you want to assign to the save set.

4. Write down the name of the save set that you assigned.
5. Submit the command procedure using the following command line (if you gave your procedure a file name other than SYS$MANAGER:SYSTEM_BACKUP.COM, substitute the appropriate file name):

```
SUBMIT/NOPRINT/AFTER="TOMORROW+2:0"/QUEUE=queue_name SYS$MANAGER:SYSTEM_BACKUP
```

6. Be sure to change the tape daily and make sure that a tape is physically loaded on the device that you specified. When the backup is complete, keep the backup tape in a safe place and do not use the tape again until after you make another image backup of your disks.

To stop the procedure after you have submitted it, use the DELETE/ENTRY command. To find the entry number, use the SHOW ENTRY command. For example:

```
$ SHOW ENTRY
  Entry  Jobname         Username     Blocks  Status
  -----  -------         --------     ------  ------
     14  SYS_BACKUP   TPROULX                Holding until 19-APR-1995 02:00
         On generic batch queue CLUSTER_BATCH
$ DELETE/entry=583
```

10.15.6.2 Using a Command Procedure for Nightly Incremental Backups

You can use a similar command procedure to perform nightly incremental backups of your disks. It might be more convenient to perform nightly incremental backups and weekly image backups if either of the following conditions applies:

- Interactive users are on your system at all times, and system performance is noticeably affected by backups.
- A full backup does not fit on a single magnetic tape, but an incremental backup does. In this case, an image backup requires the presence of an operator (to change the tape), whereas an incremental backup can be run as a batch job.

Suppose that you want to do nightly incremental backups at 11:00 P.M., except on Friday night, when you want to do an image backup. The following command procedure executes an incremental backup on three disks and automatically resubmits itself to run again the following night, except for Friday night.

How to Perform This Task

To use the procedure, follow these steps:

1. From the SYS$MANAGER directory, create the command procedure as shown and call it INCREMENTAL_BACKUP.COM.

```
$!
$! Resubmit this procedure --
$ SUBMIT/AFTER="TOMORROW+23:0" SYS$MANAGER:INCREMENTAL_BACKUP
$!
$ TODAY = f$cvtime("today",,"weekday")
$ IF TODAY .EQS. "Friday" THEN GOTO DONE
$!
$  ON ERROR THEN GOTO FAILURE
$  SET PROC/PRIV=(OPER,BYPASS)
$!
$  REPLY/ALL -
    "Incremental Backup About to Begin.  Open Files Will Not Be Saved"
$!
$  BACKUP/RECORD/SINCE=BACKUP  DRA0:[000000...]  -
   MIA0:INCREMENT1.SAV /LABEL=INC1
$  BACKUP/RECORD/SINCE=BACKUP  DRA1:[000000...]  -
   MIA1:INCREMENT2.SAV /LABEL=INC2
$  BACKUP/RECORD/SINCE=BACKUP  DRA2:[000000...]  -
   MIA2:INCREMENT3.SAV /LABEL=INC3
$  DISMOUNT MIA0:
$  DISMOUNT MIA1:
$  DISMOUNT MIA2:
$  EXIT
$!
$FAILURE:
$  WRITE SYS$OUTPUT "---> Backup failed"
$  WRITE SYS$OUTPUT ""
$  DISMOUNT MIA0:
$  DISMOUNT MIA1:
$  DISMOUNT MIA2:
$  EXIT
```

2. Edit the procedure to reflect:
   - The names of the disks that you want to back up
   - The name of the tape drive that you will use
   - The volume label of the tape
   - The name that you want to assign to the save set
   - The day of the week (if any) to be omitted in the incremental backup

   In this example, the incremental backup will not be performed on Friday, reserving that day for an image (full) backup.

3. Be sure that an image backup has been made and also be sure that you continue to make regular image backups. When you make your image backups, be sure to use the /RECORD qualifier (as well as the /IMAGE qualifier) in your BACKUP command line.

4. Submit the command procedure using the following command line (if you gave your procedure a file name other than SYS$MANAGER:INCREMENTAL_BACKUP.COM, substitute the appropriate file name):

```
$ SUBMIT/AFTER=23  SYS$MANAGER:INCREMENTAL_BACKUP
```

5. Be sure that a tape is physically loaded on the device that you specified. When the incremental backup is complete, keep the tape in a safe place and do not use the tape again until you make another image backup.

10.15.6.3 Using an Interactive Command Procedure for Backups

You can use the following command procedure to interactively back up a disk to a magnetic tape.

How to Perform This Task

To use the procedure, do the following:

1. Create the command procedure in your directory:

```
$ ! Command procedure DAILYBACK.COM
$ !
$ ! Execute this command procedure interactively
$ !  by entering the command @[directory]DAILYBACK
$ !  at the DCL prompt.
$ !
$ ! The BACKUP command in this procedure contains the
$ !  output save-set qualifier /REWIND.  Therefore, this
$ !  command procedure always initializes the output tape.
$ !
$ ON ERROR THEN GOTO FAILURE
$ INQUIRE DRIVE "Enter the drive name (without a colon)"
$ ALLOCATE 'DRIVE'
$ INQUIRE SAVESET_SPEC "Enter the save-set specifier"
$ INQUIRE LBL "Enter the tape label"
$ INQUIRE EXP "Enter the tape expiration date"
$ BACKUP/NOASSIST/RECORD/IGNORE=INTERLOCK/SINCE=BACKUP -
  [...] 'DRIVE':'SAVESET_SPEC'/REWIND/LABEL='LBL'/TAPE_EXPIRATION='EXP'
$ DISMOUNT 'DRIVE'
$ EXIT
$!
$FAILURE:
$  WRITE SYS$OUTPUT "---> Backup failed"
$  WRITE SYS$OUTPUT ""
$  DISMOUNT 'DRIVE'
$  EXIT
```

2. Run the procedure and enter the drive, save set, tape label, and tape expiration information.

3. After the specified tape drive is allocated, BACKUP searches the tape's volume header record for a volume label and compares the label you specified with the /LABEL qualifier. If the volume header record contains no volume label, BACKUP writes the label and expiration date you specified to the volume header record and initializes the tape. Otherwise, BACKUP compares the tape's volume label with the label you specified and ensures that the tape is expired.

   If the tape is not expired or the label does not match, the command procedure exits. If the tape is expired and the label matches, BACKUP writes the expiration date you specified to the volume header record and initializes the tape. After initializing the tape, BACKUP saves all files in the current default directory tree that have been created or modified since the last save operation to a save set with the name you specified.

10.15.7 Backing Up Volume Shadow Sets

Volume shadowing maintains multiple copies of the same data on two or more disk volumes. If you use volume shadowing on your system, you can form a shadow set by uniting individual disk volumes (shadow set members). Volume shadowing duplicates data on each member of the shadow set. Per-disk licensing is available for each disk you will be including in a shadow set. This option is effective in a cluster where you intend to shadow only a small number of disks. However, if you have larger systems with many more disks to shadow, traditional capacity (per-CPU) licenses may be more appropriate.

Limits on the numbers of disks allowed in shadow sets are shown in Table 10–7:

Table 10–7 Number of Shadow Sets Supported

Type of Shadow Set	Sets Supported
Single member	Unlimited sets
Multimember	Total of 400 disks in two- and three-member sets, or both

These limits apply per cluster. For example, 400 total disks could be configured into 200 two-member shadow sets or into 133 three-member shadow sets per cluster. If single, two-, and three-member shadow sets are all present on a single cluster, then a maximum of 400 disks may be contained in the two- and three-member shadow sets.

You can use the firmware implementation of RAID level 1 (shadowing) to create shadow sets using the SCSI (Small Computer Systems Interface) disks attached locally to a single SWXCR-*xx* controller. The StorageWorks RAID Array 210 Subsystem (SWXCR-EA or SWXCR-EB EISA Backplane RAID controllers) and the StorageWorks PCI Backplane RAID controller (SWXCR-PA or SWXCR-PB) have their own firmware implementations of RAID, levels 0, 1, and 5.

SCSI disks connected to these controllers can also be included in shadow sets created using host-based volume shadowing for OpenVMS. For example, with host-based volume shadowing, you can create a RAID1 shadow set containing two like disks, each of which is attached to a separate SWXCR-*xx* RAID controller located within a cluster. SCSI disks can be configured as shadow sets when attached to systems running volume shadowing for OpenVMS.

For directly connected SCSI devices that have been powered down or do not answer to polling, the elapsed time before a device is removed from a shadow set approaches one minute. In all other situations, the elapsed time closely approximates the number of seconds specified in the SHADOW_MBR_TMO parameter.

Volume shadowing checks for geometries and maximum logical block numbers (LBN)s on devices. This enables devices such as the RZ28 and the RZ28B to operate in the same shadow set. Even though their device IDs differ, their geometries and maximum LBNs will match when configured on like controllers (two HSJ controllers, for example).

When you create a shadow set, individual users access it as a virtual unit. For example, you could create a virtual unit DSA1 that consists of the disks named DUA1:, DUA2:, and DUA3. Users cannot access the individual shadow set members directly, but can perform operations on the virtual unit (DSA1:).

Because of the way volume shadowing duplicates data on each disk in the shadow set, there are special considerations for backing up a shadow set. One strategy for backing up shadow sets involves using the OpenVMS Backup utility.

Caution

Do not attempt to back up a shadow set by dismounting an individual shadow set member or by backing up an active shadow set member. You must dismount the entire shadow set and re-create it less one shadow set member. If you do not follow this restriction, the resultant backup copy may contain inconsistent data.

How to Perform This Task

The proper procedure for using BACKUP to back up a shadow set is described in detail in the *Volume Shadowing for OpenVMS* manual, and can be summarized as follows.

Note

You cannot perform an incremental backup using this procedure because the backup record date is overwritten when you add the disk volume back into the existing shadow set.

1. Make sure that all shadow set members are *full* members; none of the members should be in a merge or copy state.
2. Dismount the *entire* shadow set.
3. Re-create the shadow set less one member. The data on the excluded member will mirror the data on the shadow set members.
4. Mount the former shadow set member for the backup.
5. Perform an image backup on the former shadow set member.
6. Dismount the former shadow set member when the backup is complete.
7. Add the shadow set member that you backed up.

10.15.7.1 Mounting a Disk in a Host-Based Shadow Set

To mount a disk in the StorageWorks RAID Array 110 Subsystem in a host-based shadow set, you must use the /OVERRIDE=NO_FORCED_ERROR qualifier with the MOUNT command.

The StorageWorks RAID Array 110 Subsystem does not support the READ/WRITE LONG SCSI commands that are necessary for implementing the FORCED ERROR function in SCSI. Without FORCED ERROR, you must override that check by the shadowing driver.

10.15.7.2 Assisted Merging in Mixed-Architecture Clusters

Assisted merging, also known as minimerge, is disabled if shadow sets are mounted on an OpenVMS Alpha node and also on other types of nodes in the same cluster. To reenable assisted merging, apply the CSCPAT (TIMA) kit to all VMScluster nodes mounting the shadow set.

With minimerge disabled, shadowing will continue to function normally. However, a full merge will always be done when a merge operation is required. A full merge takes considerably longer to complete than a minimerge operation; Digital recommends that you install the CSCPAT (TIMA) kit.

10.16 Restoring User Disks

Occasionally you may want to restore the backup copy of an entire disk. For example, if the disk drive fails, you could restore the backup copy to a working disk. By occasionally saving and restoring an image backup, you can also prevent disk fragmentation.

The way in which you restore a disk depends on whether the most recent backup was an image (full) or incremental backup. Section 10.16.1 describes the process for restoring a disk when the most recent backup was an image backup. Section 10.16.2 describes the process for restoring a disk when one or more incremental backups were performed since the most recent image backup.

10.16.1 Restoring Image Backups

This section describes how to restore the entire contents of a disk when your most recent backup was an image backup (using the /IMAGE qualifier, as described in Section 10.15.2).

How to Perform This Task

To restore an image backup, use the following procedure.

Caution

When you use the /IMAGE qualifier in a restore operation, the disk to which you are restoring the files is initialized. Initializing the disk removes links to the existing files, effectively erasing them. If you want to restore individual files or directories rather than the entire disk, see Section 10.14.

1. Mount the disk to which you will restore the files, using the MOUNT /FOREIGN command as described in Section 10.8.2.
2. Load and mount the volume. If the backup is contained in a Files–11 save set, make sure you mount the volume in the Files–11 format. If the backup is contained in a sequential disk save set, make sure you load the volume and mount it using the MOUNT/FOREIGN command. If the backup copy is on a tape save set, load the first tape.
3. If you do not know the name of the save set, do one of the following:
   - If the save set is on a disk, make sure the disk is mounted in the Files–11 format and use the DIRECTORY command to determine the name of the save set. For example:

     ```
     $ DIRECTORY BACKUP_DISK:[BACKUPS]

     Directory SYS$SYSDEVICE:[BACKUPS]

     19APRIL1995.SAV;1

     Total of 1 file.
     ```

     The save set is named 19APRIL1995.SAV.

- If the save set is on magnetic tape, load the tape and then enter the following command, substituting the name of your tape drive for *MIA1*:

```
$ BACKUP/LIST/REWIND MIA1:

Listing of save set(s)

Save set:          19APRIL1995.SAV
Written by:        SYSTEM
UIC:               [000001,000004]
Date:              19-APR-1995 22:03:03.63
    .
    .
    .
```

The save set is named 19APRIL1995.SAV.

4. To restore the save set, enter the BACKUP command with the /IMAGE qualifier, using the following syntax:

   BACKUP/IMAGE device:save_set_specifier [/SAVE_SET] output_device

   If your backup save set is on a disk or diskette, then you must also use the /SAVE_SET qualifier immediately after the save-set specifier (*device:save_set_specifier*).

5. If your backup save set is on more than one tape, disk, or diskette, BACKUP dismounts and unloads the current volume. Load the next volume when BACKUP prompts for it.

6. Use the /NOUNLOAD qualifier to dismount the disk onto which you just restored the files.

Example

The following example shows how to restore an image backup, using the following assumptions:

- The saved files are contained in a tape save set named FULL_BACKUP.SAV. This save set is the result of an image backup.
- The tape containing the saved copy of the disk contents is loaded on MIA1.
- DRA2: is the device name of the disk to which the files will be restored.

```
$ MOUNT/FOREIGN DRA2: ❶
%MOUNT-I-MOUNTED, DISK1 mounted on _DRA2:
$ BACKUP/IMAGE  MIA1:FULL_BACKUP.SAV/REWIND  DRA2: ❷
$ DISMOUNT/NOUNLOAD  DRA2: ❸
```

In this example, the individual command lines do the following:

❶ Mount the disk DRA2. The files will be restored to this disk. The disk must be loaded before it can be mounted.

❷ Initialize DRA2:, effectively erasing any previous data on the disk. Restore the directory structure and all the files from the save set FULL_BACKUP.SAV to the disk DRA2. BACKUP restores the files contiguously on DRA2:, eliminating any disk fragmentation on that device.

The /IMAGE qualifier restores a logical duplicate of the original disk so that the entire directory structure is restored and the files are placed in the proper directories.

❸ Dismount the disk.

10.16.2 Restoring Incremental Backups

Restoring files after making an image backup and one or more incremental backups is a two-step process. First, restore the most recent image backup. Then, restore each of the subsequent incremental backups, starting with the most recent.

For the number of directory structure levels you can access see Section 10.14.1.

How to Perform This Task

To restore incremental backups, use the following procedure (note that the first few steps are similar to the procedure for restoring an image backup):

1. Mount the disk to which you will restore the files, using the MOUNT /FOREIGN command. (See Section 10.8.2 for information about the MOUNT command.)
2. Load the tape, disk, or diskette that contains the most recent *image* backup of the disk. If the backup save set spans more than one volume, load the first volume of the set. If the backup copy is on a disk or diskette, mount the volume.
3. If you do not know the name of the save set, do one of the following:
   - If the save set is on a disk, make sure the disk is mounted and use the DIRECTORY command to determine the name of the save set. For example:

     ```
     $ DIRECTORY BACKUP_DISK:[BACKUPS]

     Directory SYS$SYSDEVICE:[BACKUPS]

     19APRIL1995.SAV;1

     Total of 1 file.
     ```

     The save set is named 19APRIL1995.SAV.
   - If the save set is on magnetic tape, load the tape and enter the following command, substituting the name of the tape drive you use for *MIA0*:

     ```
     $ BACKUP/LIST/REWIND MIA0:

     Listing of save set(s)

     Save set:          19APRIL1995.SAV
     Written by:        SYSTEM
     UIC:               [000001,000004]
     Date:              19-APR-1995 22:03:03.63
        .
        .
        .
     ```

     The save set is named 19APRIL1995.SAV.
4. Enter the BACKUP command using the following syntax:

   BACKUP/IMAGE device:save_set_specifier[/SAVE_SET] output_specifier

   The /IMAGE qualifier indicates that you are restoring an image backup. If your backup copy is on a disk or diskette, then you must also use the /SAVE_SET qualifier immediately after the save-set specifier (*device:save_set_specifier*).
5. If your backup copy is on more than one tape or diskette, load each subsequent tape or diskette when BACKUP prompts for the next volume.

6. Use the /NOUNLOAD qualifier to dismount the disk onto which you have just restored the files from the image backup.
7. Mount the disk that you are restoring as a file-structured volume, using the following syntax:

   MOUNT device-name: volume-label

   The parameter *device-name* is the name of the drive that holds the volume you want to mount. The parameter *volume-label* is the 1- to 6-character alphanumeric identification you assigned to the volume with the INITIALIZE command.
8. Dismount the media that contained the image backup and mount the tape, disk, or diskette that contains the most recent *incremental* backup of the disk.
9. Restore your incremental save sets, beginning with the most recent backup. Use the following syntax to restore an incremental backup:

   BACKUP/INCREMENTAL save_set_specifier[/SAVE_SET] device_specifier

   Remember that you must use the /SAVE_SET qualifier after the save-set specifier if your backup copies are on a disk or diskette.

   Continue restoring the incremental backups, from the most recent to the oldest, until you have processed all of the incremental backups since the most recent image backup. If the incremental backups are on more than one tape, diskette, or disk, then you must load each one successively when prompted by BACKUP.

   When you have processed the oldest incremental backup, the restore operation is complete.

Example

The following example shows the process of restoring an entire disk after a series of incremental backups, using the following elements and assumptions:

- The save set for the image backup is named WORK_BACKUP.SAV. This save set was created using the BACKUP/IMAGE/RECORD command.
- The save sets for the incremental backups are named as follows:

  WORK_16_JAN.SAV
  WORK_17_JAN.SAV
  WORK_18_JAN.SAV

- Both the image and the incremental backup save sets are on the disk named DUA3:, which is already mounted.
- The disk to which the files will be restored is named DUA2.

```
$ MOUNT/FOREIGN DUA2: ❶
%MOUNT-I-MOUNTED, WORK_B mounted on _DUA2:
$ BACKUP/IMAGE DUA3:WORK_BACKUP.SAV/SAVE_SET DUA2: ❷
$ DISMOUNT/NOUNLOAD  DUA2: ❸
$ MOUNT DUA2: WORK_B ❹
%MOUNT-I-MOUNTED, WORK_B mounted on _DUA2:
$ BACKUP/INCREMENTAL  DUA3:WORK_18_JAN.SAV/SAVE_SET  DUA2: ❺
$ BACKUP/INCREMENTAL  DUA3:WORK_17_JAN.SAV/SAVE_SET  DUA2: ❻
$ BACKUP/INCREMENTAL  DUA3:WORK_16_JAN.SAV/SAVE_SET  DUA2: ❼
```

In this example, the individual command lines do the following:

❶ Mount the disk DUA2: with the /FOREIGN qualifier. The files will be restored to this disk.

❷ Restore the directory structure and all the files from the save set WORK_BACKUP.SAV to the disk DUA2. This was an image backup, so it must be the first save set you restore when you want to restore incremental backup save sets.

❸ Logically dismount the disk DUA2.

❹ Remount the disk DUA2:, this time as a Files–11 volume.

❺ Restore the most recent incremental backup.

❻ Restore the next incremental backup.

❼ Restore the oldest incremental backup.

Restoring the incremental backups in reverse chronological order is the most efficient way to restore files. When you have restored the last incremental backup, the restore operation is complete.

10.16.2.1 Restoring to Target Disk Structures

BACKUP examines the target disk and the save-set contents to determine which save-set entries to ignore and which target disk entries to delete. If BACKUP encounters a privilege error when attempting to delete directories or other files from the target disk, BACKUP attempts to change the protection of the files so they can be deleted.

BACKUP detects modified directory files and will subsequently save the contents of the directory and its subdirectories to allow proper restoration of renamed directories.

Note

Renaming directories is not recommended. Also, changing security information for a directory changes its modification date. Thus, a directory might appear to be ‘‘renamed’’ and its contents included in incremental save sets if the file protection or security information is changed. The addition of renamed directory contents might increase the size of some incremental save sets.

BACKUP processes the target disk directory structure by directory levels, in alphabetical order. Thus, circumstances can occur that prevent BACKUP from correctly restoring an incremental save set to a target disk. For example, the target disk does not have sufficient space to hold newly ‘‘renamed’’ directories and their contents prior to deleting the original directories and their contents on the target disk.

If incremental restore fails due to insufficient disk space, a possible solution is to apply the incremental save set a second time (before doing anything else). This causes the first incremental restore to continue and delete directories and their contents, making more space available on the target disk. A second solution is to selectively restore files from the save set.

BACKUP attempts to restore alias or synonym file entries in incremental restore operations that do not specify multiple processing of alias or synonym file entries (/NOALIAS). In cases where the alias entry cannot be restored properly, BACKUP issues an error message indicating the alias file entry, its primary file, and a secondary status of the cause of the failure.

If you specify the /LOG qualifier, then BACKUP issues a message upon successful restoration of alias file entries.

If you specify the /VERIFY qualifier, BACKUP attempts alias entry restoration during the verify pass. Otherwise, alias entry restoration is attempted along with the normal file restoration. The reason for this behavior is that BACKUP attempts to restore all primary files before attempting to restore alias entries that will eventually reference those files.

10.16.3 Restoring Volume Shadow Sets

Because of the way volume shadowing duplicates data on each disk in the shadow set, there are special considerations for restoring a shadow set. To restore a shadow set, see *Volume Shadowing for OpenVMS*.

Note

Digital does not support a restore operation from an image save set to a virtual unit. This is because the BACKUP output device (the shadow set) must be mounted using the /FOREIGN qualifier.

10.17 Backing Up and Restoring the System Disk

Backing up your system disk is critical for the following reasons:

- A system disk could become inoperable if a problem occurs during a software upgrade, update, or installation. *Before* you attempt any of these operations, back up the system disk. If a problem occurs, you can restore the backup copy of the system disk.
- System files could inadvertently be deleted. *After* you install, upgrade, or update the operating system or any other software products, back up the system disk. If a system file is deleted you can restore the backup copy and continue to use the system.
- The drive that holds the system disk could malfunction. If you have a backup copy of the operating system and your other software, you can restore it to a functioning disk and continue to use the system.
- Disk fragmentation could occur if files are stored noncontiguously on the disk. Perform an image backup of the system disk to a magnetic tape or another disk and then restore the files to the original system disk. This restores the system disk and contiguously stores files. You can also eliminate fragmentation by performing a disk-to-disk image backup without using the /SAVE_SET qualifier. This creates a functionally equivalent copy of the entire system disk, on which files are stored contiguously. (See Section 10.17.5.)

If you have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc, back up your system using the menu-driven procedure included on the OpenVMS operating system distribution compact disc. For more information about using the new menu-driven procedure, see Section 10.17.1.

If you do not have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc, you should use standalone BACKUP to back up and restore your system disk. For more information about standalone BACKUP, see Section 10.17.2.

10.17.1 Starting the Menu-Driven Procedure

Use the menu-driven procedure in this section to back up or restore system disks and user disks if you have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc.

10.17.1.1 How to Perform This Task

1. If the operating system is not running, go to step 2.

   If the operating system is running, log in to the SYSTEM account. Enter the following command and press Return:

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

   Answer the questions. When the procedure asks if an automatic system boot should be performed, press Return for NO. When the procedure is finished, it displays the following message:

   ```
   SYSTEM SHUTDOWN COMPLETE
   ```

   VAX

   On VAX systems, the following message is also displayed:

   ```
   USE CONSOLE TO HALT SYSTEM
   ```

   Halt the system if you see this message.♦

2. Boot the system:

   Alpha

   - On OpenVMS Alpha systems, boot the distribution compact disc.♦

   VAX

   - On OpenVMS VAX systems, boot the distribution compact disc from the SYS1 directory.♦

   Note

   The boot command you use for your computer depends on the type of system you have. For more information about booting your system, see the installation and operations supplement for your computer.

3. When the system boots, it displays a menu. Choose the menu item that allows you to execute DCL commands and procedures.
4. At the DCL prompt, you can back up or restore the system and user disks.

   To make a backup copy of the system disk, see Section 10.17.3.

   To restore the system disk, see Section 10.17.4.

10.17.1.2 Example

The following example shows how to start the menu-driven procedure on an OpenVMS VAX system:

```
>>>  B/R5:10000100 ESA0
Bootfile: ISL_SVAX_061
-ESA0
 Network Initial System Load Function
 Version 1.1

  FUNCTION        FUNCTION
    ID
    1     -       Display Menu
    2     -       Help
    3     -       Choose Service
    4     -       Select Options
    5     -       Stop

 Enter a function ID value: 3
  OPTION          OPTION
    ID
    1     -       Find Services
    2     -       Enter known Service Name

 Enter an Option ID value: 2
Enter a Known Service Name: VMS061
   OpenVMS VAX Version 7.0 Major version id = 1 Minor version id = 0

%SYSINIT-E, error opening page file, status = 0000025C
%SYSINIT-E, error opening swap file, status = 0000025C
%SYSINIT, primary PAGEFILE.SYS not found; system initialization continuing
%SYSINIT, no dump file - error log buffers not saved
%SYSINIT-E, error mounting system device, status = 00000F64
$!  Copyright (c) 1995 Digital Equipment Corporation.  All rights reserved.
$set noverify


    Copyright © (c) 1995 Digital Equipment Corporation.  All rights reserved.

    Installing required known files...

    Configuring devices...

    ****************************************************************

    The menu can be used to execute DCL commands and procedures for
    various "standalone" tasks, such as backing up the system disk.

    Please choose one of the following:

        1  Execute DCL commands and procedures
        2)  Shut down this system

Enter CHOICE or "?" to repeat menu: (1/2/?)) 1

    WARNING --

    The normal VMS startup procedure has not executed.
    Some commands and utilities will not work as documented.


    Enter DCL commands -- Enter "LOGOUT" when done.
    When you enter "LOGOUT" a logout message will be displayed,
    and you will be returned to the menu.

$$$
```

10.17.2 Understanding Standalone BACKUP (VAX Only)

VAX

The Backup utility (BACKUP) does not copy open files (for example, accounting files or operator log files). For this reason you should use standalone BACKUP or the menu-driven procedure (if your configuration permits) to back up your system disk. You can boot standalone BACKUP into the main memory of your computer (while the operating system is shut down) and use a subset of BACKUP command qualifiers to perform a complete backup of *every* file on the system disk. Standalone BACKUP is supported only for OpenVMS installations and for backing up and restoring your system disk. Table 10–8 lists the qualifiers that you can use with standalone BACKUP.

Table 10–8 Valid Standalone BACKUP Qualifiers

Type	Qualifier	Default
Command Qualifiers	/BRIEF	/BRIEF
	/COMPARE	None
	/FULL	/BRIEF
	/IMAGE	/IMAGE
	/[NO]INITIALIZE	See the *OpenVMS System Management Utilities Reference Manual*
	/LIST[=filespec]	See the *OpenVMS System Management Utilities Reference Manual*
	/[NO]LOG	/NOLOG
	/PHYSICAL	None
	/RECORD	None
	/[NO]TRUNCATE	/NOTRUNCATE
	/VERIFY	None
	/VOLUME=n	None
Input Save-Set Qualifiers	/[NO]CRC	/CRC
	/[NO]REWIND	/NOREWIND
	/SAVE_SET	None
Output Save-Set Qualifiers	/BLOCK_SIZE=n	See the *OpenVMS System Management Utilities Reference Manual*
	/BY_OWNER=[uic]	See the *OpenVMS System Management Utilities Reference Manual*
	/COMMENT=string	None
	/[NO]CRC	/CRC
	/DENSITY=n	See the *OpenVMS System Management Utilities Reference Manual*
	/[NO]EXACT_ORDER	/NOEXACT_ORDER
	/GROUP_SIZE=n	/GROUP_SIZE=10
	/LABEL=(string[,...])	See the *OpenVMS System Management Utilities Reference Manual*

(continued on next page)

Table 10–8 (Cont.) Valid Standalone BACKUP Qualifiers

Type	Qualifier	Default
	/PROTECTION[=(code)]	See the *OpenVMS System Management Utilities Reference Manual*
	/[NO]REWIND	/NOREWIND
	/SAVE_SET	None
	/TAPE_EXPIRATION	Tomorrow

You should have a standalone BACKUP kit that came with your OpenVMS distribution kit; however, depending on the type of media you have, standalone BACKUP boots faster if you build it on the system disk or a user disk. The installation and upgrade supplement for your computer contains instructions for building and booting standalone BACKUP on several types of media.

This section provides information about building standalone BACKUP on a disk or tape and using it to back up your system disk.

10.17.2.1 Building Standalone BACKUP on a Disk (VAX Only)

VAX

Standalone BACKUP boots faster on disk than it does on tape. For this reason, you should create a standalone BACKUP kit on disk.

You can build standalone BACKUP on either the system disk or a user disk. If you build standalone BACKUP on a user disk, the kit occupies more disk space than if you build it on the system disk. This is because certain files that boot the system already exist on the system disk.

To build standalone BACKUP, execute SYS$UPDATE:STABACKIT.COM. The procedure copies the files for booting standalone BACKUP to a new directory on the target device that you specify, creating the directory if necessary. When you build a kit on the system disk, the procedure copies the files to the [SYSE] directory. When you build the kit on a user disk, the procedure copies the files to the [SYS0] directory.

How to Perform This Task

Perform the following steps to build standalone BACKUP on a disk:

1. Log in to the SYSTEM account.
2. Enter the following command and press Return:

```
$ @SYS$UPDATE:STABACKIT
Enter the name of the device on which to build the kit:
```

3. Enter the device name of the disk that you are building standalone BACKUP on. If you are building standalone BACKUP on the system disk, enter SYS$SYSDEVICE. For example:

```
Enter the name of the device on which to build the kit: SYS$SYSDEVICE:
```

4. The procedure places the files in the appropriate directories on the disk that you are using to build standalone BACKUP. It lists the files as they are copied. When the procedure finishes, it displays the following message:

```
The kit is complete.
```

10.17.2.2 Booting Standalone BACKUP from a Disk (VAX Only)

VAX

To boot standalone BACKUP from a disk, do the following:

1. If the operating system is not running, go to step 2.

   If the operating system is running, log in to the SYSTEM account. Enter the following command and press Return:

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

   Answer the questions. When the procedure asks if an automatic system boot should be performed, press Return for NO. When the procedure is finished, it displays the following message:

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

2. Halt the system.

3. Boot standalone BACKUP from the root where the kit is located. The exact commands for booting standalone BACKUP differ among the various computer models. See the upgrade and installation supplement for your computer for booting information.

   For example, to boot a MicroVAX 3100 computer, use the following format:

   ```
   >>> B/n0000000 device-name
   ```

   where:

   - *n* is the number of the root on the disk containing the standalone backup.
   - *device-name* is the device name of the disk.

   For example, if the disk has a device name of DKA400:, and the standalone BACKUP kit was created in the [SYSE] directory, enter the following command:

   ```
   >>> B/E0000000 DKA400
   ```

   For more information about device names, see Section 7.1.

4. Standalone BACKUP displays the following message:

   ```
   VAX/VMS Version Vn.n Major version id = 01 Minor version id = 00
   ```

5. The procedure asks you for the date and time. Enter the date and time using the 24-hour clock format and press Return. For example:

   ```
   PLEASE ENTER DATE AND TIME (DD-MMM-YYYY HH:MM) 19-JAN-1995 15:00
   ```

6. The procedure displays a list of the local devices on your system. For example:

   ```
   Available device MKA500:     device type TK50
   Available device DKA100:     device type RRD40
      .
      .
      .
   ```

   Check the list of devices. If the list is incomplete, make sure that all the drives are properly connected to the system. See your hardware manuals for details.

7. When standalone BACKUP finishes booting, it displays an identification message followed by the dollar sign prompt ($):

```
%BACKUP-I-IDENT, Standalone BACKUP Vn.n; the date is 19-APR-1995 15:00
$
```

To make a backup copy of the system disk, see Section 10.17.3.

To restore the system disk, see Section 10.17.4.

10.17.2.3 Building Standalone BACKUP on a Tape Cartridge (VAX Only)

VAX

On VAX systems with a tape cartridge distribution kit, the tape cartridge that came with your distribution kit contains standalone BACKUP. Use the procedure in this section if your copy of standalone BACKUP becomes damaged or if you want to make extra copies.

How to Perform This Task

To build standalone BACKUP on a tape cartridge, do the following:

1. Obtain a blank, initialized tape cartridge. Write the name S/A BKUP V7.0 on the paper label. Insert the label into the label slot.
2. Write-enable the tape cartridge by sliding the write-protect switch away from the label slot.
3. Insert the tape cartridge labeled S/A BKUP V7.0 into the drive.
4. Log in to the SYSTEM account.
5. Enter the following command:

```
$ @SYS$UPDATE:STABACKIT
```

6. The procedure asks you for the name of the target device. Enter the device name of the tape cartridge drive you are using to build standalone BACKUP. For example:

```
Enter the name of the device on which to build the kit: MUA0
```

7. The procedure displays the following message:

```
.
.
.
Please place the scratch tape cartridge in drive _MUA0:
This volume will receive the volume label SYSTEM.

Enter "YES" when ready:
```

8. When you are ready to continue, enter YES.
9. The system displays verification messages informing you that files are being copied.
10. When standalone BACKUP is built, the procedure displays a message similar to the following:

```
Ending time   19-MAY-1995 16:44:29.90
Starting time 19-MAY-1995 16:30:39.05

The Kit is complete.


$
```

11. Remove the tape cartridge labeled S/A BKUP V7.0 from the tape cartridge drive.

12. Write-protect the tape cartridge by sliding the write-protect switch toward the label slot. Store the cartridge in a safe place.

10.17.2.4 Booting Standalone BACKUP from a Tape Cartridge (VAX Only)

VAX

If the disk containing standalone BACKUP becomes unusable (for example, if the drive fails), you can boot standalone BACKUP from a tape cartridge. Booting standalone BACKUP from a tape cartridge takes approximately 20 minutes.

How to Perform This Task

To boot standalone BACKUP from a tape cartridge, use the following procedure:

1. If the operating system is not running, see step 2.

   If the operating system is running, log in to the SYSTEM account. Enter the following command and press Return:

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

   Answer the questions. When the procedure asks if an automatic system boot should be performed, press Return for NO. When the procedure is finished, it displays the following message:

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

2. Halt the system.
3. Insert the tape cartridge that contains standalone BACKUP into the tape cartridge drive.
4. To boot standalone BACKUP, enter the BOOT command followed by the device name of the tape cartridge drive that contains standalone BACKUP. For example:

   ```
   >>> BOOT MUA0
   ```

5. Standalone BACKUP displays the following message:

   ```
   VAX/VMS Version V7.0 Major version id = 1 Minor version id = 0
   ```

6. The procedure might ask you for the date and time. Enter the date and time using the 24-hour clock format and press Return. For example:

   ```
   PLEASE ENTER DATE AND TIME (DD-MMM-YYYY HH:MM) 19-MAY-1995 15:00
   ```

7. The procedure displays a list of the local devices on your system and, if you have them, HSC and MSCP-served devices. For example:

   ```
   Available device DUA0:              device type Generic_DU
   Available device MUA0:              device type TK50
   ```

8. When standalone BACKUP finishes booting, it displays an identification message followed by the dollar sign prompt ($):

   ```
   %BACKUP-I-IDENT, standalone BACKUP V7.0; the date is 19-MAY-1995 15:50
   $
   ```

9. Remove the tape cartridge containing standalone BACKUP from the tape cartridge drive.

   To make a backup copy of the system disk, see Section 10.17.3.

   To restore the system disk, see Section 10.17.4.♦

10.17.3 Backing Up the System Disk to Tape

When backing up your system disk, it is especially important that you understand the functions of the /IMAGE and /PHYSICAL qualifiers to the BACKUP command before using standalone BACKUP:

Qualifier	Function
/IMAGE	Lets you create a functionally equivalent copy of the entire system disk. When restored, files from an image backup are placed contiguously on the system disk, eliminating disk fragmentation.
/PHYSICAL	Copies, saves, restores, or compares the entire system disk in terms of logical blocks, ignoring any file structure.

For a complete description of the Backup utility qualifiers, see the *OpenVMS System Management Utilities Reference Manual*.

How to Perform This Task

To perform an image backup of the system disk to tape, use the following procedure:

1. Obtain blank tape cartridges or magnetic tapes that you can use for the backup operation.
2. Write-enable the tape. To write-enable a tape cartridge, slide the write-protect switch away from the tape cartridge label. To write-enable a tape, insert a write-enable ring in the back of the tape reel.
3. Insert a tape into the tape drive.
4. Determine the device name of the system disk you are backing up. (See Section 7.2 for information about determining the names of your devices.) To display the device name of the system disk you are booted from, enter the DCL command SHOW LOGICAL SYS$SYSDEVICE.
5. Depending on your configuration, either boot standalone BACKUP or start the menu-driven procedure:
   - If you have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc, start the menu-driven procedure described in Section 10.17.1.
   - If you do not have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc, boot standalone BACKUP as described in either Section 10.17.2.2 or Section 10.17.2.4.
6. Enter the BACKUP command in the following format:

   ```
   BACKUP/IMAGE/VERIFY input_specifier: output_specifier:saveset.BCK/REWIND/LABEL=label
   ```

   where:
   - *input_specifier* is the device name of the system disk.
   - *output_specifier* is the device name of the drive that you want to hold the backup copy.
   - *saveset.BCK* is the name of the save set. The name should reflect the contents of the tape (for example, DEC_31_1995.BCK) and cannot exceed 17 characters in length.

- *label* is the volume label of the tape in the drive. If the tape has been initialized already, use the same volume label that was assigned by the INITIALIZE command.

For example:

```
$ BACKUP/IMAGE/VERIFY DUA1: MUA0:DEC_31_BACKUP.BCK/REWIND/LABEL=WKY101
```

7. The following message indicates that BACKUP has transferred the files and is verifying the accuracy of the backup copy:

   ```
   %BACKUP-I-STARTVERIFY, starting verification pass
   ```

8. If your system disk contains more data than a single tape cartridge or magnetic tape can store, the procedure displays the following messages and prompt:

   ```
   %BACKUP-I-RESUME, Resuming operation on volume 2
   %BACKUP-I-READYWRITE, Mount volume 2 on _MUA0: for writing
   Enter "YES" when ready.
   ```

   If you do not receive these messages, see step 9. If you do receive these messages, do the following:

   a. Remove the backup tape from the drive.

   b. Label it COMPLETE SYSTEM BACKUP and include the date and the number of the tape in the sequence.

   c. Write-protect the backup tape.

   d. Write-enable another scratch tape and insert it into the drive.

   e. When you are ready to continue, enter Y (for YES) and press Return.

   f. The procedure displays the following message, which indicates that it has transferred the files and is verifying the accuracy of the backup copy:

      ```
      %BACKUP-I-STARTVERIFY, starting verification pass
      ```

      Each time the procedure displays a mount request, follow steps a through e.

9. If you are using standalone BACKUP, when the backup is finished, the system displays the following message:

   ```
   %BACKUP-I-PROCDONE, Operation completed. Processing finished at 19-MAY-1995
   15:30. If you do not want to perform another standalone BACKUP operation,
   use the console to halt the system.

   If you do want to perform another standalone BACKUP operation,
   ensure the standalone application volume is online and ready.
   Enter "YES" to continue:
   ```

   Continue with step 11.

10. If you are using the menu-driven procedure, the DCL prompt appears after the backup is finished. Log out and choose the Shutdown option from the menu.

11. Remove the backup tape from the drive. Label it COMPLETE SYSTEM BACKUP, number it (if you used more than one cartridge), and include the date.

12. Write-protect the tape cartridge or magnetic tape.

13. Halt the system.

14. Reboot the system.
15. Store the backup tapes in a safe place.

10.17.4 Restoring the System Disk from Tape

If a problem occurs that renders your system disk unbootable, you can restore the system disk from your backup copy.

How to Perform This Task

To restore the system disk from tape, use the following procedure.

Note

The BACKUP restore operation creates a system disk that includes a set of volume parameters provided by Digital, including a cluster size (disk access scheme). You can change most volume parameters later with the SET VOLUME command. For cluster-mounted volumes, changes occur to the nodes on which the SET VOLUME command is issued.

To change the cluster size, back up the system disk to a disk that has been previously initialized with the cluster size that you want. For more information about initializing a disk, see Section 8.3. For more information about the BACKUP command qualifiers, see the *OpenVMS System Management Utilities Reference Manual*.

1. Depending on your configuration, either boot standalone BACKUP or start the menu-driven procedure:
   - If you have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc, start the menu-driven procedure described in Section 10.17.1.
   - If you do not have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc, boot standalone BACKUP as described in either Section 10.17.2.2 or Section 10.17.2.4.
2. Determine the device name of the system disk you want to restore. (See Section 7.2 for information about determining the names of your devices.)
3. Insert the first tape of the complete system disk backup into the drive. Make sure the tape is write-protected.
4. Enter the BACKUP command in the following format:

   BACKUP/IMAGE/VERIFY input_specifier:saveset.BCK/REWIND output_specifier:

   where:
   - *input_specifier* is the device name of the drive that holds the backup copy.
   - *saveset.BCK* is the name of the save set.
   - *output_specifier* is the device name of the system disk that you are restoring.

   For example:

   ```
   $ BACKUP/IMAGE/VERIFY MUA0:DEC_31_BACKUP.BCK/REWIND DUA0:
   ```

5. The procedure displays the following message:

   ```
   %BACKUP-I-STARTVERIFY, starting verification pass
   ```

6. If your system disk contained more data than one tape could store, you receive the following messages and prompt:

```
%BACKUP-I-RESUME, Resuming operation on volume 2
%BACKUP-I-READYREAD, Mount volume 2 on MUA0: for reading
Enter "YES" when ready.
```

   If you do not receive these messages, see step 7. If you do receive these messages, do the following:

   a. Remove the backup tape from the drive.

   b. Insert the next backup tape into the drive.

   c. When you are ready to continue, enter Y (for YES) and press Return.

   d. The procedure displays the following message:

```
%BACKUP-I-STARTVERIFY, starting verification pass
```

      Each time the procedure displays a mount request, follow steps a through c.

7. If you are using standalone BACKUP, when the restore is finished the system displays the following message:

```
%BACKUP-I-PROCDONE, Operation completed. Processing finished at 19-MAY-1995
15:30. If you do not want to perform another standalone BACKUP operation,
use the console to halt the system.

If you do want to perform another standalone BACKUP operation,
ensure the standalone application volume is online and ready.
Enter "YES" to continue:
```

   Continue with step 9.

8. If you are using the menu-driven procedure, the DCL prompt appears after the restore is finished. Log out and choose the shutdown option from the menu.

9. Remove the last backup tape from the drive.

10. Halt the system.

11. Reboot the system.

12. Store the backup tapes in a safe place.

10.17.5 Backing Up the System Disk to a Disk

To eliminate disk fragmentation, perform a disk-to-disk image backup without using the /SAVE_SET qualifier. This creates a functionally equivalent copy of the entire system disk, on which files are stored contiguously.

Note

This procedure initializes the output disk, effectively erasing the files on the disk.

How to Perform This Task

To perform a disk-to-disk image backup, use the following procedure:

1. Obtain a disk with enough storage capacity to use for the backup. Make sure the disk does not contain files you need, because standalone BACKUP initializes the output disk.
2. Determine the device name of the system disk you are backing up. (See Section 7.2 for information about determining the names of your devices.) To display the device name of the system disk you are booted from, enter the DCL command SHOW LOGICAL SYS$SYSDEVICE.
3. Depending on your configuration, either boot standalone BACKUP or start the menu-driven procedure:
   - If you have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc, start the menu-driven procedure described in Section 10.17.1.
   - If you do not have access to the OpenVMS Alpha or VAX Version operating system distribution compact disc, boot standalone BACKUP as described in either Section 10.17.2.2 or Section 10.17.2.4.
4. Enter the BACKUP command in the following format:

   BACKUP/IMAGE/VERIFY input_specifier: output_specifier:

   where:

   - *input_specifier* is the device name of the system disk.
   - *output_specifier* is the device name of the drive that you want to hold the backup copy.

   For example:

   ```
   $ BACKUP/IMAGE/VERIFY DUA0: DUA1:
   ```

5. BACKUP displays the following message, which indicates that it has transferred the files and is verifying the accuracy of the backup copy:

   ```
   %BACKUP-I-STARTVERIFY, starting verification pass
   ```

6. If you are using standalone BACKUP, when the backup is finished the system displays the following message:

```
%BACKUP-I-PROCDONE, Operation completed. Processing finished at 19-MAY-1995
15:30. If you do not want to perform another standalone BACKUP operation,
use the console to halt the system.

If you do want to perform another standalone BACKUP operation,
ensure the standalone application volume is online and ready.
Enter "YES" to continue:
```

   Continue with step 8.

7. If you are using the menu-driven procedure, the DCL prompt appears after the backup is finished. Log out and choose the shutdown option from the menu.
8. You can use the backup output disk as the system disk. Files are stored contiguously on the output disk, eliminating disk fragmentation.
9. Store the original system disk.
10. Halt the system.
11. Reboot the system using the newly created system disk.

10.17.6 Using InfoServer Tapes to Back Up and Restore System Disks

VAX

You can back up the system disk to an InfoServer tape and restore the system disk from an InfoServer tape.

How to Perform This Task

1. Boot the system from the SYS1 directory using the current version of the OpenVMS CD–ROM, which can be in a reader on the InfoServer or on a local drive.

   Note

   The boot command you use for your computer depends on the type of system you have. For more information about booting your system, see the installation and operations supplement for your computer.

2. Choose option 1 from the menu-driven interface.
3. At the prompt, you can perform the backup of your system disks.

Example 10–1 shows the procedure for backing up a system disk to an InfoServer tape.

Example 10–1 System Disk Backup to an InfoServer Tape

```
>>>  B/R5:10000100 ESA0
Bootfile: ISL_SVAX_061
-ESA0
 Network Initial System Load Function
 Version 1.1

  FUNCTION        FUNCTION
    ID
    1     -       Display Menu
    2     -       Help
    3     -       Choose Service
    4     -       Select Options
    5     -       Stop

 Enter a function ID value: 3
  OPTION          OPTION
    ID
    1     -       Find Services
    2     -       Enter known Service Name

 Enter an Option ID value: 2
Enter a Known Service Name: VMS062
   OpenVMS VAX Version 7.0 Major version id = 2 Minor version id = 0

%SYSINIT-E, error opening page file, status = 0000025C
%SYSINIT-E, error opening swap file, status = 0000025C
%SYSINIT, primary PAGEFILE.SYS not found; system initialization continuing
%SYSINIT, no dump file - error log buffers not saved
%SYSINIT-E, error mounting system device, status = 00000F64
$!  Copyright (c) 1995 Digital Equipment Corporation.  All rights reserved.
$set noverify


    Copyright © (c) 1995 Digital Equipment Corporation.  All rights reserved.

    Installing required known files...

    Configuring devices...

    ****************************************************************

    The menu can be used to execute DCL commands and procedures for
    various "standalone" tasks, such as backing up the system disk.

    Please choose one of the following:

        1)  Execute DCL commands and procedures
        2)  Shut down this system

Enter CHOICE or "?" to repeat menu: (1/2/?) 1

    WARNING --

    The normal VMS startup procedure has not executed.
    Some commands and utilities will not work as documented.

    Enter DCL commands -- Enter "LOGOUT" when done.
    When you enter "LOGOUT" a logout message will be displayed,
    and you will be returned to the menu.
```

(continued on next page)

Example 10–1 (Cont.) System Disk Backup to an InfoServer Tape

```
$$$ MCR ESS$LADCP SHOW SERVICE/TAPE
$$$ MCR ESS$LADCP BIND/WRITE/TAPE TZL04_TAPE
$$$ MOUNT/FOREIGN MADn
$$$ BACKUP/IMAGE DKA100:  MADn:SYS_DISK.BCK/SAVE_SET
.
    .
    .
$$$ LOGOUT
  Process SYSTEM_1 logged out at  2-FEB-1995 23:35:17.52

    ************************************************************

    The menu can be used to execute DCL commands and procedures for
    various "standalone" tasks, such as backing up the system disk.

    Please choose one of the following:

        1)  Execute DCL commands and procedures
        2)  Shut down this system

Enter CHOICE or "?" to repeat menu: (1/2/?) ♦
```

10.18 Ensuring Data Integrity

BACKUP has several qualifiers for further ensuring the integrity of your backups. Digital recommends using these qualifiers if you want to achieve maximum data integrity. This section describes some of the ways you can increase data integrity with BACKUP. For more information about these qualifiers, see the *OpenVMS System Management Utilities Reference Manual*.

10.18.1 /CRC Qualifier

The /CRC qualifier enables the software cyclic redundancy check (CRC). The default is /CRC; you must specify /NOCRC to disable checking. Disabling checking reduces processing time, but increases the risk of data error.

As an output save-set qualifier, /CRC writes the CRC checking code into the blocks of the output save set.

As an input save-set qualifier, /CRC checks the CRC information in the input save set.

Digital recommends that you use the CRC. Although it increases processing time, it also improves data integrity.

10.18.2 /GROUP_SIZE Qualifier

This output save-set qualifier causes BACKUP to write redundant data to the output save set. This allows BACKUP to attempt to correct read errors during the backup restore operation. Use the /GROUP_SIZE qualifier to define the number of blocks in each group of redundant information. For example:

```
$ BACKUP/IMAGE/RECORD
_From: DKA100:
_To: MKB100:BACKUP.SAV/LABEL=WKY101/GROUP_SIZE=20
```

This command adds a recovery block after every 20 blocks of saved data. This allows BACKUP to recover a corrupted data block for every 20 blocks of saved data. The value of the /GROUP_SIZE qualifier defaults to 10.

Although using this qualifier increases the size of the save set and the processing time for the operation, Digital recommends using the /GROUP_SIZE qualifier to increase data integrity.

10.18.3 /IGNORE Qualifier

Digital recommends that you back up your system when no interactive users are logged in. This is because if BACKUP encounters an open file during a save operation, it issues an error message and *does not* copy the file.

You can instruct the backup procedure to save open files by using the /IGNORE=INTERLOCK qualifier on the BACKUP command. When you use the /IGNORE=INTERLOCK qualifier, the contents of the file at the moment of the backup are saved.

The /IGNORE=INTERLOCK qualifier is useful for files that are constantly open (and would therefore not otherwise be saved). However, you must recognize that you might be saving inconsistent data, depending on the applications that are writing to the open files (for example, open application transactions or file data cached in memory). Also, because of the way BACKUP scans directories, *any* activity in a directory (such as creating or deleting files) can cause files to be excluded from the backup. In general, it is best to back up your system when a minimum number of files are open.

Also, because of the way the file system works, using the /IGNORE=INTERLOCK qualifier to back up open files affects subsequent incremental backups. For example, you can back up an open file with the BACKUP/IMAGE/RECORD /IGNORE=INTERLOCK command. However, the backup date field of the file is not updated until you close the file. If the file remains open during subsequent incremental backups, it is not included in those backups because its backup date field is not as recent as the last image backup.

10.18.4 /LOG Qualifier

Use the /LOG qualifier to the BACKUP command to display the file specification of the files that BACKUP processes during a backup operation. For example, if you are copying files in a directory, you can use the /LOG qualifier to display the file specification of each file copied:

```
$ BACKUP/LOG
_From: WORK3:[OCONNELL]*.*
_To: WORK1:[OCONNELL.SCRATCH]*.*
%BACKUP-S-CREDIR, created WORK1:[OCONNELL.SCRATCH.COM]
%BACKUP-S-CREATED, created WORK1:[OCONNELL.SCRATCH]DECW$MAIL.DAT;2
%BACKUP-S-CREATED, created WORK1:[OCONNELL.SCRATCH]DECW$SM.LOG;42
%BACKUP-S-CREATED, created WORK1:[OCONNELL.SCRATCH]DECW$SM.LOG;41
   .
   .
   .
```

10.18.5 /VERIFY Qualifier

Use the /VERIFY qualifier to cause BACKUP to compare the contents of the input and output specifiers after a save, restore, or copy operation. When BACKUP is executing the verification pass, it displays the following message:

```
%BACKUP-I-STARTVERIFY, starting verification pass
```

If BACKUP finds differences between the input and output files, it issues an error message.

Digital recommends that you use the /VERIFY qualifier. Although it increases processing time, it also improves data integrity.

10.19 Troubleshooting

This section describes some common BACKUP errors and how to recover from them.

10.19.1 BACKUP Fatal Error Options

If, in the course of a backup operation, the Backup utility or standalone BACKUP encounters fatal hardware- or media-related errors or encounters more media errors than considered reasonable for data reliability, BACKUP generates the following informational message and prompt:

```
%BACKUP-I-SPECIFY, specify option (CONTINUE, RESTART, QUIT)
BACKUP>
```

Note

If BACKUP is running interactively and you used the command qualifier /NOASSIST, you can enter an option in response to the BACKUP> prompt. If BACKUP is executing as a batch job or you specified the command qualifier /ASSIST, the operator must use the DCL command REPLY to enter an option.

The option you choose depends on several factors. See Table 10–9.

Table 10–9 BACKUP Error Options and Results

Option	Restrictions	Result
CONTINUE	May compromise data reliability. Use only if the position of the tape has not changed since the original error and if the error does not imply that data has already been lost.	If possible, BACKUP ignores the error and continues processing.
RESTART	Not valid if the output volume is the first volume in the backup operation.	BACKUP unloads the current tape from the drive and prompts for another volume. After you load another tape, BACKUP restarts the save operation from the point at which the original tape was mounted.
QUIT	None.	BACKUP terminates the operation and you can reenter the command.

The following example illustrates the sequence of events that occurs when BACKUP encounters an excessive rate of media errors on VOL3 and you choose the RESTART option:

1. BACKUP indicates that the magnetic tape has an excessive number of media errors and displays the following error message and prompt:

```
%BACKUP-F-WRITEERRS, excessive error rate writing VOL3
%BACKUP-I-SPECIFY, specify option (CONTINUE, RESTART, QUIT)
BACKUP>
```

2. Enter RESTART.
3. BACKUP dismounts VOL3 and prompts for a new tape. Remove VOL3 from the drive and discard this tape.
4. Place a new tape into the drive and enter YES in response to the prompt for a new tape.
5. BACKUP restarts the save operation from the beginning of VOL3; no data is lost.

10.19.2 Tape Label Errors

When you instruct BACKUP to use a tape that has a label other than the one you specified, BACKUP issues the following message:

```
%MOUNT-I-MOUNTED, DKA0 mounted on _SODAK$MUA0:
%BACKUP-W-MOUNTERR, volume 1 on _SODAK$MUA0 was not mounted because
its label does not match the one requested
%BACKUP-W-EXLABEER, volume label processing failed because
 volume TAPE4 is out of order, Volume label TAPE1 was expected
 specify option (QUIT, NEW tape, OVERWRITE tape, USE loaded tape)
BACKUP>
```

Depending on the option you specify, you can quit the backup operation (QUIT), dismount the old tape and mount a new one (NEW), overwrite the data on the tape (OVERWRITE), or USE the loaded tape.

If you use blank tapes or tapes that you intend to overwrite, use the /IGNORE=LABEL_PROCESSING qualifier. This suppresses the previous BACKUP message, which normally occurs if BACKUP encounters a non-ANSI-labeled tape during a save operation.

11
Security Considerations

This chapter outlines the security features available with the OpenVMS operating system and suggests procedures to reduce the threat of a break-in on your system or cluster. It also tells how to use the access control list editor (ACL editor) to create and modify access control list entries (ACEs) on protected objects. For a more detailed description of security management, refer to the *OpenVMS Guide to System Security*.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Managing passwords	Section 11.2
Adding to the system password dictionary	Section 11.2.1
Setting up intrusion detection	Section 11.3
Interpreting a user identification code (UIC)	Section 11.4.1
Parsing a protection code	Section 11.4.2
Creating access control lists (ACLs)	Section 11.5
Using the access control list editor (ACL editor)	Section 11.7
Auditing security-relevant events	Section 11.8.1

This chapter explains the following concepts:

Concept	Section
What security management involves	Section 11.1
Aspects of password management	Section 11.2
Ways to protect objects	Section 11.4
Construction of access control lists (ACLs)	Section 11.5
Audit log file analysis	Section 11.9

For full descriptions of all these tasks and concepts, refer to the *OpenVMS Guide to System Security*.

11.1 Understanding Security Management

As the person responsible for the day-to-day system management, you play an important role in ensuring the security of your system. Therefore, you should familiarize yourself with the security features available with the OpenVMS operating system and implement the features needed to protect systems, users, and files from damage caused by tampering. Effective operating system security measures help prevent unauthorized access and theft of proprietary software,

software plans, and computer time. These measures can also protect equipment, software, and files from damage caused by tampering.

Types of Security Problems

Security problems on most systems are generally caused by irresponsibility, probing, or penetration. The tolerance that your site might have to a breach of security depends on the type of work that takes place at your site.

Environmental Considerations

A secure system environment is a key to system security. Digital strongly encourages you to stress environmental considerations when reviewing site security.

Operating System Protections

In the OpenVMS operating system, managing system security is concerned with three major areas:

- Controlling access to the system; for example, interactively, through batch processing jobs, or over the network
- Controlling access to information and resources that are kept on the system; for example, files, application programs, or system utilities
- Managing the auditing system so security-relevant events are logged, the log file is reviewed on a regular basis, and the log is kept to a reasonable size

The following sections describe measures to control access to your system and its resources.

11.2 Managing Passwords

A site needing average security protection always requires the use of passwords. Sites with more security needs frequently require generated passwords and system passwords. Highly secure sites sometimes choose to use secondary passwords to control network access.

11.2.1 Initial Passwords

When you open an account for a new user with the Authorize utility, you must give the user a user name and an initial password. When you assign temporary initial passwords, observe all guidelines recommended in Section 11.2.5. You should consider using the automatic password generator. Avoid any obvious pattern when assigning passwords.

Using the Automatic Password Generator

To use the automatic password generator while using the Authorize utility to open an account, add the /GENERATE_PASSWORD qualifier to either the ADD or the COPY command. The system responds by offering you a list of automatically generated password choices. Select one of these passwords, and continue setting up the account.

Using the System Dictionary and the Password History List

The OpenVMS operating system automatically compares new passwords with a system dictionary to ensure that a password is not a native language word. It also maintains a password history list of a user's last 60 passwords. The operating system compares each new password with entries in the password history list to ensure that an old password is not reused.

The system dictionary is located in SYS$LIBRARY. You can enable or disable the dictionary search by specifying the DISPWDDIC or NODISPWDDIC option with the /FLAGS qualifier in AUTHORIZE. The password history list is located in SYS$SYSTEM. To enable or disable the history search, specify the DISPWDHIS or NODISPWDHIS option to the /FLAGS qualifier.

Adding to the System Password Dictionary

You might want to modify the system password dictionary to include words of significance to your site. The following procedure allows you to add words to the system dictionary. The procedure also allows you to retain a file of the passwords that you consider unacceptable.

1. Create a file containing passwords you would like to add to the dictionary. Each password should be on a separate line and in lowercase, as follows:

```
$ CREATE LOCAL_PASSWORD_DICTIONARY.DATA
somefamous
localheroes
Ctrl/Z
```

2. Enable SYSPRV and merge your local additions:

```
$ SET PROCESS/PRIVILEGE=SYSPRV
$ CONVERT/MERGE/PAD LOCAL_PASSWORD_DICTIONARY.DATA -
_$ SYS$LIBRARY:VMS$PASSWORD_DICTIONARY.DATA
```

Defining Preexpired Passwords

When you add a new user to the UAF, you might want to define that user's password as having expired previously using the AUTHORIZE qualifier /PWDEXPIRED. This forces the user to change the initial password when first logging in.

Preexpired passwords are conspicuous in the UAF record listing. The entry for the date of the last password change carries the following notation:

(pre-expired)

By default, the OpenVMS operating system forces new users to change their passwords the first time they log in. Encourage your site to use a training program for its users that includes information about changing passwords.

11.2.2 System Passwords

System passwords are used to control access to terminals that might be targets for unauthorized use, as follows:

- All terminals using dialup lines or public data networks for access
- Terminals on lines that are publicly accessible and not tightly secured, such as those at computer laboratories at universities
- Terminals the security manager wants to reserve for security operations

Implementing system passwords is a two-stage operation involving the DCL commands SET TERMINAL and SET PASSWORD. First, you must decide which terminals require system passwords. Then, for each terminal, you enter the DCL command SET TERMINAL/SYSPASSWORD/PERMANENT. To enable system passwords for all terminals, set the appropriate bit in the system parameter TTY$DEFCHAR2.

11.2.3 Primary and Secondary Passwords

The use of dual passwords is cumbersome and mainly needed at sites with high-level security concerns. The effectiveness of a secondary passwords depends entirely on the trustworthiness of the supervisor who supplies it. A supervisor can easily give out the password or worse yet, change it to a null string.

The main advantage of a second password is that it prevents accounts from being accessed through DECnet for OpenVMS using simple access control.

Another advantage of a second password is that it can serve as a detection tool when a site has unexplained break-ins after the password has been changed and the use of the password generator has been enforced. Select problem accounts, and make them a temporary target of this restriction. If the problem goes away when you institute personal verification through the secondary password, you know you have a personnel problem. Most likely, the authorized user is revealing the password for the account to one or more other users who are abusing the account. See the *OpenVMS Guide to System Security* for an explanation of how to add secondary passwords.

11.2.4 Enforcing Minimum Password Standards

Security managers can use AUTHORIZE to impose minimum password standards for individual users. Specifically, qualifiers and login flags provided by AUTHORIZE control the minimum password length, how soon passwords expire, and whether the user is forced to change passwords at expiration.

Password Expiration

With the AUTHORIZE qualifier /PWDLIFETIME, you can establish the maximum length of time that can elapse between password changes before the user will be forced to change the password or lose access to the account.

The use of a password lifetime forces the user to change the password regularly. The lifetime can be different for different users. Users who have access to critical files generally should have the shortest password lifetimes.

Forcing Expired Password Changes

By default, users are forced to change expired passwords when logging in. Users whose passwords have expired are prompted for new passwords at login. A password is valid for 90 days unless a site modifies the value with the /PWDLIFETIME qualifier.

Minimum Password Length

With the AUTHORIZE qualifier /PWDMINIMUM, you can direct that all password choices must be a minimum number of characters in length. Users can still specify passwords up to the maximum length of 32 characters.

Requiring the Password Generator

The /FLAGS=GENPWD qualifier in AUTHORIZE allows you to force the use of the automatic password generator when a user changes a password. At some sites, all accounts are created with this qualifier. At other sites, the security manager can be more selective.

11.2.5 Guidelines for Protecting Passwords

Observe the following guidelines to protect passwords:

- Make certain the password for the SYSTEM account, which is a standard account on all OpenVMS systems, is secure and is changed regularly.
- Disable any accounts that are not used regularly with the AUTHORIZE qualifier /FLAGS=DISUSER (for example, SYSTEST and FIELD).
- Do not permit an outside or an in-house service organization to choose the password for an account they use to service your system. Such service groups tend to use the same password on all systems, and their accounts are usually privileged. On seldom-used accounts, set the AUTHORIZE flag DISUSER, and enable the account only when it is needed. You can also change the password immediately after each use and notify the service group of the new password.
- Set appropriate account expiration dates, especially when you know a user has only short-term requirements for an account.
- Delete accounts no longer in use.
- If you have an account on a system that stores passwords in plaintext (unencrypted), choose a different password on all of your accounts on other machines. Passwords should not even be shared between machines that encrypt the password. A compromised machine can be used to read plaintext on its way into the machine, thereby gaining access to the other machine.
- Do not leave listings of operator logs, accounting logs, or audit logs where they could be read or stolen.
- Maintain adequate protection of authorization files. Note that the system user authorization file (SYSUAF.DAT) and network proxy authorization file (NETPROXY.DAT) are owned by the system account ([SYSTEM]). There should be no other users in this group. Accordingly, the categories SYSTEM, OWNER, and GROUP are synonymous. Normally the default UIC-based file protection for these authorization files is adequate.

The following actions are not strictly for password protection, but they reduce the potential of password detection or limit the extent of the damage if passwords are discovered or bypassed:

- Avoid giving multiple users access to the same account.
- Separate users into distinct user groups.
- Protect telephone numbers for dialup lines connected to your system.
- Make all accounts that do not require a password captive accounts.
- Extend privileges to users carefully.
- Ensure that the files containing components of the operating system are adequately protected.

11.2.6 Password History

The password history database maintains a history of previous passwords associated with each user account. By default, the system retains these records for one year. Password history records that are older than the system password history lifetime are allowed as valid password choices. When a user account is deleted, the system removes the associated password history records from the history database.

11.3 Using Intrusion Detection Mechanisms

This section describes how to set up intrusion detection and evasion and how to display the intrusion database.

Controlling the Number of Retries on Dialups

You can control the number of login attempts the user is allowed through a dialup line. If the user makes a typing mistake after obtaining the connection, the user does not automatically lose the connection. This option is useful for authorized users, while still restricting the number of unauthorized attempts.

To implement control of retries, use the following two **LGI system parameters**: LGI_RETRY_TMO and LGI_RETRY_LIM. If you do not change the values of these system parameters, the default values allow the users three retries with a 20-second interval between each.

You should keep in mind that controlling dialup retries is only a part of an overall security program and is not, in itself, sufficient to avoid break-ins. An obstacle like redialing is not going to prove an effective deterrent to a persistent intruder.

Discouraging Break-in Attempts Further

The OpenVMS operating system offers additional methods of discouraging break-in attempts. These methods also use system parameters in the LGI category.

Parameter	Description
LGI_BRK_LIM	Defines a threshold count for login failures. When the count of login failures exceeds the LGI_BRK_LIM value within a reasonable time interval, the system assumes that a break-in is in progress.
LGI_BRK_TERM	Controls the association of terminals and user names for counting failures.
LGI_BRK_TMO	Controls the time period in which login failures are detected and recorded.
LGI_HID_TIM	Controls the duration of the evasive action.
LGI_BRK_ DISUSER	Makes the effects of intrusion detection more severe. If you set this parameter to 1, the OpenVMS operating system sets the DISUSER flag in the UAF record for the account where the break-in was attempted. Thus, that user name is disabled until you manually intervene.

See the *OpenVMS Guide to System Security* for a full description of these parameters.

Displaying the intrusion Database

The Security Server process, which is created as part of normal operating system startup, performs the following tasks:

- Creates and manages the system's intrusion database
- Maintains the network proxy database file (NET$PROXY.DAT)

The intrusion database is used by the system to keep track of failed login attempts. This information is scanned during process login to determine if the system should take restrictive measures to prevent access to the system by a suspected intruder.

Use the DCL command SHOW INTRUSION to display the contents of the intrusion database. Use the DCL command DELETE/INTRUSION to remove entries from the intrusion database.

The network proxy database file (NET$PROXY.DAT) is used during network connection processing to determine if a specific remote user may access a local account without using a password. The information contained in this database is managed by the Authorize utility.

Example 11–1 shows the expanded expiration time field in the new SHOW INTRUSION output.

Example 11–1 SHOW INTRUSION Output

```
$ SHOW INTRUSION
Intrusion       Type       Count        Expiration         Source
   NETWORK      SUSPECT        1   21-MAY-1995 12:41:01.07  DEC:.ZKO.TIDY::SYSTEM
```

11.4 Understanding Ways to Protect Objects

The OpenVMS operating system offers two primary protection mechanisms. The first, **UIC-based protection**, is based on the user identification code (UIC) and is applied to all protected objects.

The second protection mechanism uses **access control lists (ACLs)**, which employ a more refined level of protection than that available with UIC-based protection. ACLs can be used to grant or deny access to individual users or groups of users.

11.4.1 Interpreting a User Identification Code

Your user identification code (UIC) tells what group you belong to and what your unique identification is within that group.

The Authorize utility assigns each user process in the system a unique UIC in the user authorization file (UAF). Each object on the system is also associated with a UIC (typically the UIC of its creator).

A UIC consists of two parts, group and member, specified in the following format:

[group,member]

A UIC can be either numeric or alphanumeric. A numeric UIC consists of a group number in the range 0 through 37776 (octal) and a member number in the range 0 through 177776 (octal). Digital reserves group 1 and groups 300–377.

11.4.2 Understanding Protection Codes

A protection code controls the type of access allowed (or denied) to a particular user or group of users. It has the following format:

[user category: list of access allowed (, user category: list of access allowed,...)]

user category

User categories include system (S), owner (O), group (G), and world (W). Each category can be abbreviated to its first character. Categories have the following definitions:

- **System**: Members of this category can include any of the following:
  - Users with low group numbers, usually from 1 to 10 (octal). These group numbers are generally for system managers, security administrators, and system programmers. (The exact range of system group numbers is determined by the security administrator in the setting of the system parameter MAXSYSGROUP. It can range as high as 37776 (octal).)
  - Users with the SYSPRV privilege.
  - Users with the GRPPRV privilege whose UIC group matches the UIC group of the object's owner.
  - In access requests to files on a disk volume, users whose UIC matches the UIC of the volume's owner.
- **Owner**: The user with the same UIC as the user who currently owns the object. In general, the creator of an object is entitled to owner access unless explicit action is taken to secure the object from its creator.
- **Group**: All users who are in the same UIC group as the object's owner.
- **World**: All users, including those in the first three categories.

When specifying more than one user category, separate the categories with commas, and enclose the entire code in parentheses. You can specify user categories and access types in any order.

A null access specification means no access, so when you omit an access type for a user category, that category of user is denied that type of access. To deny all access to a user category, specify the user category without any access types. Omit the colon after the user category when you are denying access to a category of users.

When you omit a user category from a protection code, the current access allowed that category of user remains unchanged.

access-list

Access types are object-dependent and are described in the *OpenVMS Guide to System Security*. For files, the access types include read (R), write (W), execute (E), and delete (D). The access type is assigned to each user category and is separated from its user category by a colon (:).

Example

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

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

How to Change the Default Protection

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

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

11.5 Creating Access Control Lists (ACLs)

For most interactive user accounts, the default UIC-based protection is adequate. However, in some cases (such as project accounts) you may want to set up an additional level of protection by using **access control lists (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.

11.5.1 Kinds of Entries in an ACL

An access control list (ACL) is a list of entries, each of which defines some attribute of an object. Each entry is called an **access control entry (ACE)**.

The following security-relevant types of ACEs are available:

ACE	Description
Identifier ACE	Controls the types of access allowed to specific users based on the user's identification. Each Identifier ACE includes one or more rights identifiers and a list of the types of access the user holding the identifier has permission to exercise. See Section 11.5.2 for a summary of identifiers. For example, the following ACE grants the user Jones read, write, and execute access to an object: `(IDENTIFIER=[ACCOUNTING,JONES],ACCESS=READ+WRITE+EXECUTE)`
Default Protection ACE	Allows you to specify a protection code for a directory file that is propagated to all files created within that directory and its subdirectories. For example, the following ACE assigns a protection code to newly created files in a directory. The code gives users in the system and owner categories full access, it gives group users both read and execute access, and it denies access to users in the world category. `(DEFAULT_PROTECTION,S:RWED,O:RWED,G:RE,W:)`
Creator ACE	Adds an extra ACE to the ACL of a file created within the directory to which you assign the Creator ACE. The Creator ACE applies when the file being created is not owned by the user identification code (UIC) of the process creating the file, such as when the directory is owned by a resource identifier. The following ACE, for example, specifies that any user creating a file in the directory will receive read, write, execute, and delete access to it: `(CREATOR,ACCESS=READ+WRITE+EXECUTE+DELETE)` The Creator ACE applies to directory files only.

ACE	Description

Security Alarm ACE	Allows you to request that a security alarm message be sent to the operator's terminal if an object is accessed in a particular way.
	For example, the following ACE causes an alarm message whenever a particular file is successfully read:
	`(ALARM=SECURITY,ACCESS=SUCCESS+READ)`
	The security Alarm ACE has no effect unless ACL alarms are enabled with the following command:
	`$ SET AUDIT/ALARM/ENABLE=(ACL)`

Security Audit ACE	Specifies the access criteria that cause a security alarm message be sent to the system security audit log file if an object is accessed in a particular way. For example, the following ACE causes an alarm message whenever a particular file is successfully read: `(AUDIT=SECURITY,ACCESS=SUCCESS+READ)` A message is recorded only if ACL audits are enabled with the DCL command SET AUDIT/AUDIT/ENABLE=ACL.
Subsystem ACE	Grants additional identifiers to a process while it is running the image to which the Subsystem ACE applies. Users with execute access to the image can access objects that are in the protected subsystem, such as data files and printers, but only when they run the subsystem image. The Subsystem ACE applies to executable images only. For example, the following ACE adds the identifier ACCOUNTING to processes that are executing a particular subsystem image. The identifier entitles the processes to access objects owned by the subsystem. `(SUBSYSTEM, IDENTIFIER=ACCOUNTING)`

See the *OpenVMS System Management Utilities Reference Manual* for a complete description of each kind of ACE. The *OpenVMS Guide to System Security* provides further details on how to construct and apply ACEs.

11.5.2 Types of Identifiers

An Identifier ACE can contain different types of identifiers. Any of these identifiers is an alphanumeric string of 1 to 31 characters with at least one alphabetic character. Valid characters include numbers 0 to 9, characters A to Z, the dollar sign ($), and the underscore (_). The following table lists each type of identifier:

Type	Description	Example
UIC identifiers	Based on a user's identification code (UIC), which uniquely identifies a user on the system and defines the group to which the user belongs.	[GROUP1,JONES] [JONES] GROUP1 JONES
General identifiers	Defined by the security administrator.	SALES RESERVE_DESK
Environmental identifiers	Describe different types of users based on their initial entry into the system. These identifiers are automatically created by the system.	BATCH, NETWORK INTERACTIVE LOCAL, DIALUP REMOTE

Type	Description	Example
Facility identifiers	Defined by a facility during installation	RDB$ENTRY

In addition to the environmental identifiers, a system node identifier of the form SYS$NODE_*node_name* is created by the system startup procedure (STARTUP.COM in SYS$SYSTEM).

11.6 Assigning ACLs

You can place ACLs on the following object classes:

Capability
Device
Common event flag cluster
File
Group global section
Logical name table
Queue
Resource domain
Security class
System global section
Volume

Typically, ACLs are used when you want to provide access to an object for some, but not all, users, or if you want to deny access to specific, unprivileged users. When the operating system receives a request for access to an object having an ACL, it searches each access control list entry in the ACL, stopping at the first match. If another match occurs in the ACL, it has no effect. Therefore, ACEs granting or denying access to a protected object for specific users should appear in the ACL before ACEs identifying broader classes of users.

11.7 Using the ACL Editor

The access control list editor (ACL editor) is a screen-oriented editor used to create and maintain ACLs. Use the ACL editor to define an ACL for a protected object or to edit an existing ACL.

You can use either the EDIT/ACL command or the SET SECURITY/EDIT command to invoke the ACL editor. In the command line, specify the name of the object whose ACL you want to create or modify. For example, the following command invokes the ACL editor to create an ACL for the file INVENTORY.DAT:

```
$ EDIT/ACL INVENTORY.DAT
```

If the object whose ACL you want to create or modify is not a file, you must specify the type of object with the /CLASS qualifier. For example, the following command invokes the ACL editor to create an ACL for the disk DOCD$:

```
$ EDIT/ACL/CLASS=DEVICE DOCD$
```

You can invoke the ACL editor to modify an existing ACL or to create a new ACL on the object. If an object has an ACL, the ACL will appear on the screen when the ACL editor is invoked.

The ACL editor can be invoked from within a program written in any OpenVMS common language that generates calls using the OpenVMS calling standard. Refer to the *OpenVMS Utility Routines Manual* for more information on using the callable interface to the ACL editor.

11.7.1 Adding an Identifier ACE

An Identifier ACE controls the types of access allowed to a particular user or group of users. It has the following format:

(IDENTIFIER=identifier[,options][,access])

For example, the following ACE grants user Pat, who is identified by the UIC identifier [SALES,PAT], read, write, and execute access to a file. The ACL denies Pat delete and control access because it omits them from the access statement.

```
(IDENTIFIER=[SALES,PAT],ACCESS=READ+WRITE+EXECUTE)
```

The Default attribute of an Identifier ACE allows users to define one or more default ACEs for inclusion in the ACLs for newly created files in a particular directory. Thus, if you wanted all files in the directory [MALCOLM] to have an ACE that permitted read and write access to users with the PERSONNEL identifier, you could include the following ACE in the ACL for the file MALCOLM.DIR:

```
(IDENTIFIER=PERSONNEL,OPTIONS=DEFAULT,ACCESS=READ+WRITE)
```

As a result of this ACE, any file created in the [MALCOLM] directory has the following ACE:

```
(IDENTIFIER=PERSONNEL,ACCESS=READ+WRITE)
```

See the *OpenVMS Guide to System Security* for further discussion of the Default attribute and its effect on the processing of an ACL.

11.7.2 Setting a Default Protection Code

A Default Protection ACE defines a protection code for all files that are subsequently created in the directory and in any subdirectories under that directory, unless protection is specified for one of those files individually. The ACE does not apply if a previous version of the file exists (in this case, the previous file protection is used). This ACE type has the following format:

(DEFAULT_PROTECTION[,options],protection-code)

For example, the following ACE specifies that users in the system and owner categories have read, write, execute, and delete access to any files subsequently created in the directory, and that group and world users have no access:

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

Note

The Default Protection ACE does not apply to existing subdirectories. It applies to subdirectories created after the ACE is applied to the parent directory.

11.7.3 Generating Security Alarms and Audits

Security ACEs allow you to specify that an event message be sent when a protected object is accessed in a particular manner. The security Alarm ACE directs the event message to the security operator's terminal and the security Audit ACE directs the event message to the system security audit log file.

Refer to the *OpenVMS Guide to System Security* for more information on how to use these types of ACEs.

11.8 Auditing Security-Relevant Events

System managers can select the destination for security-relevant event messages. Alarm messages are sent to the operator's terminal and audit messages are sent to the system security audit log file. You can choose to have an event reported as an alarm, as an audit, or as both.

11.8.1 Enabling Classes of Security Alarms

The OpenVMS operating system automatically monitors a certain number of events, as listed in Table 18-6.

You can enable additional classes of events by listing one or more of the keywords of the /ENABLE qualifier to the DCL command SET AUDIT listed in Table 11–1.

Table 11–1 Kinds of Security Events OpenVMS Can Report

Event Class	Description
Access	Specifies access events for all objects in a class. You can audit selected types of access, both privileged and nonprivileged, to all protected objects of a particular class.
ACL	Events requested by a security Audit or Alarm ACE in the access control list (ACL) of an object.
Authorization	Modification of any portion of SYSUAF.DAT, NETPROXY.DAT, NET$PROXY.DAT, or RIGHTSLIST.DAT.
Breakin	Breakin attempts.
Connection	Logical link connections or terminations through SYSMAN, DECnet for OpenVMS Phase IV, DECwindows products, or an interprocess communication (IPC) call.
Create	Creation of a protected object.
Deaccess	Deaccess from a protected object.
Delete	Deletion of a protected object.
Identifier	Use of identifiers as privileges.
Install	Modifications made to the known file list through the Install utility.
Logfailure	Failed login attempts.
Login	Successful login attempts.
Logout	Logouts.
Mount	Volume mounts and dismounts.
NCP	Modification to the network configuration database, using the network control program (NCP).

(continued on next page)

Table 11–1 (Cont.) Kinds of Security Events OpenVMS Can Report

Event Class	Description
Privilege	Successful or unsuccessful use of privilege.
Process	Use of one or more of the process control system services.
SYSGEN	Modification of a system parameter with the System Generation utility (SYSGEN) or AUTOGEN.
Time	Modification of system time.

See the *OpenVMS DCL Dictionary* for more information about the SET AUDIT command.

11.9 Analyzing Audit Log Files

The Audit Analysis utility (ANALYZE/AUDIT) enables system managers and site security administrators to selectively extract and display information from security audit log files. Using ANALYZE/AUDIT qualifiers, you can choose from among a variety of report formats and select the event criteria to be included in the report. See the *OpenVMS Guide to System Security* for a description of how to use the utility.

12

Managing the Queue Manager and Queue Database

This chapter describes tasks you perform to set up and manage the queue manager and queue database for the OpenVMS batch and print queuing system. You must set up the queue manager and queue database before creating and starting queues.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Specifying the location of queue database files	Section 12.3
Starting the queue manager and creating the queue database	Section 12.4
Customizing queue manager failover	Section 12.5
Stopping the queue manager	Section 12.6
Restarting the queue manager	Section 12.7
Creating an additional queue manager	Section 12.8
Displaying information about the queue manager	Section 12.9
Saving the queue database	Section 12.10
Restoring the queue database	Section 12.11
Maximizing queue system performance	Section 12.12
Solving queue manager and queue database problems	Section 12.13
Reporting a queuing system problem to Digital	Section 12.14

This section explains the following concepts:

Concept	Section
The queue manager	Section 12.1
The queue database	Section 12.2

12.1 Understanding the Queue Manager

If you have a printer on your system, or if you want to use batch processing, you must use **queues**. The **queue manager** controls queuing activity. The **queue database** stores information about queues and jobs. Before you can perform any queue operation, you must start the queue manager and create the queue database. For information, see Section 12.4.

Figure 12–1 OpenVMS Batch and Print Queuing System

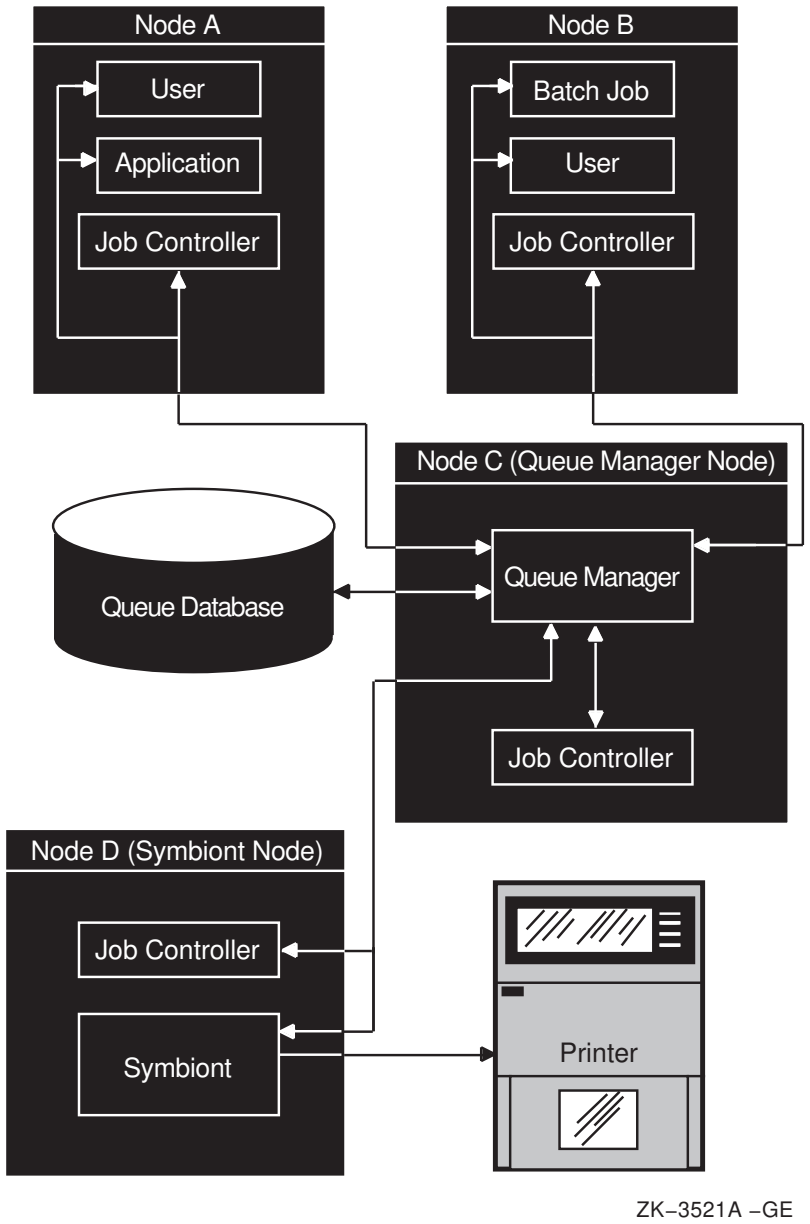


Figure 12–1 illustrates how the queue manager works to manage queue activity in a VMScluster environment.

When a user submits a batch or print job to a queue, the queue manager does the following:

1. Receives the user's queue request, including information about the type of job, the file name or names, the name of the queue, and any special options.
2. Stores and retrieves appropriate information from the queue database to print or execute the job.
3. Places the job in the appropriate queue to await its turn for processing.

Print jobs are sent to an independent process, called a **symbiont**, for formatting and are finally sent to the printer for printing. For batch jobs, the **job controller** creates a process to perform the tasks in the batch job.

A single queue manager process controls queuing for all processes on a node or in a VMScluster environment. Jobs can be submitted from one node and executed on a queue running on another VMScluster node. User processes, symbionts, and job controllers on each node communicate directly with the queue manager.

In addition, the job controller works with the queue manager to perform the following queue management tasks:

- Create and monitor batch, symbiont, and queue manager processes
- Restart the queue manager process on reboot
- Handle failover of the queue manager in a VMScluster environment

12.1.1 Queue Manager Failover

By default, in a VMScluster environment, the queue manager will attempt to fail over to another node if the node on which the queue manager is running leaves the VMScluster. You can specify the order in which VMScluster nodes should claim the queue manager process or limit the nodes that can run the queue manager. For more information, see Section 12.4.

12.1.2 Multiple Queue Managers

To work around CPU, disk space, or memory limitations in a VMScluster environment, you can use multiple queue managers to distribute the batch and print work load among nodes as well as to distribute the database among disks.

For example, you might create separate queue managers for batch queues and print queues. You could then run the batch queue manager on one node and the print queue manager on a different node. You might also choose to maintain the databases on separate disks.

For information about creating additional queue managers, see Section 12.8. Use the command SHOW QUEUE/MANAGER to display information about queue managers on the system. For more information, see Section 12.9.

Restrictions for Using Multiple Queue Managers

Multiple queue managers have the following restrictions:

- Queues running on one queue manager cannot reference queues running on a different queue manager. For example, a generic queue running on queue manager A cannot direct jobs to an execution queue running on queue manager B.
- You cannot move a job from a queue on one queue manager to a queue on a different queue manager.
- The operating system allows a maximum of five queue managers in a VMScluster environment.

Naming of Multiple Queue Managers

You should know the process names of all your queue managers to troubleshoot queue manager problems as explained in Section 12.13.2.

The default queue manager name is SYS$QUEUE_MANAGER and the process name for the default queue manager process is QUEUE_MANAGE.

When you create an additional queue manager, you specify its name with the /NAME_OF_MANAGER qualifier. The process name for a queue manager other than the default is the first twelve characters of the queue manager name assigned with /NAME_OF_MANAGER qualifier.

For example, if you create a queue manager with the name PRINT_MANAGER, the process name is PRINT_MANAGE.

How Multiple Queue Managers Use the Queue Database

Multiple queue managers share a single queue database. However, a queue database for multiple queue managers contains more files than one for a single queue manager. For more information, see Section 12.2.

Commands for Managing Multiple Queue Managers

By default, the following commands affect the default queue manager SYS$QUEUE_MANAGER or the queues running on the default queue manager:

- START/QUEUE/MANAGER
- ENABLE AUTOSTART/QUEUES and DISABLE AUTOSTART/QUEUES
- STOP/QUEUES/ON_NODE
- STOP/QUEUE/MANAGER
- DELETE/QUEUE/MANAGER

The /NAME_OF_MANAGER qualifier allows you to use these commands with a different queue manager. For more information, see the *OpenVMS DCL Dictionary*.

12.2 Understanding the Queue Database

The queue database stores information used to keep the queuing system operating, including information about jobs, queues, and the queue manager. The queue database for a single default queue manager (SYS$QUEUE_MANAGER) is made up of the following files:

File	Description
Master file (QMAN$MASTER.DAT)	Contains: • The location of the queue and journal files • Definitions of forms and characteristics • A list of queue names • A list of nodes allowed to run the queue manager • A list of queue managers
Queue file (SYS$QUEUE_MANAGER.QMAN$QUEUES)	Contains the queue definitions you form when you create, start, or modify queues.

File	Description
Journal file (SYS$QUEUE_MANAGER.QMAN$JOURNAL)	Contains information allowing the queue manager to return to the last known state if: • A standalone machine is unexpectedly stopped • A VMScluster node that is running the queue manager leaves VMScluster environment The journal file also contains job definitions.

On systems with multiple queue managers, the queue database contains an additional queue file and journal file for each additional queue manager. Additional queue files are named in the format *name_of_manager*.QMAN$QUEUES. Additional journal files are named in the format *name_of_manager*.QMAN$JOURNAL.

SYS$COMMON:[SYSEXE] is the default location for all queue database files. However, you might store the files in another location for certain reasons. For more information, see Section 12.3.

12.3 Specifying the Location of Queue Database Files

If the queue database files are not located in the default location of SYS$COMMON:[SYSEXE], you must specify the location of the queue database. You would move the files from the default location for the following reasons:

- In a VMScluster environment with multiple system disks, the default location does not work. The files must be stored on a disk that is shared by all VMScluster member nodes.

  With this configuration, you must also make sure the disk holding the master file (QMAN$MASTER.DAT) is available to all nodes using the queue database *at the beginning of system startup*.

- In configurations without multiple system disks, to save space on the system disk or to help improve performance. To satisfy this reason, moving just the queue and journal files is sufficient.

How to Move the Queue Database Files

You can move the files in either of the following ways:

- When you create the queue database file the first time, create them in the alternate location:

  1. Define the logical name QMAN$MASTER in SYLOGICALS.COM (to specify where the master file is to be created).
  2. Start the queue manager with the START/QUEUE/MANAGER command, specifying the *dirspec* parameter (to specify where the queue and journal files are to be created).

  For detailed information, see Section 12.4.

- If you previously created the files in the default location:

  1. Stop the queue manager.
  2. Copy the files to the new location.
  3. Verify that the files have been successfully copied to the new location, and then delete the files in the old location.

4. If you move the master file, define the logical name QMAN$MASTER on all nodes (to specify where the master file is located), and add this logical name definition to SYLOGICALS.COM on all nodes.
5. Restart the queue manager with the START/QUEUE/MANAGER command, specifying the *dirspec* parameter (to specify where the queue and journal files are located).

Section 12.3.1 describes in detail how to specify the location of the master file. Section 12.3.2 describes in detail how to specify the location of the queue and journal files.

12.3.1 Master File

To specify the location of the master file, perform the following steps before starting the queue manager and creating the queue database:

1. Entering a command in the following format:

   ```
   DEFINE/SYSTEM/EXECUTIVE_MODE QMAN$MASTER location
   ```

   The *location* parameter specifies the device and directory where the master file is to be created.

   In a VMScluster environment, enter this command on every node in the VMScluster.

   Note

   In a VMScluster environment, the directory you specify for the master file must be available to all nodes in the cluster. If the directory specification is a concealed logical name, it must be identically defined on all nodes in the VMScluster, and the disk must be mounted on all VMScluster member nodes *early in system startup*.

2. Add the command you entered in step 1 to the SYLOGICALS.COM startup command procedure on every node in the VMScluster environment.
3. If the location you specify is on a disk other than the node's system disk, add a command to mount the disk in the command procedure SYS$MANAGER:SYLOGICALS.COM. Although SYLOGICALS is normally used to define logical names, it is important that the disk holding the master file be mounted in the SYLOGICALS startup procedure, so the master file is available before the job controller starts the queue manager.

   For information on mounting a disk in SYLOGICALS.COM, see the section Coordinating Shared System Files with Multiple Common System Disks in *VMScluster Systems for OpenVMS*.

12.3.2 Queue and Journal Files

Specify the location of the queue and journal files with the *dirspec* parameter to the START/QUEUE/MANAGER command.

Note

In a VMScluster environment, the directory you specify for the queue and journal files must be available to all nodes that can run the queue manager. If the directory specification is a concealed logical name, it must

be identically defined on all nodes in the VMScluster, and the disk must be mounted on all nodes capable of running the queue manager.

Once you enter the START/QUEUE/MANAGER command, the directory location you enter is stored in the queue database. If you must restart the queue manager, respecifying the directory location is not necessary.

12.4 Starting the Queue Manager and Creating the Queue Database

Before you can create any queues, you must create the queue database. Use the START/QUEUE/MANAGER command with the /NEW qualifier to start the queue manager and create the queue database. You normally perform this task only once.

Once you enter the START/QUEUE/MANAGER command, the command is stored, along with any qualifier or parameter you entered, in the queue database. Thereafter, the job controller automatically starts the queue manager during reboot unless a STOP/QUEUE/MANAGER/CLUSTER command has been entered. For this reason, including the START/QUEUE/MANAGER command in your startup command procedure is unnecessary.

Important

This section describes how to start the queue manager and create the queue database on systems and VMSclusters with a single system disk.

For systems in a VMScluster with multiple system disks (this includes any mixed-architecture VMScluster) you must perform special tasks. For instructions, see *VMScluster Systems for OpenVMS*.

How to Perform This Task

1. Make sure the values of the system parameters SCSNODE and SCSSYSTEMID match the DECnet for OpenVMS node name and node id. These values must be correctly defined for the queuing system to operate correctly.

   Use the PARAMETERS SHOW command of the System Management utility (SYSMAN) to determine the value of the system parameters SCSNODE and SCSSYSTEMID. Use the Network Control Program (NCP) SHOW EXECUTOR SUMMARY command to show the DECnet for OpenVMS node name and node id. For more specific instructions, see Section 12.13.5.

2. If you have only one system disk and you are creating the files in the default location, go to step 3. (The default location is SYS$COMMON:[SYSEXE].)

   If you want to create the queue database files in a location other than the default, do the following:

   a. Read the instructions in Section 12.3.

   b. Choose a location for the queue database files; the files must be stored on a disk that is shared by all VMScluster member nodes. You can keep the master file in a different location than the pairs of queue and journal files, but each pair of queue and journal files must reside together in the same directory. (Different pairs of queue and journal files can reside in different directories.)

c. If you want to move the master file, follow the instructions in Section 12.3.1 to define the location of the master file. However, with only one system disk, moving the master file is probably unnecessary; moving the queue and journal files should be sufficient.

d. Make sure the disk or disks that are to hold the queue database files are mounted.

3. Enter a command in the following format:

   START/QUEUE/MANAGER/NEW_VERSION[/ON=(node-list)] [dirspec]

   This command starts the queue manager process and, optionally, creates the queue database files, where:

/NEW_VERSION	Specifies that a new, empty queue database is to be created. Specify this *only* if you do not have a currently functioning database. If you specify this qualifer and you already have a queue database, the system will overwrite your current queue database.
/ON=*(node-list)*	Allows you to customize failover of the queue manager. For more information, see Section 12.5.
dirspec	Specifies the location of the queue and journal files, as explained in Section 12.3.2. Use this parameter if you are creating the queue and journal files in a location other than the default.

Caution

Specify the /NEW_VERSION qualifier *only* if you want to create a new queue database. If your queuing system is already functioning, creating a new queue database is not necessary.

If the queue manager does not start, see Section 12.13.1 for a troubleshooting checklist.

For more information on the START/QUEUE/MANAGER command, see the *OpenVMS DCL Dictionary*.

For information about creating an additional queue manager see Section 12.8.

Example

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE QMAN$MASTER DUA4:[MASTER]
$ MOUNT/SYSTEM/NOASSIST DUA4:
$ !
$ ! Add the two previous commands to SYLOGICALS.COM
$ !
$ START/QUEUE/MANAGER/NEW_VERSION DUA2:[SYSQUE]
```

In this example:

- The master file is located in the directory DUA4:[MASTER].
- The queue and journal files are located in the directory DUA2:[SYSQUE].

12.5 Customizing Queue Manager Failover

By default, all nodes in a VMScluster environment are able to run the queue manager, in no specified order of preference. The /ON qualifier of the START /QUEUE/MANAGER command lets you specify a list of VMScluster member nodes that can run the queue manager, in the preferred order in which they should claim the process. Digital recommends you specify the asterisk character (*) at the end of the node list to make sure that at least one node is always available to run the queue manager. For more information, see the /ON qualifier for the START/QUEUE/MANAGER command in the *OpenVMS DCL Dictionary*.

12.6 Stopping the Queue Manager

To stop the queue manager, enter the DCL command STOP/QUEUE/MANAGER /CLUSTER. This command shuts down the queue manager on a standalone node or VMScluster node. The queue manager performs the following tasks:

- Aborts all current jobs that cannot be restarted and requeues all current restartable jobs
- Stops all execution queues and disables autostart on all nodes
- Closes all queue database files associated with that queue manager

Once the STOP/QUEUE/MANAGER/CLUSTER command is entered, the queue manager process remains stopped, and requests to that queue manager are denied until you restart the queue manager by entering the START/QUEUE /MANAGER command as explained in Section 12.7. (Note that the queue system remains running as long as one or more queue managers are running.) VMScluster transitions do not change the state of the queue manager. Newly available nodes do not attempt to start the queue manager (unless the START /QUEUE/MANAGER command is entered).

The /CLUSTER qualifier is required to stop a clusterwide queue manager. If you enter the obsolete command STOP/QUEUE/MANAGER (without the /CLUSTER qualifier), the command performs the same function as the DCL command STOP /QUEUES/ON_NODE.

To stop all queues on a single node without stopping the queue manager, enter the STOP/QUEUES/ON_NODE command.

12.7 Restarting the Queue Manager

Section 12.4 explains how to start your queue manager the first time with the START/QUEUE/MANAGER/NEW_VERSION command. After you initially enter the START/QUEUE/MANAGER command with the /NEW_VERSION qualifier, you do not normally need to enter the START/QUEUE/MANAGER command again. The queue manager is automatically restarted, unless you enter the STOP/QUEUE/MANAGER/CLUSTER command.

However, after starting the queue manager initially, you might need to reenter the START/QUEUE/MANAGER command for one of the following reasons:

- If the STOP/QUEUE/MANAGER/CLUSTER command has been executed, enter the START/QUEUE/MANAGER command to restart the queue manager.
- In a VMScluster environment, enter the START/QUEUE/MANAGER command with the /ON qualifier to modify the list of preferred nodes on which the queue manager can run. For more information, see Section 12.5.

- In a VMScluster environment, enter the START/QUEUE/MANAGER command to ensure that the queue manager process is executing on the most preferred, available node. If the queue manager is not running on the most preferred, available node, it will be moved to that node without interruption of service. If you are using the default node list (*), the queue manager will not move. For more information, see the description of the /ON qualifier in the *OpenVMS DCL Dictionary*.

How to Perform This Task

To restart the queue manager, use the START/QUEUE/MANAGER command in the following format:

START/QUEUE/MANAGER[/ON=(node-list)] [dirspec]

If you restart the queue manager, you specify the /ON=(*node-list*) qualifier or the *dirspec* parameter only if you want to change the value you are currently using for the qualifier or parameter. The command you enter to start the queue manager is stored in the queue database, with any qualifier or parameter you specify. If you do not specify a qualifier or parameter, the queue manager is started using the node list and location (if any) stored in the queue database.

For more information about the START/QUEUE/MANAGER command, see the *OpenVMS DCL Dictionary*.

If the queue manager does not start, see Section 12.13.1 for a troubleshooting checklist.

12.8 Creating an Additional Queue Manager

To work around CPU, disk space, or memory limitations in a VMScluster environment, you can use multiple queue managers to distribute the batch and print work load between nodes and disk volumes. For more information, including restrictions for multiple queue managers, see Section 12.1.2.

How to Perform This Task

1. Follow the steps in Section 12.4. When you enter the START/QUEUE /MANAGER command, specify the /ADD and /NAME_OF_MANAGER qualifiers. For example:

   ```
   $ START/QUEUE/MANAGER/ADD/NAME_OF_MANAGER=BATCH_MANAGER
   ```

   Caution

   Do not specify the /NEW_VERSION qualifier when creating an additional queue manager. Multiple queue managers share a single queue database. Additional queue and journal files for the additional queue manager will automatically be created.

2. When you create a queue with the INITIALIZE command (as explained in Section 13.7.2), specify the name of the queue manager on which it is to run by including the /NAME_OF_MANAGER qualifier.

   If you do not specify the /NAME_OF_MANAGER qualifier, the queue will be created to run on the default queue manager, SYS$QUEUE_MANAGER.

   To move an existing queue from its original queue manager to a different queue manager, delete the queue with the DELETE/QUEUE command and re-create the queue with the INITIALIZE/QUEUE command.

3. When entering DCL commands to maintain the queue manager, be sure to specify the /NAME_OF_MANAGER command to specify the queue manager to which the command is to apply. If you do not specify the /NAME_OF_MANAGER qualifier, the command will be executed on the default queue manager, SYS$QUEUE_MANAGER. The SHOW QUEUE/MANAGER command allows you to display information about the queue managers running on your system.

Example

```
$ START/QUEUE/MANAGER/NEW_VERSION/NAME_OF_MANAGER=PRINT_MANAGER -
_$ /ON=(JADE,RUBY,*)
$ START/QUEUE/MANAGER/ADD/NAME_OF_MANAGER=BATCH_MANAGER -
_$ /ON=(OPAL,PEARL,*)
$ SHOW QUEUE/MANAGER/FULL
Master file:  SYS$COMMON:[SYSEXE]QMAN$MASTER.DAT;

Queue manager PRINT_MANAGER, running, on JADE::
  /ON=(JADE,RUBY,*)
  Database location:  SYS$COMMON:[SYSEXE]

Queue manager BATCH_MANAGER, running, on OPAL::
  /ON=(OPAL,PEARL,*)
  Database location:  SYS$COMMON:[SYSEXE]
```

12.9 Displaying Information About the Queue Manager

To obtain information about one or more queue managers, enter the SHOW QUEUE/MANAGER command.

Example

```
$ SHOW QUEUE/MANAGER
Queue manager PRINT_MANAGER, running, on NODEA::

Queue manager SYS$QUEUE_MANAGER, running, on NODED::

$ SHOW QUEUE/MANAGER/FULL
Master file:  SYS$SPECIFIC:[SYSEXE]QMAN$MASTER.DAT;

Queue manager PRINT_MANAGER, running, on NODEA::
  /ON=(NODEA,NODEB,*)
  Database location:  SYS$COMMON:[SYSEXE]

Queue manager SYS$QUEUE_MANAGER, running, on NODED::
  /ON=(NODEC,NODED,NODEE,*)
  Database location:  SYS$SYSROOT:[SYSEXE]
```

For more information, see the *OpenVMS DCL Dictionary*.

12.10 Saving the Queue Database

You should save a copy of the queue database each time you want to preserve changes to your queue configuration. If, for some reason, your queue database files are not accessible, you can restore the queue database so that you do not have to redefine forms and characteristics and reinitialize each queue.

To save a record-by-record copy of the queue database *while the queuing system is functioning*, perform the following steps. This procedure saves definitions of queues, forms, and characteristics. No job information is preserved. (Digital recommends not saving the journal file because timed and pending jobs might be reexecuted after the journal file is restored.)

How to Perform This Task

1. Enter an OpenVMS Convert utility (CONVERT) command in the following format to save the master file:

   CONVERT/SHARE QMAN$MASTER.DAT master-filename

   For *master-filename*, specify the name of the file to which QMAN$MASTER.DAT is to be copied.

   For more information about CONVERT see the *OpenVMS Record Management Utilities Reference Manual*.

2. Enter a CONVERT command in the following format to save the queue file:

   CONVERT/SHARE SYS$QUEUE_MANAGER.QMAN$QUEUES queue-filename

   For *queue-filename*, specify the name of the file to which SYS$QUEUE_ MANAGER.QMAN$QUEUES is to be copied.

3. Use the Backup utility (BACKUP) to save the files created with CONVERT. Use a command in the following format:

   BACKUP/LOG masterfile-name,queuefile-name device:saveset-name/LABEL=label

   For more information about the Backup utility, see the *OpenVMS System Management Utilities Reference Manual*.

Example

```
$ SET DEFAULT SYS$COMMON:[SYSEXE]
$ CONVERT/SHARE QMAN$MASTER.DAT MASTERFILE_9SEP.KEEP;
$ CONVERT/SHARE SYS$QUEUE_MANAGER.QMAN$QUEUES QFILE_9SEP.KEEP;
$ INITIALIZE MUA0: QDB
$ MOUNT/FOREIGN MUA0:
%MOUNT-I-MOUNTED, QDB mounted on _LILITH$MUA0:
$ BACKUP/LOG MASTERFILE_9SEP.KEEP,QFILE_9SEP.KEEP MUA0:QDB_9SEP.SAV/LABEL=QDB
%BACKUP-S-COPIED, copied SYS$COMMON:[SYSEXE]MASTERFILE_9SEP.KEEP;
%BACKUP-S-COPIED, copied SYS$COMMON:[SYSEXE]QFILE_9SEP.KEEP;
$ DISMOUNT MUA0:
```

12.11 Restoring the Queue Database

If, for some reason, you lose the queue database, you can restore your queue configuration from copies you saved as explained in Section 12.10. All queue, form, characteristic, and queue manager information will be restored. Information about jobs in the queues will not be restored.

How to Perform This Task

1. If the queue manager is running, stop it by entering the DCL command STOP /QUEUE/MANAGER/CLUSTER.

2. Delete all three queue database files. You must delete all three files, even if only one or two of them are lost.

3. Use the Backup utility (BACKUP) to restore the queue file and master file from the save set you created in step 3 of Section 12.10. If the master file or queue file is stored in a location other than the default, make sure you restore it to the correct location or specify the new location when starting the queue manager.

Note

When you restore your queue database, you must always restore both the master and queue file of the database, even if you lost only one of those files.

4. Start the queue manager with the START/QUEUE/MANAGER command as you would normally. Do not enter the /NEW_VERSION qualifier. A new (empty) journal file will automatically be created.

Example

```
$ STOP/QUEUE/MANAGER/CLUSTER
$ SET DEFAULT SYS$COMMON:[SYSEXE]
$ DELETE SYS$QUEUE_MANAGER.QMAN$JOURNAL;,SYS$QUEUE_MANAGER.QMAN$QUEUES;, -
_$ QMAN$MASTER.DAT;
$ MOUNT/FOREIGN MUA0:
%MOUNT-I-MOUNTED, QDB mounted on _LILITH$MUA0:
$ BACKUP/LOG MUA0:QDB_9SEP.SAV/SELECT=[SYSEXE]MASTERFILE_9SEP.KEEP; -
_$ QMAN$MASTER.DAT;
%BACKUP-S-CREATED, created SYS$COMMON:[SYSEXE]QMAN$MASTER.DAT;1
$ SET MAGTAPE/REWIND MUA0:
$ BACKUP/LOG MUA0:QDB_9SEP.SAV/SELECT=[SYSEXE]QFILE_9SEP.KEEP; -
_$ SYS$QUEUE_MANAGER.QMAN$QUEUES
%BACKUP-S-CREATED, created SYS$COMMON:[SYSEXE]SYS$QUEUE_MANAGER.QMAN$QUEUES;1
$ DISMOUNT MUA0:
$ START/QUEUE/MANAGER
```

12.12 Maximizing Queue System Performance

The following resources have the most effect on queuing system performance:

- CPUs—CPU speed and availability are important on the nodes on which the queue managers are running. The more time the queue manager has and the faster the CPU, the faster the queue manager can process information.
- Disks—For disks, rotational speed is more important than seek time because most access to the disk is for writing to the sequential journal file. In addition to disk speed, nonqueuing activity (such as paging and heavy access to the database) on the database disk can have a great affect on performance.

You can use to the following methods to maximize your queuing system's performance:

- Maintain queue manager processes on the fastest and most available nodes in a cluster. (See Section 12.1.2.)
- Create an addition queue manager to distribute the CPU load on heavily used queuing systems. (See Section 12.8.)
- Move queue and journal files to disks with less activity. (See Section 12.4.)

  Note that moving the master file provides little difference in performance.

12.13 Solving Queue Manager Problems

Use the following sections to help solve queue manager problems.

Topic	For More Information
Avoiding common problems: a troubleshooting checklist	Section 12.13.1
If the queue manager does not start	Section 12.13.2
If the queuing system stops or the queue manager does not run on certain nodes	Section 12.13.3
If the queue manager becomes unavailable	Section 12.13.4
If the queuing system does not work on a certain VMScluster node	Section 12.13.5
If you see inconsistent queuing behavior on different VMScluster nodes	Section 12.13.6
Reporting a queuing system problem to Digital	Section 12.14

12.13.1 Avoiding Common Problems: A Troubleshooting Checklist

To avoid the most common queuing system problems, make sure you have met the following requirements:

Requirement	For More Information
QMAN$MASTER (if defined) is correctly defined on all nodes in the cluster.	Section 12.3
The queue database is in the specified location.	Section 12.3
The queue database disk is mounted and available.	Section 12.3
The system parameters SCSNODE and SCSSYSTEMID match the DECnet for OpenVMS node name and node id.	Section 12.13.5
The node list specified /ON qualifier contains a sufficient number of nodes. If you specify a node list, Digital recommends you include an asterisk (*) at the end of the node list.	Section 12.13.4

12.13.2 If the Queue Manager Does Not Start

If the queue manager does not start when you enter the START/QUEUE /MANAGER command, the system displays the following message:

```
%JBC-E-QMANNOTSTARTED, queue manager could not be started
```

12.13.2.1 Investigating the Problem

Search the operator log file SYS$MANAGER:OPERATOR.LOG (or look on the operator console) for messages from the queue manager and job controller for information about the problem, as follows:

```
$ SEARCH SYS$MANAGER:OPERATOR.LOG/WINDOW=5 QUEUE_MANAGE,JOB_CONTROL
```

Use the information provided with these messages to further investigate the problem, making sure you have met the requirements listed in Section 12.13.1. For example, the following message indicates that the master queue file does not exist in the expected location:

```
%%%%%%%%%%  OPCOM  13-MAR-1995 15:53:52.84  %%%%%%%%%%
Message from user JOB_CONTROL on ABDCEF
%JBC-E-OPENERR, error opening SYS$COMMON:[SYSEXE]QMAN$MASTER.DAT

%%%%%%%%%%  OPCOM  13-MAR-1995 15:53:53.04  %%%%%%%%%%
Message from user JOB_CONTROL on ABDCEF
-SYSTEM-W-NOSUCHFILE, no such file
```

On systems with multiple queue managers, also search for messages displayed by additional queue managers by including their process names in the search string. To display information about queue managers running on your system, use the SHOW QUEUE/MANAGERS command as explained in Section 12.9. For more information about multiple queue managers and their process names, see Section 12.1.2.

Example

```
$ START/QUEUE/MANAGER DUA55:[SYSQUE] ❶
%JBC-E-QMANNOTSTARTED, queue manager could not be started ❷
$ SEARCH SYS$MANAGER:OPERATOR.LOG /WINDOW=5 QUEUE_MANAGE,JOB_CONTROL ❸
%%%%%%%%%%  OPCOM  14-APR-1995 18:55:18.23  %%%%%%%%%%
Message from user QUEUE_MANAGE on CATNIP
%QMAN-E-OPENERR, error opening DUA55:[SYSQUE]SYS$QUEUE_MANAGER.QMAN$QUEUES;

%%%%%%%%%%  OPCOM  14-APR-1995 18:55:18.29  %%%%%%%%%%
Message from user QUEUE_MANAGE on CATNIP
-RMS-F-DEV, error in device name or inappropriate device type for operation

%%%%%%%%%%  OPCOM  14-APR-1995 18:55:18.31  %%%%%%%%%%
Message from user QUEUE_MANAGE on CATNIP
-SYSTEM-W-NOSUCHDEV, no such device available ❹
$ START/QUEUE/MANAGER DUA5:[SYSQUE] ❺
```

❶ This command attempts to start the queue manager specifying DUA55:[SYSQUE] as the location of the queue and journal files.

❷ The error message indicates that the queue manager did not start.

❸ This command searches the operator log file for relevant messages.

❹ One of the messages reveals that the queue file could not be opened because device DUA55: does not exist.

❺ This command successfully starts the queue manager, specifying DUA5:[SYSQUE] as the correct location for the queue and journal files.

12.13.3 If the Queuing System Stops or the Queue Manager Does Not Run on Certain Nodes

Use this section if the queue manager does not run on a certain node in the cluster, or if the queuing system stops, especially after one of the following:

- The node on which the queue manager was running leaves the cluster.
- A new node boots into the cluster.
- You change the node list specified with the /ON qualifier of the START /QUEUE/MANAGER command.
- You start the queue manager after moving the queue database.

12.13.3.1 Investigating the Problem

Check the operator log that was current at the time the queue manager started up or failed over. Search the log for operator messages from the queue manager (QUEUE_MANAGE process name).

On systems with multiple queue managers, also search for messages displayed by additional queue managers, by including their process names in the search string. To display information about queue managers running on your system, use the SHOW QUEUE/MANAGERS command as explained in Section 12.9. For more information about multiple queue managers and their process names, see Section 12.1.2.

The following messages indicate that the queue database is not in the specified location:

```
%%%%%%%%%%  OPCOM   4-FEB-1995 15:06:25.21  %%%%%%%%%%
Message from user QUEUE_MANAGE on MANGLR
%QMAN-E-OPENERR, error opening CLU$COMMON:[SYSEXE]SYS$QUEUE_MANAGER.QMAN$QUEUES;

%%%%%%%%%%  OPCOM   4-FEB-1995 15:06:27.29  %%%%%%%%%%
Message from user QUEUE_MANAGE on MANGLR
-RMS-E-FNF, file not found

%%%%%%%%%%  OPCOM   4-FEB-1995 15:06:27.45  %%%%%%%%%%
Message from user QUEUE_MANAGE on MANGLR
-SYSTEM-W-NOSUCHFILE, no such file
```

The following messages indicate that the queue database disk is not mounted:

```
%%%%%%%%%%  OPCOM   4-FEB-1995 15:36:49.15  %%%%%%%%%%
Message from user QUEUE_MANAGE on MANGLR
%QMAN-E-OPENERR, error opening DISK888:[QUEUE_DATABASE]SYS$QUEUE_MANAGER.QMAN$QUEUES;

%%%%%%%%%%  OPCOM   4-FEB-1995 15:36:51.69  %%%%%%%%%%
Message from user QUEUE_MANAGE on MANGLR
-RMS-F-DEV, error in device name or inappropriate device type for operation

%%%%%%%%%%  OPCOM   4-FEB-1995 15:36:52.20  %%%%%%%%%%
Message from user QUEUE_MANAGE on MANGLR
-SYSTEM-W-NOSUCHDEV, no such device available
```

12.13.3.2 Cause

The queuing system does not work correctly under the following circumstances:

- If the *dirspec* parameter specified with the START/QUEUE/MANAGER command (specifying the location of the queue and journal files) is not translated exactly the same on all nodes, and the queue manager starts on one of the affected nodes. You typically find this problem in a VMScluster environment when you add an additional system disk or move the queue database.
- If the queue database disk is not mounted for the node on which the queue manager attempts to run.

In general, the queuing system will be shut off completely if the queue manager encounters a serious error and forces a crash or failover twice in two minutes consecutively on the same node. Therefore, the queuing system may have stopped, or it may continue to run if the queue manager moves to yet another node on which it can access the database after the original failed startup.

12.13.3.3 Correcting the Problem

Perform the following steps:

1. If the queue manager is stopped, enter the START/QUEUE/MANAGER command and include the following:
   - An appropriate list of nodes with the /ON qualifier.
   - The appropriate *dirspec* parameter (to specify the location of the queue and journal files). All the nodes included in the node list with the /ON qualifier must be able to access this directory.
2. On all nodes specified in the node list, (except on any nodes that boot from the disk where the queue database files are stored), add a MOUNT command to the SYLOGICALS.COM procedure to mount the disk that holds the master file. You do not need to explicitly mount the disk on a node where it is the system disk.

12.13.4 If the Queue Manager Becomes Unavailable

If the queue manager will not start or has stopped running, an insufficient failover node list might have been specified for the queue manager, such that none of the nodes in the failover list is available to run the queue manager.

12.13.4.1 Correcting the Problem

Make sure the queue manager node list contains a sufficient number of nodes by entering the START/QUEUE/MANAGER command and specifying a node list appropriate for your configuration with the /ON qualifier. For more information, see the START/QUEUE/MANAGER command in the *OpenVMS DCL Dictionary*.

If you are in doubt about what nodes to specify, Digital recommends you specify an asterisk (*) wildcard character as the last node in the list to specify that any remaining node in the VMScluster can run the queue manager. Specifying the asterisk wildcard prevents your queue manager from becoming unavailable because of an insufficient node list.

12.13.5 If the Queuing System Does Not Work on a Certain VMScluster Node

Use this section if the queuing system does not work on a certain node when it starts up.

12.13.5.1 Investigating the Problem

Perform the following steps:

1. Search the operator log that was current when the problem existed for the following messages; these messages are broadcast every 30 seconds after the affected node boots:

```
%%%%%%%%%%  OPCOM   4-FEB-1995 15:36:49.15  %%%%%%%%%%
Message from user QUEUE_MANAGE on ZNFNDL
%QMAN-E-COMMERROR, unexpected error #5 in communicating with node CSID 000000

%%%%%%%%%%  OPCOM   4-FEB-1995 15:36:49.15  %%%%%%%%%%
Message from user QUEUE_MANAGE on ZNFNDL
-SYSTEM-F-WRONGACP, wrong ACP for device_
```

2. Compare the node's value for the system parameters SCSNODE and SCSSYSTEMID with the values for the DECnet for OpenVMS node name and node id, as follows:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAMETERS SHOW SCSSYSTEMID

Parameter Name            Current    Default     Min.     Max.     Unit  Dynamic
--------------            -------    -------    -------  -------   ----  -------
SCSSYSTEMID                 19941          0        -1        -1 Pure-numbe

SYSMAN> PARAMETERS SHOW SCSNODE

Parameter Name            Current    Default     Min.     Max.     Unit  Dynamic
--------------            -------    -------    -------  -------   ----  -------
SCSNODE                  "RANDY  "    "    "    "    "    "    "ZZZZ" Ascii

SYSMAN> EXIT
$ RUN SYS$SYSTEM:NCP
NCP> SHOW EXECUTOR SUMMARY


Node Volatile Summary as of  5-FEB-1995 15:50:36

Executor node = 19.45 (DREAMR)

State                    = on
Identification           = DECnet for OpenVMS VAX V7.0


NCP> EXIT
$ WRITE SYS$OUTPUT 19*1024+45
19501
```

12.13.5.2 Cause

If the DECnet for OpenVMS node name and node id do not match the SCSNODE and SCSSYSTEMID system parameters, IPC (interprocess communication, an operating system internal mechanism) cannot work properly and the affected node will not be able to participate in the queuing system.

12.13.5.3 Correcting the Problem

Perform the following steps:

1. Modify the system parameters SCSNODE and SCSSYSTEMID or modify the DECnet for OpenVMS node name and node id, so the values match.

   For more information on these system parameters, see the *OpenVMS System Management Utilities Reference Manual*. For more information on the DECnet for OpenVMS node name and node id, see the *DECnet for OpenVMS Guide to Networking*.

2. Reboot the system.

12.13.6 If You See Inconsistent Queuing Behavior on Different VMScluster Nodes

Use this section if you see the following symptoms:

- After submitting a print job, you can display the job with a SHOW ENTRY command on the same node, but not on other nodes in the VMScluster environment.
- After defining or modifying a queue, the changes appear in a SHOW QUEUE display on some nodes, but not on others.

- You can successfully submit or print a job on some nodes but on other nodes, you receive a JOBQUEDIS error.

12.13.6.1 Investigating the Problem

Perform the following steps:

1. Use the SHOW LOGICAL command to translate the QMAN$MASTER logical name within the environment of each node in the VMScluster. If there is no translation on any given node, then translate the default value of SYS$COMMON:[SYSEXE].

   If the SHOW LOGICAL translations show a different physical disk name on one or more nodes, you have identified the problem.

2. Check the operator log files that were current at the time that one of the affected nodes booted. Search for an OPCOM message similar to the following from the process JOB_CONTROL:

```
%%%%%%%%%%  OPCOM   4-FEB-1995 14:41:20.88  %%%%%%%%%%
Message from user JOB_CONTROL on MANGLR
%JBC-E-OPENERR, error opening BOGUS:[QUEUE_DIR]QMAN$MASTER.DAT;

%%%%%%%%%%  OPCOM   4-FEB-1995 14:41:21.12  %%%%%%%%%%
Message from user JOB_CONTROL on MANGLR
-RMS-E-FNF, file not found
```

12.13.6.2 Cause

This problem may be caused by different definitions for the logical name QMAN$MASTER on different nodes in the cluster, causing multiple queuing environments. You typically find this problem in VMScluster environments when you have just added an additional system disk or moved the queuing database.

12.13.6.3 Correcting the Problem

Perform the following steps:

1. If only one queue manager and queue database exist, skip to Step 2.

   If more than one queue manager and database exist, perform the following steps:

   a. Enter STOP/QUEUE/MANAGER/CLUSTER on one of the nodes where the QMAN$MASTER logical name is incorrectly defined to stop the invalid queue manager.

   b. Delete all three files for the invalid queue database. (On systems with multiple queue managers, you might have more than three invalid files.)

2. Reassign the logical name QMAN$MASTER on the affected systems and correct the definition in the startup procedure where the logical name is defined (usually SYLOGICALS.COM.)

3. Enter a STOP/QUEUE/MANAGER/CLUSTER command on an unaffected node to stop the valid queue manager.

4. Enter START/QUEUE/MANAGER on any node and verify that the queuing system is working properly.

12.14 Reporting a Queuing System Problem to Digital

If you encounter problems with the queuing system that you need to report to Digital, provide the information in the following table. This information will help Digital representatives to diagnose your problem. Please provide as much of the information as possible.

Information	Description
Summary of the problem	Include the following: • The environment in which the problem occurred. For example, does the problem occur only on certain nodes, from certain user accounts, or when using certain layered products? • How this problem affects your operations. What site operations are being affected (for example, printing checks or submitting crucial batch jobs)? How often does the problem occur (for example, one printout per month, several printouts per day)? • What events occurred on the system between the time the queuing system operated correctly and the time the problem appeared. • Any workarounds you are currently using.
Steps for reproducing the problem	Specify the exact steps and include a list of any special hardware or software required to reproduce the problem.
Configuration information	For example: • Is the configuration a VMScluster system, and does it have multiple system disks? • Do you intend the queue database to be located in the default location (SYS$COMMON:[SYSEXE])? Do you intend the master file to be included in a different location than the queue and journal files?

Output from the SHOW QUEUE /MANAGER/FULL command	Use SYSMAN to enter the command on all nodes, as follows:
	`$ RUN SYS$SYSTEM:SYSMAN`
	`SYSMAN> SET ENVIRONMENT/CLUSTER`
	`SYSMAN> DO/OUTPUT SHOW QUEUE/MANAGERS/FULL`
	`SYSMAN> EXIT`
	`$ TYPE SYSMAN.LIS`
	Type the output file SYSMAN.LIS to verify that the output for all nodes match.

Location of the queue and journal files	If possible, find out the most recent value that was specified in the *dirspec* parameter of the START/QUEUE/MANAGER command (to specify the location of the queue and journal files). If none was specified, the default is SYS$COMMON:[SYSEXE].

Translation of QMAN$MASTER logical name	Verify that the translation is the same on all nodes.
	Enter the following commands, and include the resulting output:
	`$ RUN SYS$SYSTEM:SYSMAN`
	`SYSMAN> SET ENVIRONMENT/CLUSTER`
	`SYSMAN> DO SHOW LOGICAL QMAN$MASTER`
	If the translations returned from the SHOW LOGICAL command are not physical disk names, repeat the SHOW LOGICAL command within the environment of each node to translate the returned value until you reach a translation that includes the physical device name.

Information	Description

Operator log file output	Enter the following commands to search the operator log for any message output by the job controller or queue manager:
	`$ SEARCH SYS$MANAGER:OPERATOR.LOG/WINDOW=5 -`
	`_$ JOB_CONTROL,QUEUE_MANAGE`
	On systems with multiple queue managers, for queue managers other than the default, specify the first 12 characters of the queue manager name of any additional queue manager. For example, for a queue manager named PRINT_MANAGER, specify PRINT_MANAGE as follows:
	`$ SEARCH SYS$MANAGER:OPERATOR.LOG/WINDOW=5 -`
	`_$ JOB_CONTROL,QUEUE_MANAGE,PRINT_MANAGE`

Information returned from relevant DCL commands	Include this information if entering a DCL command shows evidence of the problem.
A copy of the journal file of the queue database	Use the Backup utility (BACKUP) with the /IGNORE=INTERLOCK qualifier to create a copy of the file SYS$QUEUE_ MANAGER.QMAN$JOURNAL, and provide this copy to Digital. On systems with multiple queue managers, include copies of journal files for all queue managers. Journal files for queue managers other than the default are named in the format *name_of_ manager*.QMAN$JOURNAL.

Copies of any process dumps that might have been created	Perform the following steps to find any related process dumps, and provide copies of the files to Digital:
	`$ RUN SYS$SYSTEM:SYSMAN`
	`SYSMAN> SET ENVIRONMENT/CLUSTER`
	`SYSMAN> DO DIRECTORY/DATE SYS$SPECIFIC:[SYSEXE]JBC$*.DMP, -`
	`_SYSMAN> QMAN$*.DMP,PRTSMB.DMP,LATSYM.DMP`
	If the problem involves an execution queue using a symbiont other than PRTSMB or LATSYM, also include process dump files from the symbiont. The file name has the format *image_file_name*.DMP.

Output from the SHOW QUEUE command	If your problem affects individual queues, enter the SHOW QUEUE command to show each affected queue.
Any other relevant information	For example: • When was the queue database last created or modified? Was it created or modified since the last reboot of the node or nodes? • Does the IPCACP process exist on the affected nodes? If not, try to determine whether the process existed earlier. For example, check the system accounting records.

13
Managing Batch and Print Queues

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

Before working with queues, you must start the queue manager and create the queue database. Section 12.4 describes this task.

Information Provided in This Chapter

This chapter describes the following tasks:

Task	Section
Managing queues on small systems	Section 13.1
Designing your batch queue environment	Section 13.5
Designing your output queue environment	Section 13.6
Setting up and starting queues	Section 13.7
Using queue options	Section 13.8
Managing queues	Section 13.9.1
Managing characteristics	Section 13.9.2
Managing forms	Section 13.9.4
Managing device control libraries	Section 13.9.5
Managing jobs	Section 13.9.6
Solving queue problems	Section 13.10

This chapter explains the following concepts:

Concept	Section
Queuing process	Section 13.2
Types of queues	Section 13.3
Autostart feature	Section 13.4
Options for controlling access to queues	Section 13.8.1
Job retention	Section 13.8.2
Queue characteristics	Section 13.8.3
Batch processing options	Section 13.8.5
Job scheduling options	Section 13.8.6
Banner pages	Section 13.8.7
Forms and stock	Section 13.8.9

Concept	Section
Device control libraries	Section 13.8.13

13.1 Managing Queues on Small Systems

Many of the features available for queues are not required on small systems with minimal queuing needs (for example, on workstations). If you are managing a small system, you probably need only the information in the following sections:

Topic	Section
Simple batch queue configuration	Section 13.5.1
Simple output queue configuration	Section 13.6.1
Setting up and starting queues	Section 13.7
Choosing and specifying queue options	Section 13.8
Managing queues	Section 13.9.1
Managing jobs in queues	Section 13.9.6

13.2 Understanding the Queuing Process

When a user submits a batch or print job, either by entering the DCL command SUBMIT or PRINT or through an application, the job is sent to a queue for processing. Information about the user's queue request, including the type of job, the file name or names, the name of the queue, and any special options, is sent to the queue manager. The queue manager stores and retrieves appropriate information from the queue database to print or execute the job. For more information on the queue manager and queue database, see Chapter 12.

The queue manager places the job in the appropriate queue to await its turn for processing. Only one print job can be printed on a printer at a single time; however, more than one batch job can execute simultaneously in a batch queue.

Print jobs are sent to an independent process, called a **symbiont**, for formatting; the symbiont sends the formatted job to the appropriate printer. For batch jobs, the job controller creates a process to perform the tasks in the batch job.

13.3 Understanding Types of Queues

Batch and print queues can be divided into two general classes:

Class	Description
Execution queues	Queues that accept batch or print jobs for processing.
Generic queues	Queues that hold jobs until they are transferred to an execution queue for processing. A generic queue holds a job until an appropriate execution queue becomes available to initiate the job. The queue manager then requeues the job to the available execution queue.

An execution and generic queue can be classified as either a batch or output queue. The following sections provide more details about execution and generic queues.

13.3.1 Batch Execution Queues

A batch execution queue can accept only batch jobs. A batch job executes as a detached process that sequentially runs one or more command procedures. The user defines the list of command procedures when submitting the job.

13.3.2 Output Execution Queues

An output execution queue accepts jobs for processing by a symbiont. The queue manager sends the symbiont a list of files, which the user defines when submitting the job. An output symbiont transfers data from a disk to an output device. As the symbiont processes each file, it produces output for the device it controls, such as a printer or a terminal.

The standard print symbiont provided by the operating system is named PRTSMB and is designed to print files on hardcopy devices. The LAT print symbiont LATSYM is used to print files on output devices attached to LAT ports. User-modified or user-written symbionts can also be designed for this or any other file processing activity managed by the OpenVMS batch and print queuing system. (See the *OpenVMS Utility Routines Manual* for more information.) Output execution queues include the following types:

Type	Description
Printer queue	Uses a symbiont to direct output to a printer.
Terminal queue	Uses a symbiont to direct output to a terminal printer.
Server queue	Uses a user-modified or user-written symbiont to process the files that belong to jobs in the queue.

When you create an output execution queue, you designate it as either a printer, a terminal, or a server execution queue; you can also specify the symbiont to be associated with the queue. However, when the queue is started, the symbiont process associated with the queue can override the queue designation if the queue type specified does not match the type of device. The standard symbiont provided with the operating system determines whether it is controlling a printer or a terminal. It communicates this information to the queue manager and, if necessary, the queue manager changes the type designation of the output execution queue. By convention, a user-written or user-modified symbiont that does not deliver output to a printer defines its queue as a server queue.

13.3.3 Generic Batch Queues

A **generic batch queue** can direct jobs only to batch execution queues. Generic batch queues are typically used in VMScluster environments to distribute the batch work load across several nodes (see Section 13.5.3). Generic batch queues are not automatically stopped when a node shuts down and therefore, do not need to be started when a node reboots.

13.3.4 Generic Output Queues

A **generic output queue** can direct jobs to any of the three types of output execution queues: print, terminal, or server. Generic output queues are typically used to distribute the output work load among several like printers (see Section 13.6.5).

Generic output queues are not automatically stopped when a node shuts down and therefore, do not need to be started when a node reboots.

13.3.4.1 Logical Queues

A **logical queue** is a special type of generic output queue that transfers jobs to another output execution queue. You might use this kind of queue to temporarily redirect a queue when the device on which it runs is broken.

A logical queue transfers its jobs into the execution queue specified with the ASSIGN/QUEUE command. For information on setting up a logical queue, see Section 13.9.1.14.

13.4 Understanding the Autostart Feature

Digital recommends you use autostart queues whenever possible. The **autostart** feature simplifies startup and ensures high availability of execution queues. It lets you do the following:

- Start all autostart queues on a node with a single command
- Specify a list of nodes (within a VMScluster environment) to which a queue can automatically attempt to fail over if necessary

  Autostart's failover feature is particularly useful on LAT queues. Because LAT printers are usually shared among users of multiple systems or VMScluster systems, many users will be affected if a LAT queue is unavailable. For highest availability, set up your LAT queues with a list of nodes to which the queue can fail over, so the queue will continue to run even if a node becomes unavailable.

To use autostart queues, you must do the following:

Task	For More Information
Enable autostart on a node. You can do this before or after you create a queue.	Section 13.9.1.3
Create the queue as an autostart queue and, optionally, specify a failover list.	Section 13.7.2.2
Activate the queue for autostart. You can do this when you create the queue, or after.	Section 13.9.1.4

When you enable autostart on a node, all stopped, active autostart queues capable of running on the node are automatically started by the queue manager. Any autostart queue that fails over to the node is also automatically started.

13.5 Designing Your Batch Queue Environment

Use the following sample configurations to design your batch processing environment. Your configuration may combine elements from several of these examples.

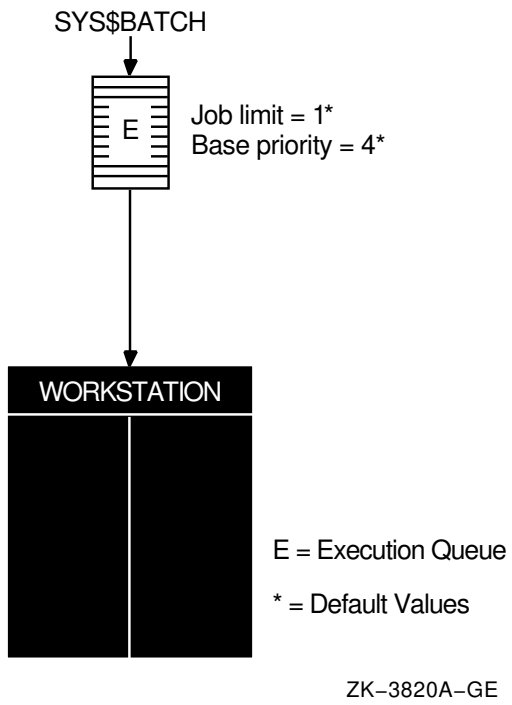
Configuration	For More Information
A single queue for limited batch processing	Section 13.5.1
Multiple queues for heavy batch processing, or customized queues for specialized batch processing	Section 13.5.2
A VMScluster environment	Section 13.5.3
Queues that must be highly available	Section 13.4

This section provides examples of some typical batch configurations. Use these sample configurations to design your batch environment. Your configuration will probably combine elements from several of these examples.

13.5.1 Simple Batch Queue Configuration: For Limited Batch Needs

Figure 13–1 illustrates a single, default batch queue. This simple configuration is suitable for a standalone system supporting mainly interactive processing.

Figure 13–1 Default Batch Queue



ZK–3820A–GE

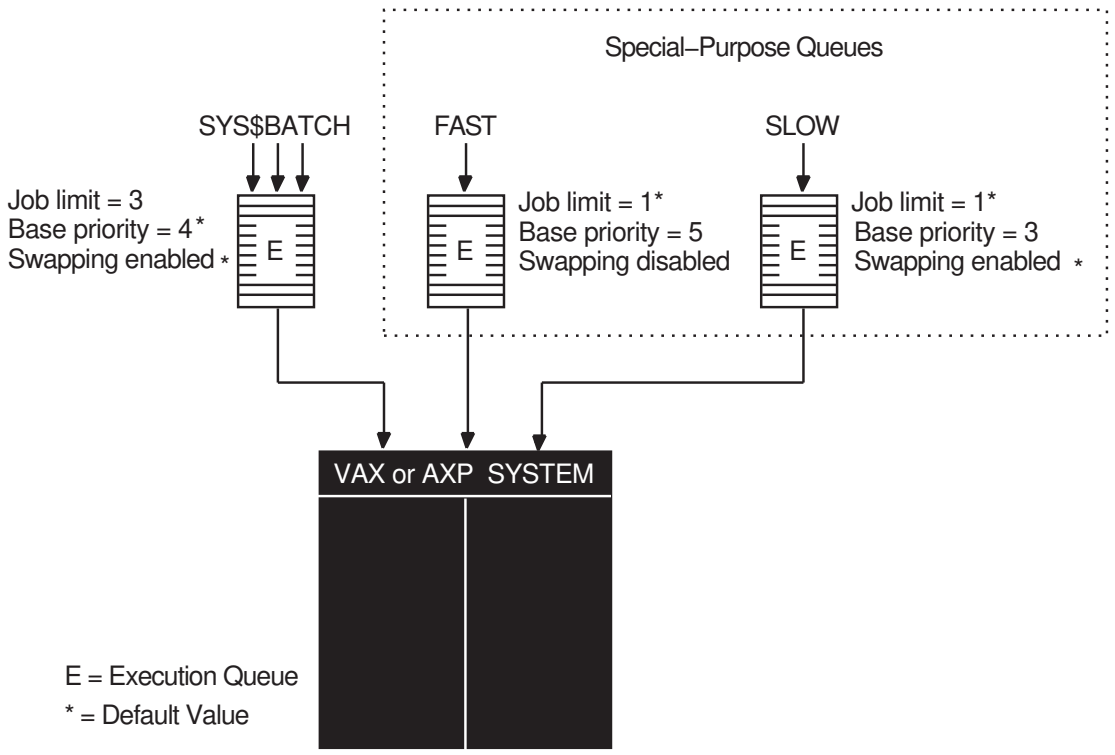
By default, when a user submits a batch job with the SUBMIT command, the job is placed in the queue named SYS$BATCH. To set up a single default queue on a standalone system, name the queue SYS$BATCH.

13.5.2 Specialized Batch Queues: Meeting Special Needs

If your users rely heavily on batch processing or have special processing needs, you might want to set up more than one queue. You can customize batch queues to handle specialized jobs by specifying performance and resource options for jobs in the queue. For more information, see Section 13.8.5.

The configuration illustrated in Figure 13–2 has several queues, each customized to process certain types of batch jobs.

Figure 13–2 Multiple Batch Queues with Special Resource and Performance Options



ZK–3819A–GE

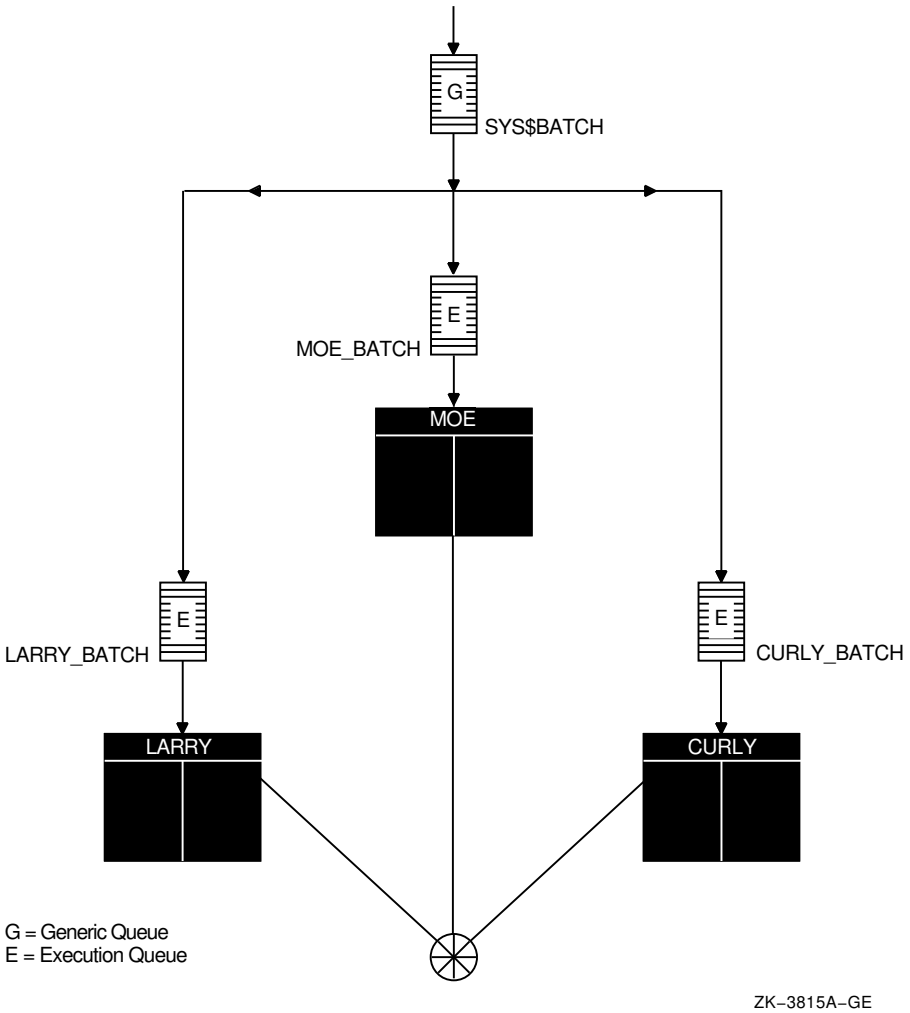
In this example, SYS$BATCH is the default queue. Normal batch jobs of no special importance would typically be submitted to this queue. FAST is a queue for executing high-priority jobs that should not be swapped out of memory. SLOW is a background queue for processing low-priority jobs. Typically, these are large jobs with large requirements for physical memory. Be conservative when changing base priority and swapping on a queue. Even a slight change can have a significant negative effect on batch and interactive performance. For example, even an increase of 1 in a queue's base priority can affect performance significantly.

For information about specifying these options for a batch queue, see Section 13.8.5.

13.5.3 Generic Batch Queues in a VMScluster: Distributing the Work Load

Generic queues can be used in a VMScluster environment to balance the use of processing resources by distributing batch processing across nodes in the VMScluster. (For an explanation of generic queues, see Section 13.3.) Figure 13–3 illustrates a typical configuration.

Figure 13–3 Batch Queue Configuration with Clusterwide Generic Queue



In this example, a generic batch queue named SYS$BATCH is set up to feed jobs to execution queues on each node in the VMScluster. When a user submits a job to the clusterwide generic queue SYS$BATCH, the job is placed in the appropriate execution queue to minimize the ratio of executing jobs to job limits for all execution queues fed by SYS$BATCH. For example, suppose execution queues MOE_BATCH, LARRY_BATCH, and CURLY_BATCH all have a job limit of 5. If MOE_BATCH and LARRY_BATCH are executing four jobs and CURLY_BATCH is executing one job, the generic queue SYS$BATCH will feed the next job to CURLY_BATCH.

See *VMScluster Systems for OpenVMS* for more information about VMScluster queue configurations. For information about how to create a generic queue, see Section 13.7.2.3.

13.6 Designing Your Output Queue Environment

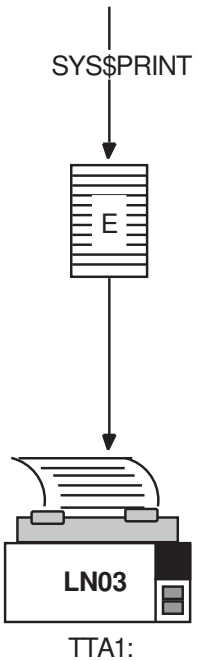
Use the following sample configurations to design your output environment. Your configuration will probably combine elements from several of these examples.

Configuration	For More Information
Queues that must be highly available	Section 13.4
A single print queue for limited printing	Section 13.6.1
Printers of different types	Section 13.6.2
PostScript printing	Section 13.6.3
Access to printers from multiple systems	Section 13.6.4
Multiple printers of the same type	Section 13.6.5
A VMScluster environment	Section 13.6.6
Applications that print output by writing directly to a printer rather than submitting to an output queue	Section 13.6.7
Distributed printing	Section 13.6.8

13.6.1 Simple Output Queue Configuration: For Limited Printing Needs

Figure 13–4 illustrates a very simple queue configuration. This configuration is appropriate for a standalone system supporting a single printer.

Figure 13–4 Simple Output Queue



E = Execution Queue

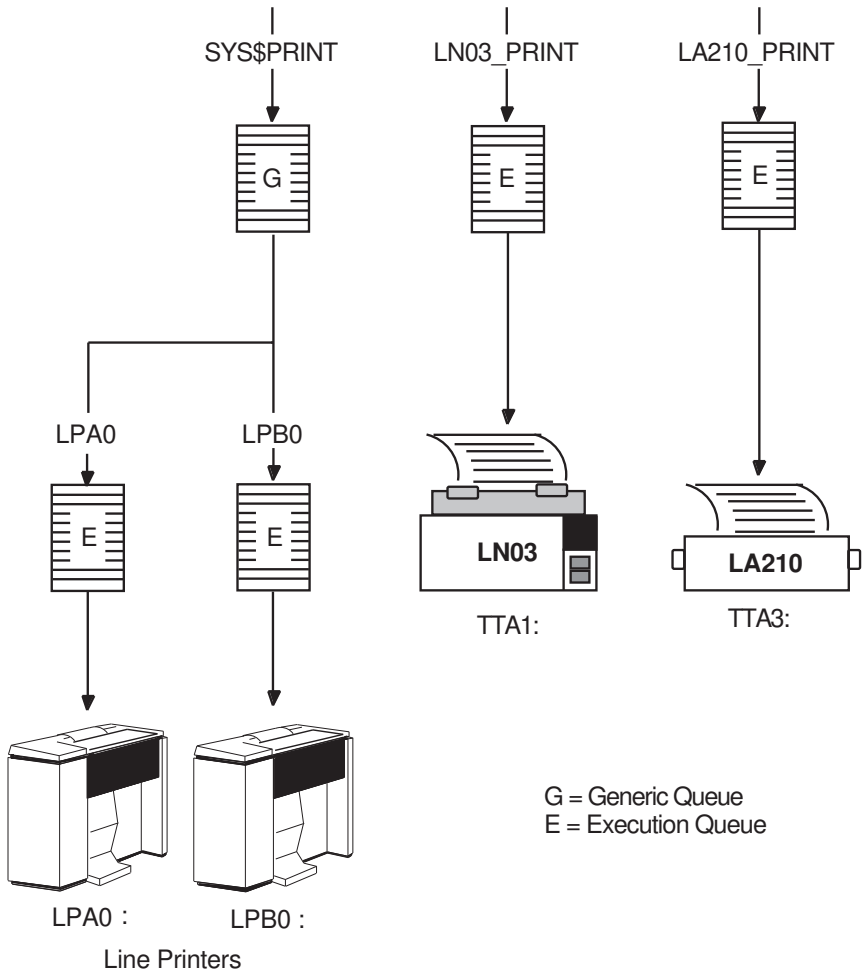
ZK–3816A–GE

By default, when a user submits a print job with the PRINT command, the job is placed in the queue named SYS$PRINT. To set up a single default printer queue on a standalone system, name the queue SYS$PRINT.

13.6.2 Mixed Printers: Specialized Output Execution Queues

If you have several different kinds printers (for example, an LN03 printer, an LA210 printer, and an LP27 line printer), you must set up a separate queue for each printer. The options, such as the default form or device control library, that you use with these queues will probably differ according to the printer to which the queue's output is sent. For example, the default form for a line printer might have a width of 132 columns, while the default form for an LN03 printer might have a width of 80 columns. Figure 13–5 illustrates such a configuration.

Figure 13–5 Queue Configuration with Mixed Printers



13.6.3 PostScript Printing

The operating system does not include software to support PostScript printing. In order to print PostScript files, you must have either of the following:

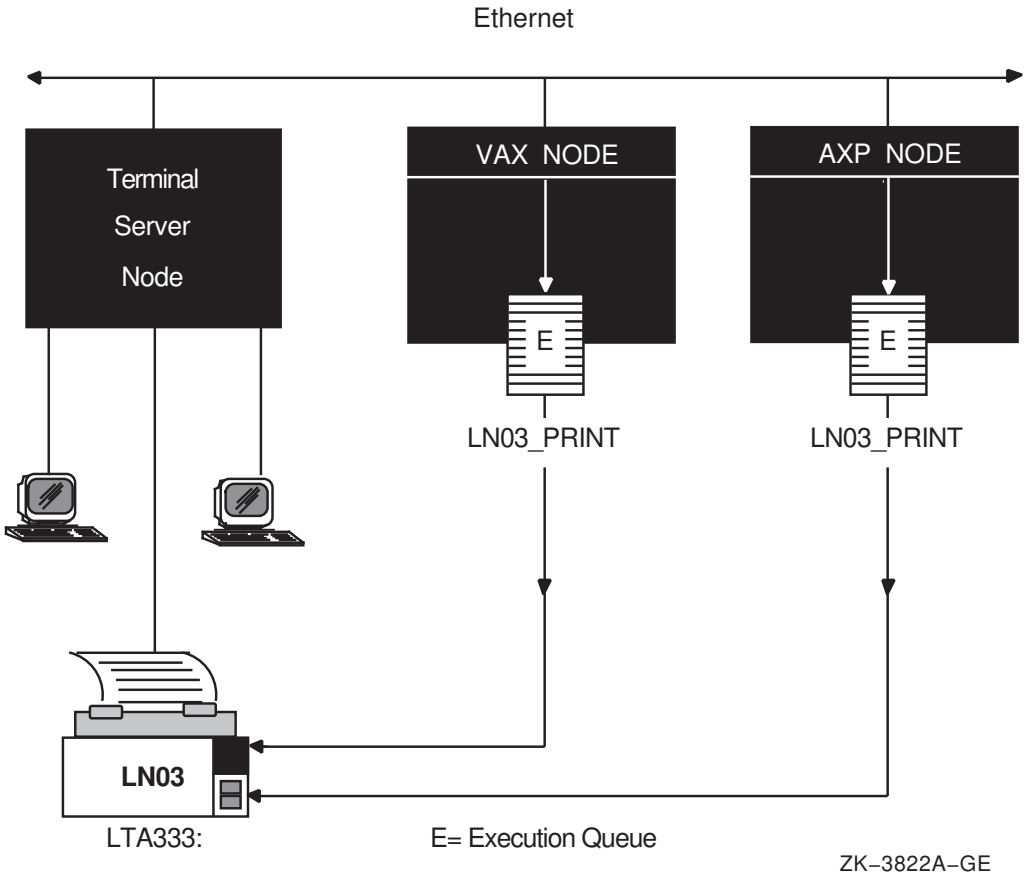
- A printer capable of printing PostScript files, and supporting software
- Software that provides PostScript-to-sixel printing, and a supported printer

For more information, see your Digital sales representative.

13.6.4 LAT Printers: Sharing Printers Among Multiple Systems

To share printers among multiple systems, or VMScluster environments, you can connect printers to a LAT port on a terminal server. Figure 13–6 illustrates an output queue configuration with a remote printer on a terminal server.

Figure 13–6 Configuration for Remote Printers on a Terminal Server

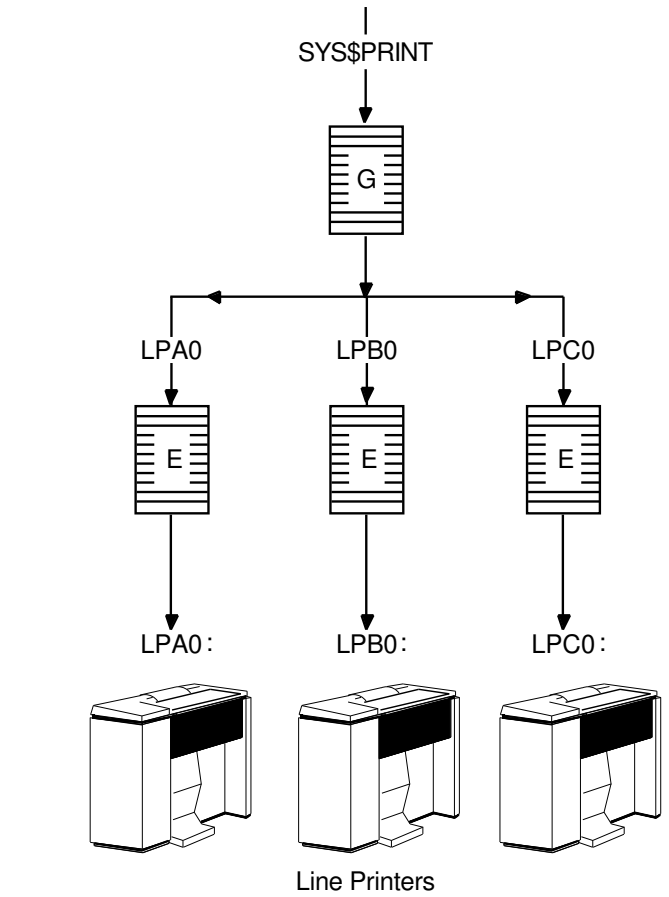


Digital recommends you set up your LAT queues as autostart queues with failover lists to ensure that these queues are highly available. Because LAT printers are usually shared among users of multiple systems or VMSclusters, many users will be affected if a LAT queue is unavailable. For more information on the autostart feature, see Section 13.4. For information about how to create autostart queues with failover lists, see Section 13.7.2.2.

13.6.5 Generic Output Queues: Distributing the Work Load

If you have more than one printer of the same kind (for example, if you have three line printers), you should use generic queues to balance the print load among the printers. Figure 13–7 illustrates such a configuration.

Figure 13–7 Queue Configuration with Three Like Printers and a Generic Queue

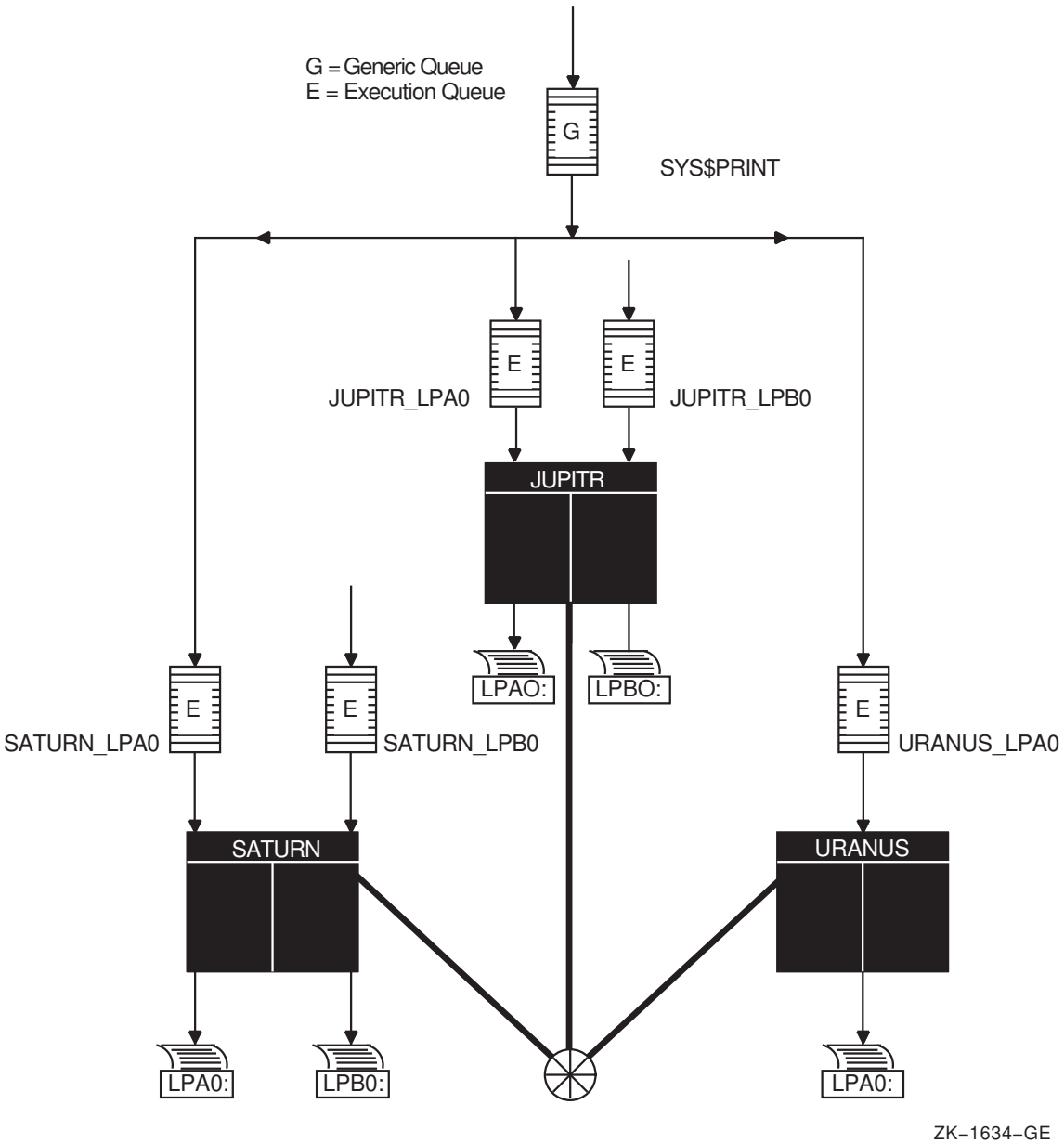


For information about how to create a generic queue, see Section 13.7.2.3.

13.6.6 VMScluster Queues: Sharing Devices in a Cluster

Figure 13–8 illustrates a typical VMScluster output queue configuration. For information about VMScluster queue configurations, see *VMScluster Systems for OpenVMS*.

Figure 13–8 Output Queue Configuration in a VMScluster

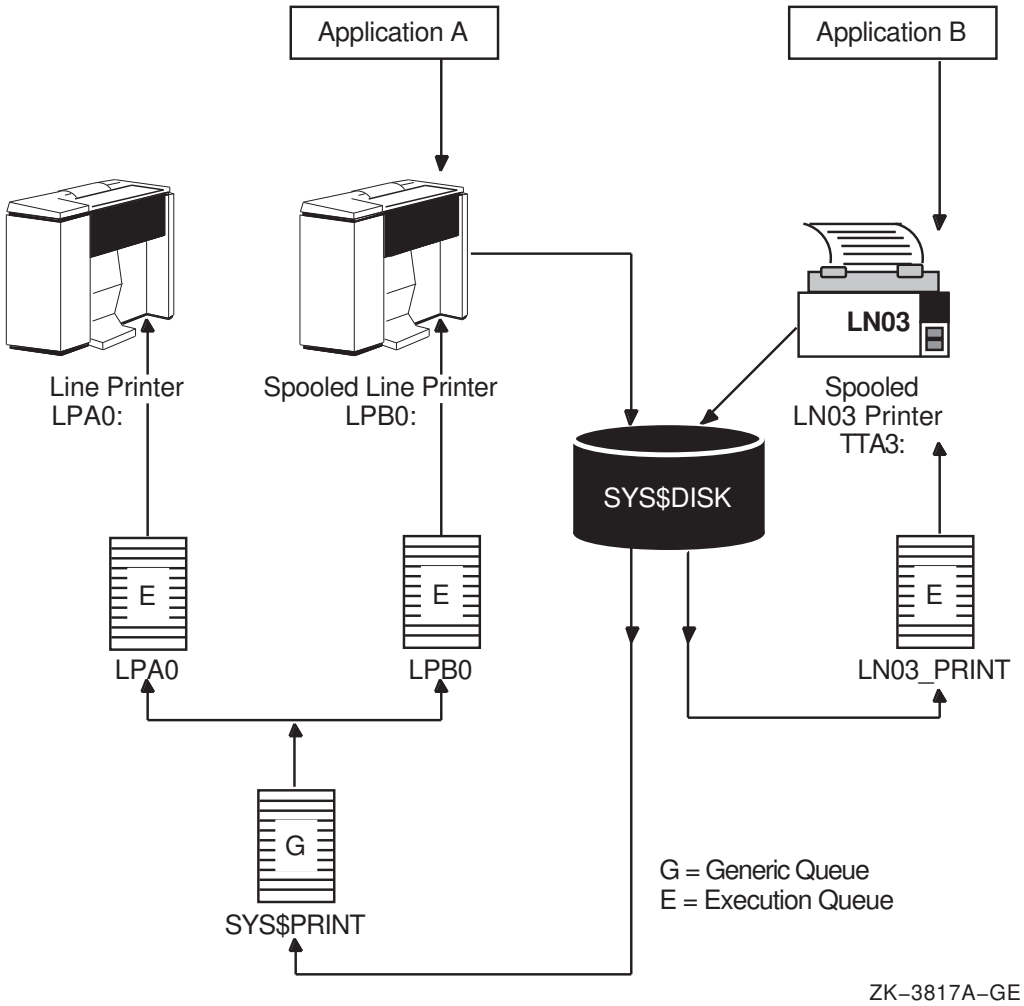


13.6.7 Spooled Printers: Printing from Applications

If your system runs application programs that write output directly to a printer rather than submitting it to an output queue, or if you will be using LAT queues, you should spool your printers. Spooling your printers causes application programs to write output to an intermediate storage device so that the printer remains available to other users while the program is running.

Figure 13–9 illustrates an output configuration with spooled printers.

Figure 13–9 Queue Configuration with Spooled Devices



For more information about spooling printers see Section 7.6.2.1.

13.6.8 Distributed Printing

The OpenVMS batch and print queuing system enables users to print files on output devices attached to the local system or VMScluster.

The Distributed Queuing Service (DQS) is a layered product that extends the printing capabilities of the OpenVMS queuing system to a distributed environment. DQS enables users to print files on output devices attached to remote nodes in your network.

For more information, see your DQS documentation or your Digital sales representative.

13.7 Setting Up and Starting Queues

You must create queues in order for users to submit jobs; you must start the queues so that jobs can begin processing. To set up and start queues, perform the following tasks, in order:

Task	For More Information
Make sure you have started the queue manager and created the queue database.	Section 12.4
If your configuration includes output queues, set up output devices and create a command procedure to set up the devices on reboot.	Section 13.7.1
If you plan to use any queue options, such as forms and characteristics and banner pages, determine the qualifiers needed to specify those options. In addition, define any forms and characteristics you will use before you create queues.	Section 13.8
Create and start queues.	Section 13.7.2
Create a command procedure to perform the necessary setup tasks each time your system reboots.	Section 13.7.2

13.7.1 Setting Up Output Devices

Before creating output queues, you must set up the devices to which the queues will direct output.

How to Perform This Task

1. Install any printers, plotters, and other output devices to which your users will have access. For information, see the documentation provided with the hardware.
2. If you will use LAT printers, create logical LAT ports. You must create a logical LAT port on each service node to which a LAT printer is to be available, and associate the logical port with a physical port or service on the terminal server node. To do so, use the LATCP commands CREATE PORT and SET PORT. For more information, see Section 24.5.2.
3. Set device characteristics for line printers and printers attached to terminal ports. To do so, use a series of SET commands. For more information, see Section 7.6.1. In step 6, you will create a command procedure to set up your devices each time the system reboots. The commands you enter to set device characteristics must be included in this command procedure.
4. Spool printers. If you use LAT printers, or if you run applications that write output directly to a printer, you should spool your printers. For more information on spooled printers, see Section 13.6.7.

   To spool a printer, use the SET DEVICE/SPOOLED command, as explained in Section 7.6.2.1.
5. Create a command procedure to set up your device characteristics and spool printers each time the system reboots. The commands you entered in steps 3 and 4 must be included in the command procedure. (The commands you entered to set up logical ports in step 2 should be included in your site-specific LAT startup command procedure SYS$MANAGER:LAT$SYSTARTUP.COM.)

If your configuration is simple, you can add the commands to SYSTARTUP_VMS.COM. If your configuration requires a large number of commands, create a separate command procedure (for example, DEVICE_SETUP.COM) and execute it from SYSTARTUP_VMS.COM.

In the command procedure, a SET TERMINAL command must precede a SET DEVICE/SPOOLED command for the same output device.

Example

```
$ SET PRINTER/TAB/PAGE=66/WIDTH=132/LOWER/FF/NOCR  -
_$ /FALLBACK/NOWRAP/NOTAB LPA0:
$ SET TERMINAL/SPEED=9600/PAGE=100/WIDTH=200/DEVICE=LN03/NOBROADCAST -
_$ /NOECHO/HARDCOPY/NOTYPE_AHEAD/NOFORM/NOWRAP/PASTHRU/PERMANENT LTA3331:
$ SET DEVICE/SPOOLED=(LPA0,SYS$SYSDEVICE) LPA0:
$ SET DEVICE/SPOOLED=(LN03_1,SYS$SYSDEVICE) LTA3331:
```

13.7.2 Creating and Starting Queues

Create execution queues before you create generic queues. You can start a queue either at the time you create the queue or later. For detailed instructions on creating and starting queues, see the following sections.

Task	For More Information
Autostart execution queues	Section 13.7.2.1
Nonautostart execution queues	Section 13.7.2.2
Generic queues	Section 13.7.2.3

13.7.2.1 Autostart Execution Queues

To create an autostart execution queue, specify the /AUTOSTART_ON qualifier with the INITIALIZE/QUEUE command.

For...	Use This Command...
Output queues	INITIALIZE/QUEUE[/START]/AUTOSTART_ON=(node::[device:][,...]) queue-name For *node::*, specify the name of the node on which the queue is to run. For *device:*, specify the name of the output device to which the queue's output is sent. To allow the autostart queue to fail over to another node and device, specify a list of nodes and devices, separated by commas. The optional /START qualifier activates the queue for autostart.
Batch queues	INITIALIZE/QUEUE/BATCH[/START]/AUTOSTART_ON=(node::[,...]) queue-name The /BATCH qualifier creates a batch queue. For *node::*, specify the name of the node on which the queue is to run. To allow the autostart queue to fail over to another node, specify a list of nodes, separated by commas. The optional /START qualifier activates the queue for autostart.

Caution

The node name you specify as *node::* is not checked to determine if it is an existing node name, so be sure to correctly specify the node name.

You can omit the /START qualifier, and activate the queue later using the START /QUEUE command. For more information, see the INITIALIZE/QUEUE and START/QUEUE commands in the *OpenVMS DCL Dictionary*.

Enabling Autostart to Start Autostart Queues

You must enable autostart on a node to start autostart queues. Once you have created autostart execution queues, perform the following steps to enable autostart:

1. For each node on which you want autostart queues to run (including those to which the queues can later fail over), enter the ENABLE AUTOSTART /QUEUES command. This command starts all stopped active autostart queues on a node and starts any active autostart queue that fails over to the node from another node in the VMScluster environment.
2. Add the ENABLE AUTOSTART/QUEUES command to your startup command procedure on each node that is to run autostart queues to ensure that autostart is enabled each time the node reboots.

Section 13.9.1.3 explains enabling autostart in detail.

Example

```
$ INITIALIZE/QUEUE/START/DEFAULT=(NOBURST,FLAG=ALL,TRAILER=ONE) -
_$ /AUTOSTART_ON=(LILITH::LPA0:,SMITTN::LPA0:)  LPA0 ❶
$ INITIALIZE/QUEUE/START/DEVICE=TERMINAL/ -
_$ /AUTOSTART_ON=(LILITH::LTA3331:,SMITTN::LTA555:) -
_$ /RECORD_BLOCKING/BLOCK_LIMIT=600/CHARACTERISTICS=(EAST)-
_$ /SEPARATE=(NOBURST,NOTRAILER,NOFLAG,RESET=ANSI$RESET) -
_$ /DEFAULT=(NOFEED,NOBURST,FLAG=ONE,NOTRAILER,FORM=MEMO) -
_$ /LIBRARY=LN03LIBRARY /PROCESSOR=LATSYM LN03_1 ❷
$ ENABLE AUTOSTART/QUEUES ❸
$ ENABLE AUTOSTART/QUEUES/ON_NODE=SMITTN  ❹
```

This example does the following:

❶ Creates an autostart queue named LPA0 and activates it for autostart. Because this is an autostart queue with a failover list, this queue can run on either LILITH::LPA0 or SMITTN::LPA0.

❷ Creates an autostart queue named LN03_1 for LAT printers and activates it for autostart. Because this is an autostart queue with a failover list, this queue can run on either of the printers attached to LAT ports LTA3331: on node LILITH or LTA555: on node SMITTN.

❸ Enables autostart on the node on which the process is running. Assume this is node LILITH. Because both LPA0 and LN03_1 are active autostart queues capable of running on node LILITH, these queues will start up on this node.

❹ Enables autostart on node SMITTN. If LILITH becomes unavailable, both LPA0 and LN03_1 can fail over to node SMITTN.

13.7.2.2 Nonautostart Execution Queues

To create a nonautostart execution queue, specify the /ON qualifier with the INITIALIZE/QUEUE command.

Type of Queue	Command
Output queues	INITIALIZE/QUEUE[/START]/ON=node::device queue-name For *node::*, specify the node on which the queue is to execute. For *device:*, specify the device to which the queue's output is sent. The /START qualifier starts the queue.
Batch queues	INITIALIZE/QUEUE/BATCH[/START]/ON=node:: queue-name The /BATCH qualifier is required to create a batch queue; *node::* is the node on which the queue is to execute. The /START qualifier starts the queue.

You can omit the /START qualifier, and start the queue later using the START /QUEUE command. For more information, see the INITIALIZE/QUEUE and START/QUEUE commands in the *OpenVMS DCL Dictionary*.

Example

```
$ INITIALIZE/QUEUE/BATCH/START/JOB_LIMIT=6/BASE_PRIORITY=4 -
_$ /ON=LILITH:: SYS$BATCH
$ INITIALIZE/QUEUE/BATCH/START/JOB_LIMIT=1/BASE_PRIORITY=5 -
_$ /NODISABLE_SWAPPING/AUTOSTART_ON=LILITH:: FAST_BATCH
```

13.7.2.3 Generic Queues

To create a generic queue, enter a command in the following format:

INITIALIZE/QUEUE[/START]/GENERIC=(execution-queue-list) generic-queue-name

For example:

```
$ INITIALIZE/QUEUE/START/GENERIC=(LPA0,LPB0,LPC0) LINE_PRINT
```

The /GENERIC qualifier specifies that the queue is a generic queue, and explicitly names a list of execution queues to which the generic queue sends jobs.

The /START qualifier starts the queue. You can omit the /START qualifier, and start the queue later using the START/QUEUE command.

You can also set up a generic queue without explicitly naming the execution queues to which it may send jobs. Instead, you enable certain execution queues to receive jobs from generic queues by specifying the /ENABLE_GENERIC qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command for the execution queue. This method is not normally recommended. However, if your queue configuration is simple, you can use this method. For more information, see the INITIALIZE/QUEUE, START/QUEUE, and SET QUEUE commands in the *OpenVMS DCL Dictionary*.

Example

```
$  INITIALIZE/QUEUE/START -
_$ /GENERIC=(LN03_1,LN03_2,LN03_3,LN03_4) LN03_PRINT
$  INITIALIZE/QUEUE/START/GENERIC=(LPA0:,LPB0:,LPC0:) SYS$PRINT
```

13.7.3 Creating a Command Procedure to Restart Execution Queues on Reboot

Information about forms and characteristics and queues is stored in the queue database. For this reason, creating forms queues and characteristics each time the node or VMScluster reboots is unnecessary. However, you must start nonautostart execution queues and enable autostart each time a node reboots. To do so, create a command procedure.

If your configuration is simple, you can add the commands to the site-specific startup command procedure SYSTARTUP_VMS.COM. If your configuration requires a large number of commands, create a separate command procedure and execute it from SYSTARTUP_VMS.COM.

Generic queues are not automatically stopped when a node shuts down. Therefore, including commands to start generic queues in your startup command procedure is unnecessary.

Starting Nonautostart Queues

For each queue that is not an autostart queue, you must include a START /QUEUE command naming the queue in your startup command procedure.

Enabling Autostart

You can start all stopped active autostart queues on a node by enabling autostart for queues with the single command ENABLE AUTOSTART/QUEUES. Including a separate START/QUEUE command to start an active autostart queue is not necessary. For details on enabling autostart, see Section 13.9.1.3.

When a node reboots, autostart is disabled until you enter the ENABLE AUTOSTART/QUEUES command. Add the ENABLE AUTOSTART/QUEUES command to your startup command procedure following the commands that configure printer devices and mount important disks. Digital recommends you add the ENABLE AUTOSTART/QUEUES command to your startup procedure on all of your nodes. This eliminates the necessity of adding the command later, if you need to add autostart queues or add nodes to autostart queue failover lists.

Example

The following example illustrates some sample commands that might be added to a node's SYSTARTUP_VMS.COM procedure.

```
$! Start the nonautostart batch queue
$ START/QUEUE SYS$BATCH
$! Start all autostart queues
$ ENABLE AUTOSTART/QUEUES
```

For more examples, see the SYS$MANAGER:SYSTARTUP_VMS.COM template on your system disk.

13.8 Using Queue Options

You can use certain options with your queues. For more information, see the following sections.

Options	Type of queue	Section
Controlling access to queues	Batch and output	Section 13.8.1
Job retention	Batch and output	Section 13.8.2
Characteristics	Batch and output	Section 13.8.3
Controlling batch processing	Batch	Section 13.8.5
Controlling job scheduling	Output	Section 13.8.6
Banner pages	Output	Section 13.8.7
Forms	Output	Section 13.8.9
Controlling line and page overflow	Output	Section 13.8.11

Options	Type of queue	Section
Suppressing initial form feed	Output	Section 13.8.12
Device control library modules	Output	Section 13.8.13

Most options can be implemented by specifying appropriate qualifiers with the INITIALIZE/QUEUE command when you create the queue. (You can also specify options after you create a queue by including qualifiers with the START/QUEUE or SET QUEUE command.) Table 13–1 lists the qualifiers you use to specify options for batch queues, and indicates the type of queue for which you can specify the option. See the *OpenVMS DCL Dictionary* for more details about these qualifiers.

Table 13–1 Qualifiers for Specifying Queue Options

Qualifier	Type of Queue	Description
/AUTOSTART_ON	Batch and output	Creates an autostart execution queue and specifies the node or nodes (and for output queues, the device or devices) on which the queues can run. For more information, see Section 13.7.2.2.
/BASE_PRIORITY	Batch and output	Specifies a base process priority (not the same as the job scheduling priority). For a batch queue, specifies the base priority for processes executing jobs in the queue. For output queues, specifies the base priority of the symbiont process. For more information, see Section 13.8.5.1 and the *OpenVMS DCL Dictionary*.
/BLOCK_LIMIT	Output	Limits the size of print jobs that can be processed on an output execution queue. For more information, see Section 13.8.6.
/CHARACTERISTICS	Batch and output	Specifies one or more characteristics associated with the queue. For more information, see Section 13.8.3.
/CPUDEFAULT	Batch	Defines the base value for the default CPU time limit for batch jobs executed in the queue. For more information, see Section 13.8.5 and the *OpenVMS DCL Dictionary*.
/CPUMAXIMUM	Batch	Defines a maximum CPU time limit for batch jobs executed in the queue. For more information, see Section 13.8.5 and the *OpenVMS DCL Dictionary*.
/DEFAULT	Output	Establishes defaults for certain options of the PRINT command. After you set an option for the queue with the /DEFAULT qualifier, users do not have to specify that option in their PRINT commands. However, they can specify options to override the defaults set on the queues. Possible default options are as follows: BURST—Controls whether file flag pages with a burst bar between them are printed preceding output. For more information, see Section 13.8.7.

(continued on next page)

Table 13–1 (Cont.) Qualifiers for Specifying Queue Options

Qualifier	Type of Queue	Description
		FEED—Specifies whether a form feed is inserted automatically at the end of a page. For more information, see Section 13.8.11.
		FLAG—Controls whether a file flag page is printed preceding output. For more information, see Section 13.8.7.
		FORM—Specifies the default form for an output execution queue. If a job is submitted without an explicit form definition, this form is used to process the job. For more information, see Section 13.8.9.
		TRAILER—Controls whether a file trailer page is printed following output. For more information, see Section 13.8.7.
/DESCRIPTION	Batch and output	Specifies a text string to provide users with information about the queue. For more information, see the *OpenVMS DCL Dictionary*.
/DEVICE	Output	Specifies the type of output execution queue. The keywords are as follows:
		PRINTER (default)—Specifies a printer queue. This is the default.
		TERMINAL—Specifies a terminal queue.
		SERVER—Specifies a server queue.
		For more information, see Section 13.3.2 and the *OpenVMS DCL Dictionary*.
/DISABLE_SWAPPING	Batch	Specifies whether batch jobs executed from a queue can be swapped in and out of memory. For more information, see Section 13.8.5.
/FORM_MOUNTED	Output	Specifies the mounted form for an output execution queue. For more information, see Section 13.8.9.
/GENERIC	Batch and output	Creates a generic queue and names the execution queues it feeds. For more information, see Section 13.7.2.3.
/JOB_LIMIT	Batch	Indicates the number of batch jobs that can be executed concurrently from a batch queue. For more information, see Section 13.8.5.
/LIBRARY	Output	Specifies the file name for a device control library. For more information, see Section 13.8.13.
/NAME_OF_MANAGER	Batch and output	Specifies the name of the queue manager with which the queue will be associated. For more information, see Section 12.8.
/NO_INITIAL_FF	Output	Specifies the qualifier for an output execution queue; suppresses the initial form feed sent to an output execution queue. For more information, see Section 13.8.12.

(continued on next page)

Table 13–1 (Cont.) Qualifiers for Specifying Queue Options

Qualifier	Type of Queue	Description
/ON	Batch and output	Creates a nonautostart execution queue and specifies the node (and, for output queues, the device) on which the queue is to run. For more information, see Section 13.7.2.2.
/OWNER_UIC	Batch and output	Specifies the user identification code (UIC) for the queue. For more information, see Section 13.8.1.
/PROCESSOR	Output	Specifies the symbiont to be used with an output execution queue. The default is the standard operating system print symbiont PRTSMB. For more information, see Section 13.3.2 and the *OpenVMS DCL Dictionary*.
/PROTECTION	Batch and output	Specifies a protection for the queue. For more information, see Section 13.8.1.
/RECORD_BLOCKING	Output	Determines whether the symbiont can concatenate (or block together) output records for transmission to the output device. For more information, see the *OpenVMS DCL Dictionary*.
/RETAIN	Batch and output	Holds jobs in the queue after they have executed. For more information, see Section 13.8.2.
/SCHEDULE	Output	Specifies whether pending jobs in a queue are scheduled based on the size of the job. For more information, see Section 13.8.6.
/SEPARATE	Output	Specifies required job separation or job reset options for an output execution queue. Required options cannot be overridden by the PRINT command. Possible options are as follows: BURST—Controls whether two job flag pages with a burst bar between them are printed at the beginning of each job. For more information, see Section 13.8.7. FLAG—Controls whether a job flag page is printed at the beginning of each job. For more information, see Section 13.8.7. RESET—Specifies one or more device control library modules you want to use to reset the device after any file trailer and before any job trailer. For more information, see Section 13.8.13. TRAILER—Controls whether a job trailer page is printed at the end of each job. For more information, see Section 13.8.7.

(continued on next page)

Table 13–1 (Cont.) Qualifiers for Specifying Queue Options

Qualifier	Type of Queue	Description
/WSDEFAULT	Batch and output	For batch queues, specifies a default working set size for batch jobs executed in the queue. For output queues, specifies a default working set size for the symbiont process. The value set by this qualifier overrides the value defined in the UAF of any user submitting a job to the queue. For more information, see Section 13.8.5.
/WSEXTENT	Batch and output	For batch queues, specifies the working set extent for batch jobs executed in the queue. For output queues, specifies a working set extent for the symbiont process. The value set by this qualifier overrides the value defined in the UAF of any user submitting a job to the queue. For more information, see Section 13.8.5.
/WSQUOTA	Batch and output	For batch queues, specifies the working set quota for batch jobs executed in the queue. For output queues, specifies a working set quota for the symbiont process. The value set by this qualifier overrides the value defined in the UAF of any user submitting a job to the queue. For more information, see Section 13.8.5.

13.8.1 Controlling Access to Queues

As with a file or directory, you can use UIC-based or ACL-based protection to control access to a queue. See the *Security Guide* for detailed information about establishing system security.

13.8.1.1 Understanding UIC-Based Queue Protection

UIC-based protection restricts the jobs and the users who have access to a queue. Operations that apply to queues are controlled by UIC-based protection in the same way that access to other protected objects (such as files) is controlled.

UIC-BASED Protection

When you create a queue, the queue is assigned an owner UIC and a protection code. The default owner is [SYSTEM], but you can specify another owner with the /OWNER_UIC qualifier. On VAX systems, the default UIC-based protection for a queue is as follows:

```
(System:M,Owner:D,Group:R,World:S)
```

Jobs are assigned an owner UIC equal to the UIC of the process that submitted the job, unless the job was submitted with the /USER qualifier. Each job in a queue, and each operation that is performed on a queue, is checked against the UIC of the owner, the protection of the queue, and the privileges of the requester.

All operations are checked as follows:

Operation Type	Check
Operations that apply to jobs	Checked against the read and delete protection specified for the queue and the owner UIC of the job. In general, read access to a job allows a user to see the information about a job, and delete access allows the user to modify and delete the job.
Operations that apply to queues	Checked against the submit and manage protection specified for the queue and the owner UIC of the queue. Submit access to a queue allows a user to submit jobs to that queue. Read access allows a user to see security information about a queue. Manage access allows a user to act as the operator for that queue, with the ability to affect any jobs in the queue. Users with operator (OPER) privilege have manage access to all queues. OPER privilege also enables users to create queues and modify security aspects of queues.

The following table lists each type of access available on VAX systems, and explains what functions the access controls when applied to a queue. It also explains the default value provided on a queue for the access type.

Access	Function	Default
Submit	Submit jobs to the queue	World:Submit—lets all users submit jobs to the queue.
Delete	Modify and delete jobs in a queue	Owner:Delete—lets users modify and delete their own jobs.
Read	Display attributes of jobs in a queue	Group:Read—lets users see the attributes of the jobs of all users in their UIC group. Also lets users in the queue's UIC group see security information.
Manage	Modify any non-security attribute of a queue and delete the queue, plus submit and read access to the queue and read and delete access to all jobs in the queue	System:Manage—lets users with a system UIC act as operators for the queue.

For more information on UIC-based protection, see the *OpenVMS VAX Guide to System Security*.

13.8.1.2 Setting and Showing UIC-Based Queue Protection

Use the following commands to set and show UIC-based protection on queues. For more information on these commands, see the *OpenVMS DCL Dictionary*.

Command	Description
INITIALIZE/QUEUE/PROTECTION=(*ownership[:access],...*) START/QUEUE/PROTECTION=(*ownership[:access],...*) SET QUEUE/PROTECTION=(*ownership[:access],...*)	Changes the default UIC protection code on a queue.
INITIALIZE/QUEUE/OWNER_UIC=*uic* START/QUEUE/OWNER_UIC=*uic* SET QUEUE/OWNER_UIC=*uic*	Changes the owner UIC of a queue.

Command	Description
SHOW QUEUE/FULL	Shows the protection currently set for a queue.
SET SECURITY/CLASS=QUEUE/OWNER=*uic*	Changes the UIC protection code on a queue.
SET SECURITY/CLASS=QUEUE /PROTECTION=*ownership=access*	Changes the default UIC protection code on a queue.
SHOW SECURITY/CLASS=QUEUE	Shows protection currently set for a queue.

Examples

1. The following example sets UIC-based protection on a queue, and then displays information, including security information, about the queue.

   Note

   On VAX systems, the following display would show S (SUBMIT) access instead of W (WRITE) access and M (MANAGE) access instead of E (EXECUTE) access.

```
$ INITIALIZE/QUEUE/GENERIC=(SYS_QUE1,SYS_QUE2)/PROTECTION=(S:E,O:D,G:R,W:W) -
_$ /OWNER_UIC=[1,4]/RETAIN=ERROR SYS_PRINT
$ SHOW QUEUE/FULL SYS_PRINT
Generic printer queue SYS_PRINT/GENERIC=(SYS_QUE1,SYS_QUE2) -
_$ /OWNER=[1,4]/PROTECTION=(S:E,O:D,G:R,W:W)/RETAIN=ERROR
```

2. The following example gives owner manage and delete access to this queue and makes user AGBELL the owner. With manage acccess, the owner AGBELL can manage the queue, but cannot modify security information.

```
$ SET SECURITY/CLASS=QUEUE/OWNER=[AGBELL]/PROTECTION=OWNER=MD  -
_$ TELEPHONE_QUE
$ SHOW SECURITY/CLASS=QUEUE TELEPHONE_QUEUE
TELEPHONE_QUEUE object of class QUEUE
     Owner: [INVENTORS,AGBELL]
     Protection: (System: M, Owner: MD, Group: R, World: S)
     Access Control List: <empty>
```

13.8.1.3 Understanding ACL-Based Queue Protection

In addition to UIC-based protection, you can associate access control lists (ACLs) with a queue. ACL-based protection provides a more refined level of protection in cases where certain members of a project group require access to a queue, excluding others of the same UIC group or of other groups. See the *Security Guide* for detailed information about establishing ACLs for protected objects.

13.8.1.4 Setting and Showing ACL-Based Queue Protection

Use the following commands to set and show ACL-based protection on queues:

Command	Description
SET ACL/OBJECT_TYPE=QUEUE - /ACL=(IDENTIFIER=(*identifier*,ACCESS=*access-code*)[,...])	Sets ACL protection on a queue.
SHOW QUEUE/FULL	Shows the ACLs set on a queue, if any.

Command	Description
†SET SECURITY/CLASS=QUEUE/ACL=(IDENTIFIER=*identifier*, ACCESS=*access*)	On VAX systems, sets ACL protection on a queue.
†SHOW SECURITY/CLASS=QUEUE	On VAX systems, shows ACLs set on a queue, if any.

†VAX specific

For more information on these commands, see the *OpenVMS DCL Dictionary*. For more information on ACL-based security see the *Security Guide*.

Examples

1. The SET QUEUE/PROTECTION command in the following example modifies the default UIC-based protection of queue SYS_QUE1 to prevent access by nonprivileged users. The SET ACL command then restricts access to only those members of a project group who hold the ULTRA_LITE or MINUTES identifiers. Members with the MINUTES identifier have only read and submit access to the queue. The SHOW QUEUE/FULL command displays information, including security information, about the queue.

```
$ SET QUEUE/PROTECTION=(S,O,G,W)
$ SET SECURITY/CLASS=QUEUE SYS_QUE1 -
_$/ACL=((IDENTIFIER=ULTRA_LITE, ACCESS=READ+SUBMIT+MANAGE+DELETE), -
_$ (IDENTIFIER=MINUTES, ACCESS=READ+SUBMIT)) SYS_QUE1
$ SHOW QUEUE/FULL SYS_QUE1
Batch queue SYS_QUE1, stopped
    /BASE_PRIORITY=4 /JOB_LIMIT=1 /OWNER=[1,4] /PROTECTION=(S,O,G,W)
          (IDENTIFIER=ULTRA_LITE,ACCESS=READ+SUBMIT+MANAGE+DELETE)
          (IDENTIFIER=MINUTES,ACCESS=READ+SUBMIT)
```

2. The following example shows how to use ACLs to restrict queue access to members of a particular project group:

```
$ SET QUEUE/PROTECTION=(S,O,G,W)
$ SET ACL/OBJECT_TYPE=QUEUE SYS_QUE1 -
_$/ACL=((IDENTIFIER=ULTRA_LITE, ACCESS=READ+SUBMIT+MANAGE+DELETE), -
_$ (IDENTIFIER=MINUTES, ACCESS=READ)) SYS_QUE1
```

3. The following example shows a queue that has only UIC-based protection, and then gives user AGBELL control access with an ACL. Control access allows user AGBELL to modify security information.

```
$ SHOW SECURITY/CLASS=QUEUE TELEPHONE_QUEUE
TELEPHONE_QUEUE object of class QUEUE
     Owner: [INVENTORS,AGBELL]
     Protection: (System: M, Owner: MD, Group: R, World: S)
     Access Control List: <empty>

$ SET SECURITY/CLASS=QUEUE/ACL=(ID=[AGBELL],ACCESS=CONTROL) TELEPHONE_QUEUE
$ SHOW SECURITY/CLASS=QUEUE TELEPHONE_QUEUE

TELEPHONE_QUEUE object of class QUEUE
     Owner: [INVENTORS,AGBELL]
     Protection: (System: M, Owner: MD, Group: R, World: S)
     Access Control List:
          (IDENTIFIER=[CLASS,AGBELL],ACCESS=CONTROL)
```

13.8.1.5 Understanding How Privileges Affect Queues

Certain account privileges allow users to access a queue in spite of UIC-based and ACL-based protection. The following table lists these account privileges and the type of access they allow on a queue:

Privilege	Access
OPER	Manage and control access to all queues.
BYPASS	Manage and control access to all queues.
READALL	Read access to all jobs and to queue security information.
SYSPRV	The access specified for users with system UICs.
GRPPRV	The access specified for users with system or group UICs.

13.8.2 Understanding Job Retention Options

Users can request that a job be retained in a queue after the job completes by specifying the /RETAIN qualifier when they submit a job with the PRINT or SUBMIT command.

In addition, you might want to set a job retention policy on a queue to keep information about all jobs in the queue after the jobs complete. This might be helpful when an individual must track jobs submitted by other users.

By default, no job retention policy is set on a queue. To specify a job retention policy for a queue, use the /RETAIN=*option* qualifier with the INITIALIZE /QUEUE, START/QUEUE, or SET QUEUE command. You can specify one of the following options:

Option	Description
ALL	Holds all jobs in the queue regardless of the jobs' completion statuses.
ERROR	Holds jobs in the queue if they complete unsuccessfully.

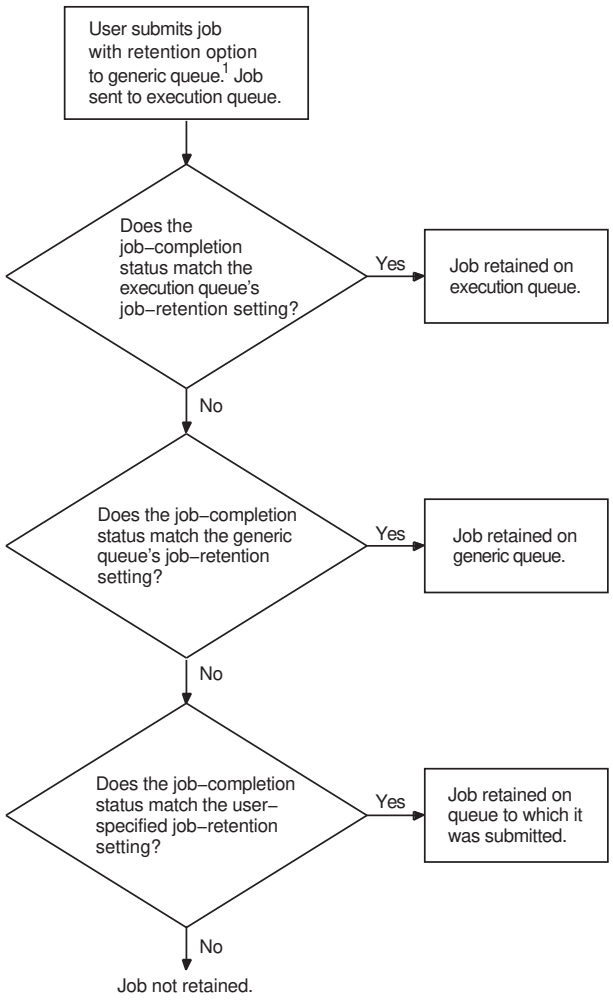
The following command specifies that the queue retain all jobs that complete with a status other than success:

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

For example, if an operator must be aware of all batch jobs that do not complete successfully on a certain queue, you could set the queue to retain jobs that complete with an error status. This would let the operator enter the SHOW QUEUE command to display a list of jobs that completed unsuccessfully. The SHOW QUEUE display includes the completion status of jobs retained in the queue. If a job completes unsuccessfully, this message can help determine why. The displays also include the date and time at which a retained job completed.

If you set a job retention policy on a queue, the queue's job retention option overrides the job's retention option. Figure 13–10 shows how job retention affects a job submitted to a generic queue.

Figure 13–10 Determining Job Retention



[1] Jobs submitted directly to execution queues are not affected by the job–retention settings on generic queues.

ZK–5926A–GE

Whether and where a job is retained is determined by the following:

- The retention setting on the execution queue in which the job executes
- The retention setting on the generic queue (if the job is submitted to a generic queue)
- The completion status of the job
- The retention requested by the user upon submitting the job (if retention was requested)

If jobs are retained in queues, you should periodically delete jobs that no longer need to be retained.

Users can specify timed job retention to retain a job only as long as they need it. For example:

```
$ SUBMIT/RETAIN=UNTIL=19-MAY-1995:07:31:0.0 MYFILE.DAT
```

This eliminates the necessity of deleting retained jobs from queues. If your users make use of the /RETAIN qualifier, encourage them to use timed retention.

For more information about job retention, see the INITIALIZE/QUEUE/RETAIN and SUBMIT/RETAIN commands in the *OpenVMS DCL Dictionary*.

To remove a job retention option from a queue, specify the /NORETAIN qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command.

13.8.3 Understanding Queue Characteristics

You can define characteristics and assign them to queues to control the jobs that execute on certain queues.

When users include the /CHARACTERISTIC qualifier with a PRINT or SUBMIT command, all the characteristics they specify must also be assigned to the queue that will execute the job. If a job is placed in a queue that does not have the characteristics required by that job, the job enters a pending state. The job remains pending in the queue until you correct the characteristic mismatch as explained in Section 13.10.2.2. Users do not need to specify all characteristics of a queue when they enter a PRINT or SUBMIT command, as long as the characteristics specified are a subset of those established for the queue. Similarly, you can define print forms for output queues (including, for example, a specific paper stock and page format) and assign the forms to queues to control the jobs that execute on certain queues. For more information, see Section 13.8.9.

Example

Suppose a system manager manages 12 LN03 printers, with three printers in each of the four corners of a building. A generic queue LN03$PRINT feeds execution queues for each of the LN03 printers. The system manager defines four characteristics: EAST, WEST, NORTH, and SOUTH. The EAST characteristic is assigned to the execution queues that feed printers in the eastern corner of a building. WEST is assigned to the execution queues feeding the printers in the western corner, and so on. When a user submits a print job to LN03$PRINT and specifies the EAST characteristic, the job prints on the first idle LN03 printer in the eastern corner of the building.

If the system had queues for printers on multiple floors, the system manager could further define a characteristic for each floor, for example, FIRST, SECOND, and THIRD.

13.8.4 Specifying Queue Characteristics Options

To assign characteristics to a queue, specify the /CHARACTERISTICS qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. You must create a characteristic with the DEFINE/CHARACTERISTIC command before assigning the characteristic to the queue. For more information on creating and managing characteristics, see Section 13.9.2.

13.8.5 Specifying Batch-Processing Options

You can use queue options to control batch job performance and the use of system resources by processes executing batch jobs.

You use the following qualifiers with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command to set these queue options:

Option	Description
/JOB_LIMIT=*n*	Specifies the number of jobs that can execute simultaneously in the queue.
/[NO]DISABLE_SWAPPING	Specifies whether the processes running jobs on the queue can be swapped out of memory.
/CPUDEFAULT=*time*	Specifies the base value for the default CPU time limit. The time cannot exceed the time limit set by the /CPUMAXIMUM qualifier.
/CPUMAXIMUM=*time*	Specifies the maximum CPU time limit.

You can also use the following qualifiers. Although these qualifiers are not specific to batch queues, they are commonly used like the batch-specific options, to control batch job performance and the use of system resources by batch processes.

Option	Description
/BASE_PRIORITY=*n*	Specifies the base process priority at which jobs are initiated from a batch queue.
/WSDEFAULT=*n*	Specifies the default working set size for jobs executed in a batch queue. (For output queues, specifies the default working set size for symbiont processes.)
/WSEXTENT=*n*	Specifies the working set extent for jobs executed in a batch queue. (For output queues, specifies the working set extent for symbiont processes.)
/WSQUOTA=*n*	Specifies the working set quota for jobs executed in a batch queue. (For output queues, specifies the working set quota for symbiont processes.)

For more information about these limits, quotas, and priorities, see the following manuals:

- The INITIALIZE/QUEUE command in the *OpenVMS DCL Dictionary*
- The *Guide to OpenVMS Performance Management*

By default, a process running a batch job uses values taken from the UAF record of the user submitting the job or from system parameter settings. For more information about the default values for these options, see the description of the qualifiers for the INITIALIZE/QUEUE command in the *OpenVMS DCL Dictionary*. If you specify values for any of these options, processes for jobs executed in the queue will use the values you set unless the user specifies values when the job is submitted. (A user can specify values for CPU time and for the working set options default, quota, and extent.) If a user specifies a value for one of these options when the job is submitted, this value cannot exceed the value you set for the queue. If you did not specify a value, the value specified by the user cannot exceed the value specified in the associated UAF limit or system parameter.

The following sections provide guidelines for choosing values for these options.

Option	For More Information
Base process priority	Section 13.8.5.1
Job limit	Section 13.8.5.2
Working set default, quota, and extent	Section 13.8.5.3
CPU default and maximum	Section 13.8.5.4
Swapping	Section 13.8.5.5
Options for memory-constrained systems	Section 13.8.5.6
Optimizing for the Sort/Merge utility	Section 13.8.5.7

13.8.5.1 Base Process Priority

Choose a value based on how quickly you will allow batch jobs to progress. If you choose a value equal to the system parameter value DEFPRI (typically 4), jobs in this queue will progress at the same rate as typical interactive jobs. This choice might be appropriate for systems that have an abundance of available CPU time.

If, on the other hand, you choose a value less than DEFPRI, jobs in this queue will potentially progress more slowly than the typical interactive job. CPU time will be allocated to jobs in this queue only after servicing jobs of DEFPRI priority.

In the case of a queue defined for a very special purpose—for example, high-priority jobs—a value greater than DEFPRI (for example, 5) might be appropriate. However, this choice can have a significant negative effect on the performance of other users and batch jobs.

13.8.5.2 Job Limit

Keep this value low when using a base process priority of DEFPRI or greater, since each batch job can affect the performance of interactive jobs.

13.8.5.3 Working Set Default, Quota, and Extent

If you do not specify values for these options, a job uses the value specified in its owner's user authorization file (UAF) record.

Process Limit	Description
Working set default	The value to which the working set returns at the exit of each image. Typically, the value should be relatively small and is usually best left at the value specified in the user's UAF record.
Working set quota	The value that approximates the amount of physical memory used by each batch job in the queue in a *memory-constrained* system.
Working set extent	The value that approximates the amount of physical memory in a *memory-rich* system. Typically, you should set this to a high value. The working set extent value is an upper limit for the size of the working set; the working set cannot be expanded beyond this value even if more memory is required by the job. If you set this value too low, a job might page fault heavily even if the system has plenty of memory available. To be safe, choose a working set extent value equal to the system parameter value of WSMAX, which specifies the maximum working set size possible for your system.

In general, the working set quota and extent values specified in the UAF record for each user are sufficient. However, you might want to specify more generous

or stringent values for a queue, depending on the purpose of the queue. For example, you might encourage users to submit large jobs (such as compiling and linking large programs) as batch jobs to reserve interactive use of the system for jobs that do not require extensive resources, as follows:

- Set a large working set size for batch jobs by specifying larger WSQUOTA and WSEXTENT values when you create the batch queue.
- Restrict the working set size of interactive jobs by providing smaller WSQUOTA and WSEXTENT values in users' UAF records.

13.8.5.4 CPU Default and Maximum

You can restrict and expand the amount of time a job is allowed in the CPU by setting CPU limit values.

Be careful to not set CPU time limit values too low. When a job's CPU time limit is exceeded, the job is terminated with an error status. If working set values (explained in the previous section) are set too low, the system might use memory less efficiently but the jobs can still complete normally. However, if CPU time limit values are set too low, the system might terminate jobs that are making legitimate and authorized use of the CPU.

For example, you might use a CPU time limit on a batch queue devoted to execution of newly coded software that could unexpectedly enter a CPU loop. The CPU time limit would terminate infinitely looping jobs so they do not waste the CPU resource. However, you must be careful to choose a sufficiently high time limit so normally executing jobs do not terminate prematurely.

Another way to control allocation of the CPU resource is to create a special-purpose batch queue that shifts execution of its jobs to a less busy time of day when CPU time is more available.

13.8.5.5 Swapping

Typically, swapping is enabled on batch queues. However, for a very special-purpose, high-priority queue, you might disable swapping. This provides a favorable status to jobs in this queue by removing them from consideration when memory must be reclaimed from processes (through the operating system's swapping and trimming mechanism). For more information, see the *Guide to OpenVMS Performance Management*.

13.8.5.6 Setting Up Batch Queues on Memory-Constrained Systems

On memory-constrained systems, you should add up the pages required for the batch working sets on all batch queues and make sure that enough fluid memory remains for interactive jobs. Fluid memory can be reassigned from one process to another through swapping and paging. (You can calculate fluid memory as the space that remains when you subtract the number of pages permanently allocated to the operating system from the total memory. To obtain these values, enter the DCL command SHOW MEMORY.)

13.8.5.7 Optimizing Batch Queues for the Sort/Merge Utility

You can improve the efficiency of the OpenVMS Sort/Merge utility (SORT /MERGE) by designating one batch queue for sorting jobs and giving this queue a very large working set quota.

In addition, you can set process quotas for greatest SORT efficiency. The values recommended here are based solely on SORT considerations; you should integrate other system considerations with these to determine appropriate values.

- Working set extent per process (WSEXTENT)—For maximum sorting efficiency, use the /WSEXTENT qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command to set this value on the queue to the largest value any sorting job would ever need. Generally, the smaller the working set, the slower the sort.

  For very large files, working sets of 4000 or more pages (on VAX systems) or pagelets (on Alpha systems) are not unreasonable, provided the system has enough physical memory to support them.

- Virtual page count per process (VIRTUALPAGECNT system parameter)—For maximum sorting efficiency, set this system parameter to at least 1000 pages (on VAX systems) or pagelets (on Alpha systems) more than the maximum working set extent. Although SORT limits the size of the sort data structure to the process's working set extent, if work files are required, SORT can use the extra memory for I/O buffers. If this parameter is too low relative to the working set extent, SORT might be unable to create buffers for the work files when the files cannot be sorted in memory.

13.8.6 Specifying Job Scheduling Options

For output queues, you can choose options that control how the size of a print job affects its scheduling. Specify the following qualifiers with the INITIALIZE /QUEUE, START/QUEUE, or SET QUEUE command to control print job scheduling:

Option	Description
/SCHEDULE=[NO]SIZE	Lets you control whether pending jobs are scheduled for printing based on the size of the job.
/[NO]BLOCK_LIMIT=([*lowlim,*]*uplim*)	Lets you reserve certain printers for certain size jobs.

For more information on job scheduling, see Section 13.9.6.4.

/SCHEDULE

By default, printer and terminal queues are created with the /SCHEDULE=SIZE qualifier, causing shorter jobs to print before long ones. However, you can specify the /SCHEDULE=NOSIZE qualifier, causing jobs to be printed in the order they were submitted, regardless of size. For more information, see the /SCHEDULE qualifier for the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command in the *OpenVMS DCL Dictionary*.

/BLOCK_LIMIT

By default, printer and terminal queues are created with no block limit, so jobs of any size will be printed. However, you can specify the /BLOCK_LIMIT qualifier to limit the size of jobs that can be printed. You can specify just an upper limit or both an upper and lower limit.

For example, you could submit a batch job that runs a command procedure to set a maximum block limit of 500 blocks for a queue at 8:00 a.m. The command procedure might resubmit itself and remove the block limit after 5:00 p.m. each day. This would let users print jobs of any size after normal work hours, when the printing work load is lighter. Users can specify the /AFTER qualifier with the PRINT qualifier to specify that a job is to be printed after a certain time.

For more information about restricting print job sizes, see the /BLOCK_LIMIT qualifier for the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command in the *OpenVMS DCL Dictionary*.

13.8.7 Understanding Banner Pages

You should understand the following terms:

Term	Definition
Banner pages	Specially formatted pages that you can set up for queues, to help in identifying and separating output jobs when they exit a printer; also to help identifiy and separate files within jobs.
Job pages	Banner pages that identify and separate *jobs*; users cannot override job banner pages that you set for a queue.
File pages	Banner pages that identify and separate *files* within a job; users can override the file banner page settings you set for a queue by specifying the BURST, FLAG, and TRAILER qualifiers with the PRINT command.

Most sites use only a subset of the available banner pages at any given time. The following table describes the types of banner pages you can use:

Type	Description	For More Information
	Job Pages	
Flag page	A single page printed before each job.	Figure 13–11
Burst page	Two flag pages, divided by a separation (burst) bar, printed before each job	Figure 13–11
Trailer page	A page printed after each job.	Figure 13–13
	File Pages	
Flag page	A single page printed before each file in a job.	Figure 13–12
Burst page	Two flag pages, divided by a separation (burst) bar, printed before each file in a job.	Figure 13–12
Trailer page	A page printed after each file in a job.	Figure 13–14

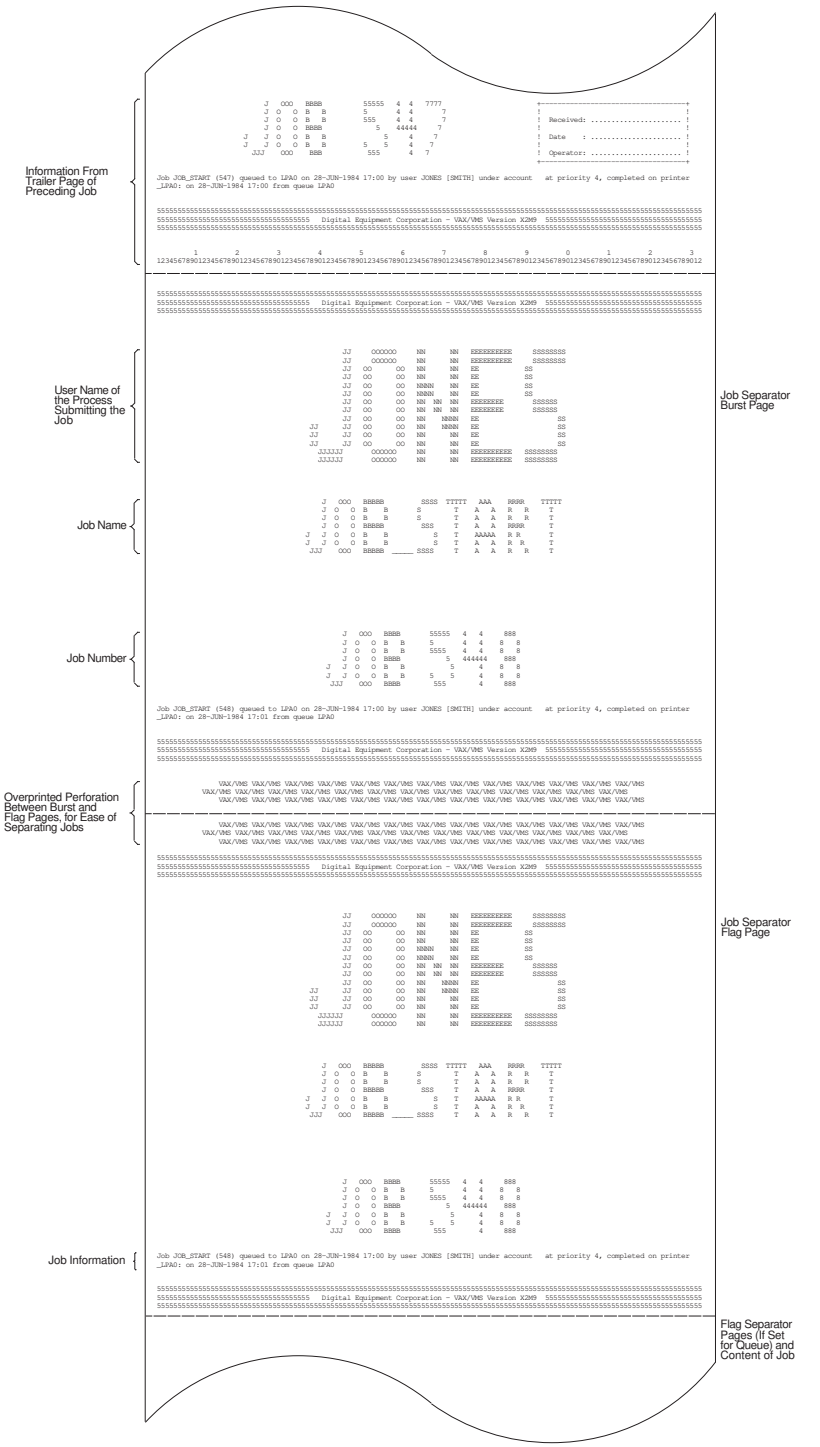
13.8.7.1 Flag and Burst Pages

Flag and burst pages both indicate that a print file follows. However, a burst page is two flag pages divided by a **burst bar**, which is designed to print over the crease between two pages of continuous-form paper. The burst bar is used for ease in separating continuous-form paper. If you do not need to burst pages of a printer's output—for example, if your printer uses cut sheet paper—avoid the burst page option. Flag pages, or flag and trailer pages, are usually sufficient for identifying the end of a job.

Figure 13–11 illustrates job flag and burst pages produced by the default print symbiont, PRTSMB.

Figure 13–12 illustrates file flag and burst pages produced by PRTSMB.

Figure 13–11 Job Flag and Burst Pages



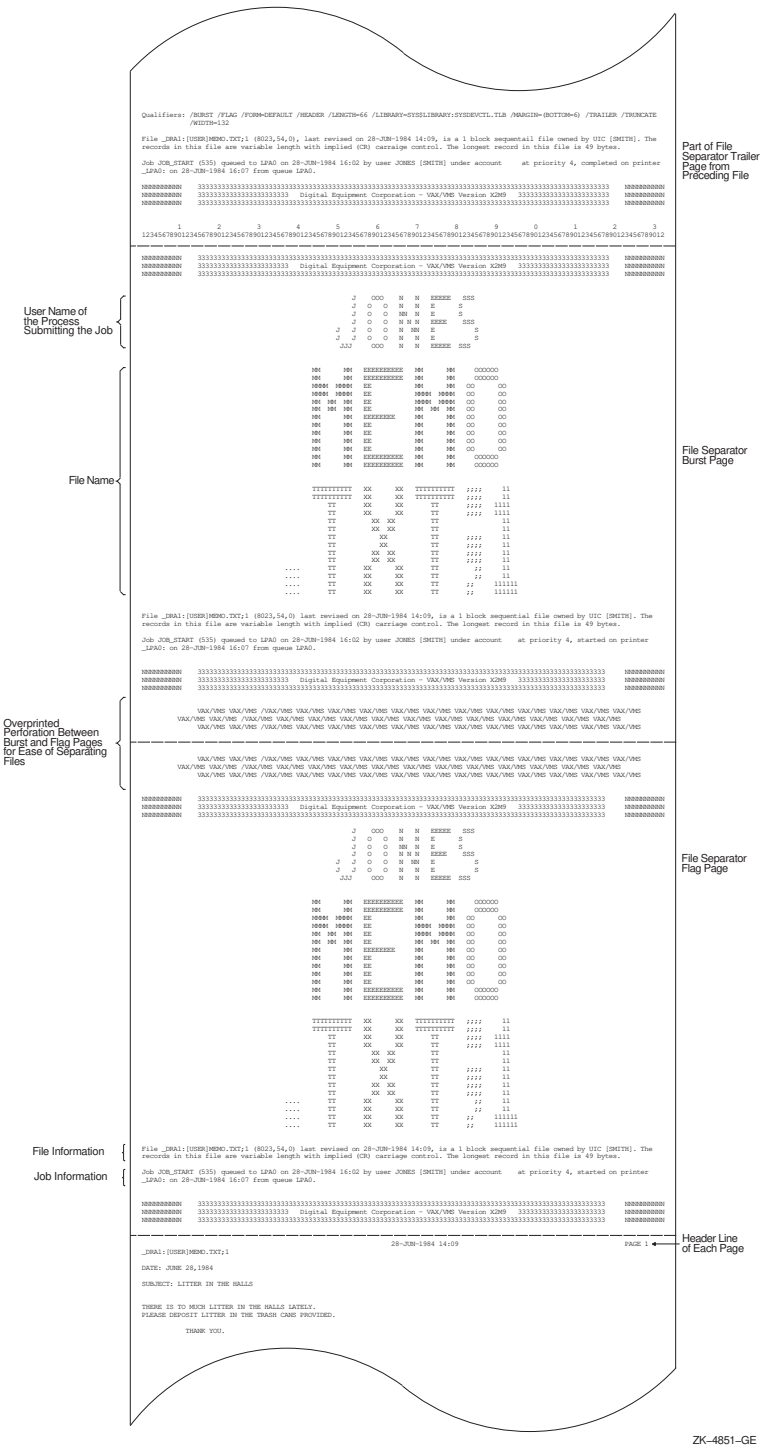
ZK–4852–GE

The job flag page indicates that a new job follows; the job can include one or more files. Information found on a job flag page includes the following:

Item	Description
Header bar	A single-segment bar composed of the following elements: • Rows of a repeated numeral indicating the sequence of the job in the printer queue. • An embedded text string specified by defining the PSM$ANNOUNCE system logical name. The length of the string is limited to one form width. If PSM$ANNOUNCE is not defined, the default text string is ''Digital Equipment Corporation'' followed by the system version number. (''Digital Equipment Corporation'' is abbreviated to ''DEC'' for shorter form widths.)
Note	Users can specify a string of up to 255 characters with the /NOTE qualifier of the DCL command PRINT.
Identification banner	This includes user name of the process submitting the job, the job name, and the job number.
Job sentence	Indicates the following information: • Job name and number • Name of queue to which job was submitted • Submission date and time • Process user name, UIC, and account • Job scheduling priority • Print device name • Job start time • Execution queue name
Footer bar	Identical to the header bar except that the definition of the system logical name (PSM$ANNOUNCE) is not used as the embedded string. The default text is always used as the embedded string.

Figure 13–12 illustrates a file flag and burst page.

Figure 13–12 File Flag and Burst Pages



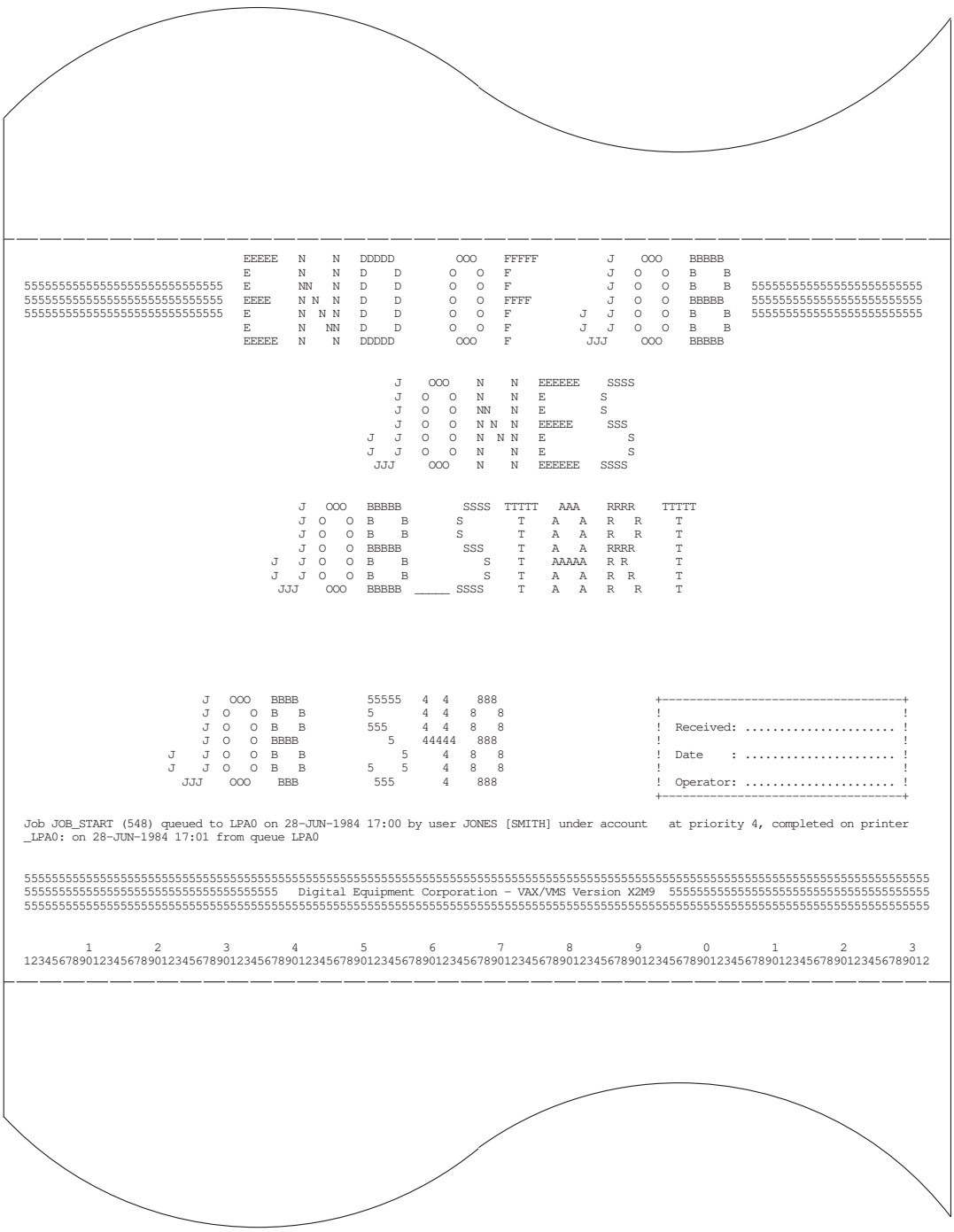
Information found in the file flag page includes the following items:

Item	Description
Header bar	A three-segment bar composed of the following elements: • A central segment identical to the job header bar • Flanking segments with rows of a repeated character indicating the sequence of the file in the job
Note	Users can specify a string up to 255 characters in length using the /NOTE qualifier of the DCL command PRINT.
Identification banner for the process submitting the job	Includes the user name and file name.
File sentence	Indicates the following information: • Full file specification, including device and directory, file name, type, version, and revision time and date • File size in blocks • File organization • File owner's UIC • File record characteristics: record type, carriage control, and size of longest record
Job sentence	Indicates the following information: • Job name and number • Name of queue to which job was submitted • Submission date and time • Process user name, UIC, and account • Job scheduling priority • Print device name • Job start time • Execution queue name
Footer bar	Identical to header bar except that the embedded text string is ''Digital Equipment Corporation'' followed by the operating system version number.

13.8.7.2 Trailer Pages

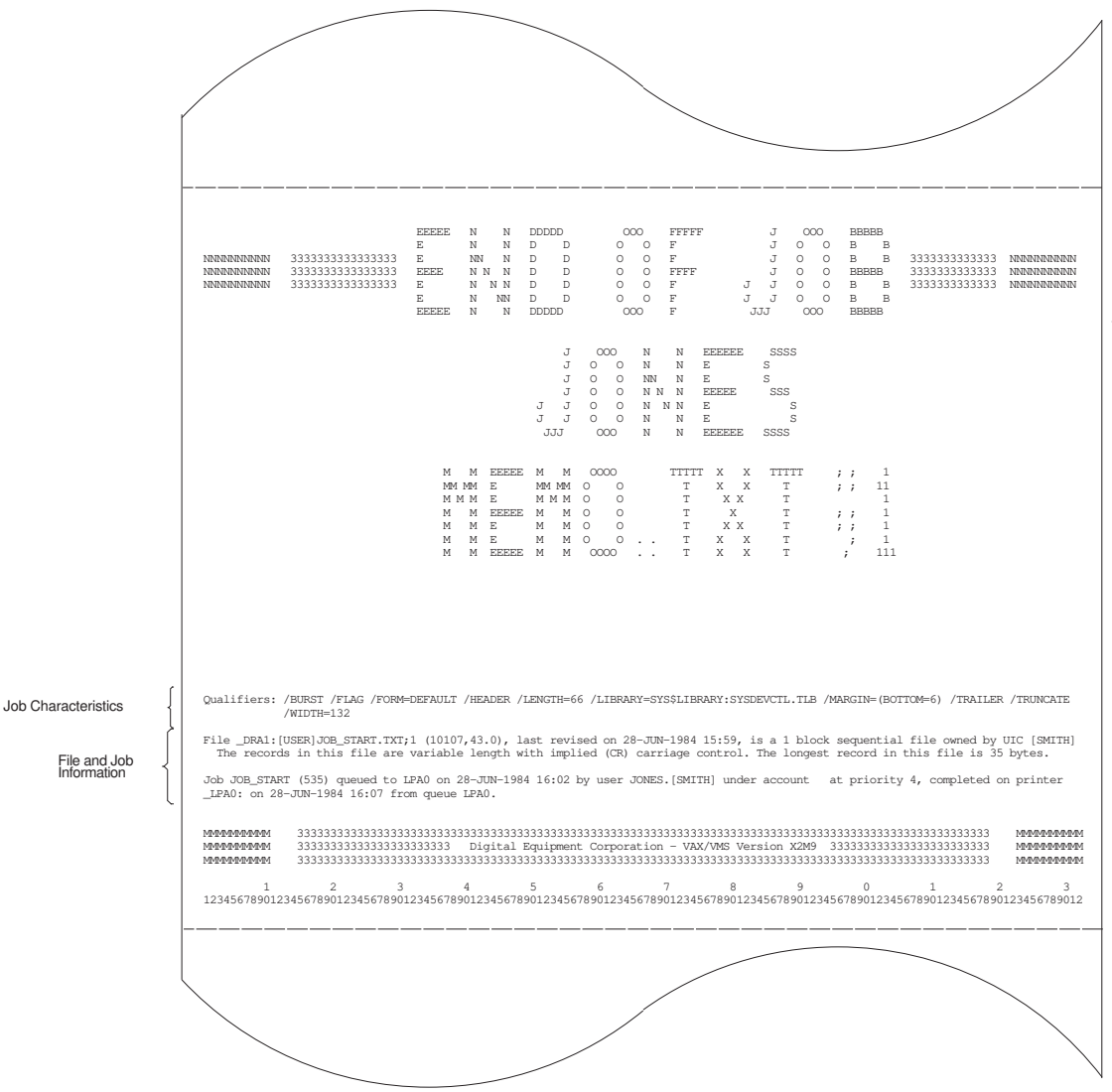
A trailer page indicates the end of a print job or print file. Figure 13–13 illustrates a job trailer page. Figure 13–14 illustrates a file trailer page.

Figure 13–13 Job Trailer Page



ZK–4850–GE

Figure 13–14 File Trailer Page



ZK–4849–GE

The job trailer page indicates the end of a print job. Information found on the job trailer page includes the following:

Item	Description
Header bar	A three-segment bar composed of the following elements: • Central segment with ''END OF JOB'' banner • Flanking segments with job-sequence numeral
Identification banner	For the process submitting the job. Includes job name and job number.
Receipt box	Contains the following signoff fields: Received:, Date:, and Operator:.
Job sentence	Indicates the following information: • Job name and number • Name of queue to which job was submitted • Submission date and time • Process user name, UIC, and account • Job scheduling priority • Print device name • Job start time • Execution queue name
Ruler	A sequence of numbers counting to the end of the form.

The file trailer page includes the following items:

Item	Description
Header bar	A five-segment bar composed of the following elements: – Central segment with ''END OF JOB'' banner – Inner flanking segments with job-sequence numeral – Outer flanking segments with file-sequence character
Identification banner	For the process submitting the job. Includes user name, file name.
Qualifier phrase	Indicates the print, queue, and form qualifiers active when the job was submitted; nonactive qualifiers (except /NORECORD_BLOCKING and /NOFEED) are not included.

Item	Description
File sentence	Indicates the following information: • Full file specification, including device and directory, file name, type, version, and revision time and date • File size in blocks • File organization • File owner's UIC • File record characteristics: record type, carriage control, and size of longest record
Job sentence	Indicates the following information: • Job name and number • Name of queue to which job was submitted • Submission date and time • Process user name, UIC, and account • Job scheduling priority • Print device name • Job start time • Execution queue name
Footer bar	Identical to file flag/file burst footer bar.
Ruler	A sequence of numbers counting to the end of the form.

Widths greater than 40 characters and less than 200 characters and lengths of any size greater than 40 characters are supported for file and job banner pages. Pages requested for widths greater than 200 characters are formatted and printed at 200-character widths. Lengths less than 40 characters are extended by that form length until the 40-character threshold is exceeded. Margins are not taken into account when formatting banner pages.

Note

All banner pages format information to the width and length of the default form size of 80 characters by 51 lines. Therefore, information might be truncated, depending on the form sizes you specify. See Section 13.8.11 for information about controlling line and page overflow.

13.8.8 Specifying Banner Page Options

You specify banner page options by including the following qualifiers with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command:

Page Type	Qualifier
Job pages (to separate jobs)	/SEPARATE=*option* Where *option* can be one or more of the following: [NO]BURST [NO]FLAG [NO]TRAILER
File pages (to separate files within jobs)	/DEFAULT=*option*=*keyword* Where *option* can be one or more of the following: [NO]BURST [NO]FLAG [NO]TRAILER *Keyword* can be either ALL or ONE.

13.8.9 Understanding Forms

Print forms serve the following functions:

- Forms determine certain page formatting attributes (such as margins and page length).
- The paper stock specified in the form definition is used in determining whether a job is eligible to print.

If your printing needs are limited, using special forms is not necessary; Digital supplies a systemwide default form (named DEFAULT) for all queues.

However, if you want to help users format output or if certain print jobs require special paper, you can create one or more forms. Users can then request a form for a print job by specifying the form when they submit a job. If a user does not specify a form, the job uses the queue's **default form**. See Section 13.9.4.4 for information about specifying a default form for a queue.

You can create forms to format output in the following ways:

Formatting	For More Information
Control line overflow by truncating or wrapping lines.	Section 13.8.11
Control page width and length and margin size.	Section 13.9.4.1
Pause output at the end of each page so a new sheet of paper can be inserted (use this for printers that require sheet-feed paper).	Section 13.9.4.1
Specify that a certain paper stock must be physically loaded in the printer.	Section 13.9.4.1
Specify device control library modules to set up the device at the start of each page or each file, or both.	Section 13.9.5.3

The stock of a form is particularly important because it affects whether a job is eligible to print. The stock of a job's form must match the stock of the form currently being used by the queue (the queue's **mounted form**), or the job will

not be scheduled to print; instead, it remains pending in the queue until you perform special steps. (These steps are described in Section 13.10.2.1.)

A queue's mounted form is the form of the current job or, if no job is processing, the form of the last job printed in a queue. As long as the stock of a job's form matches the stock of the queue's mounted form, the job is processed using the options in the job's form, and the mounted form changes automatically to the form of the job being processed in the queue.

13.8.10 Specifying Forms Options

To use forms with your queues, take the following steps:

1. Create a form using the DEFINE/FORM command as explained in Section 13.9.4.1.
2. If you want to change the DEFAULT form, do so *before* creating any queues that reference the DEFAULT form. For more information, see Section 13.9.4.3.
3. Assign a default form for each output execution queue. To assign a default form, specify the /DEFAULT=FORM=*form-name* qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. For more information, see Section 13.9.4.4.

   If you do not assign a default form to a queue, the queue uses the systemwide default form, DEFAULT.
4. Inform users of the available forms and the queues with which they should be used.

For information on creating and managing forms, see Section 13.9.4.

13.8.11 Specifying Options for Controlling Page and Line Overflow

Under certain conditions, lines and pages formatted by the print symbiont might exceed the length of lines and pages for a printer. You can use queue options to control line and page overflow.

13.8.11.1 Controlling Line Overflow with Forms

Digital recommends that you control line overflow by using form definitions. To do this, you must set terminals and printers to avoid wrapping or truncating the print line before the physical limit of the device's width.

Note

The print symbiont uses the form to determine the width of a line. Once the print symbiont has finished formatting the data, if the width of the line exceeds the /WIDTH setting for the device, the device driver will use the /TRUNCATE or /WRAP settings (if set) to truncate or wrap the line.

Different forms can have different right, left, top, and bottom margin settings. By using forms to control page and line overflow, users can switch from one form to another without requiring operators to stop the queue, alter the device setup, and restart the queue. The queue manager automatically mounts any forms with the same stock as the currently mounted form. Operator assistance is needed only to mount a form that has a stock that differs from the stock of the currently mounted form. For more information, see Section 13.10.2.1.

To control line overflow, create forms using the DCL command DEFINE/FORM and specify the following qualifiers:

Qualifier	Function
/[NO]TRUNCATE	Specifies that data in the right margin be discarded. The default is /TRUNCATE. If you specify /TRUNCATE, you cannot specify /WRAP; the /TRUNCATE qualifier forces /NOWRAP.
/[NO]WRAP	Specifies that a carriage return and line feed be inserted when the right margin is exceeded. The default is /NOWRAP. If you specify /WRAP, you cannot specify /TRUNCATE.

For more information on creating and managing forms, see Section 13.8.9.

13.8.11.2 Controlling Page Overflow with the Form-Feed Character

To control page overflow errors, specify the /DEFAULT=[NO]FEED option with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. This option controls whether a form-feed character is automatically inserted when the symbiont detects overflow into the bottom margin area. Users can use the PRINT/FEED or PRINT/NOFEED command to override the default feed option specified for a queue.

Users can specify the /PASSALL qualifier with the PRINT command to bypass all formatting performed by the print symbiont. The default is /NOPASSALL. This qualifier should be used in cases where the print symbiont formatting might interfere with the desired formatting of the file. The /PASSALL qualifier causes the print symbiont to send I/O to the device driver with all formatting suppressed and to behave as follows:

- Not interpret Fortran or print carriage control characters
- Not perform line or page overflow error handling
- Not interpret escape sequences

13.8.12 Suppressing Initial Form Feed

When you start a print queue, a form feed is sent to the output device to ensure that the paper is at the top of the page before printing begins. The initial form feed causes a blank form to be printed when a queue starts.

You can suppress the printing of the blank form by setting the /NO_INITIAL_FF attribute for the queue. To do so, add the /NO_INITIAL_FF qualifier to any of the following DCL commands:

```
INITIALIZE/QUEUE
SET QUEUE
START/QUEUE
```

13.8.13 Understanding Device Control Libraries

A device control library is a text library that contains user-written modules consisting of text or escape sequences. Device control library modules can be used for the following purposes:

- With programmable printers, to insert device-dependent escape sequences that set up a printer for selected print options such as point size, character set, and bold or italic print

- With both programmable and nonprogrammable printers, to insert text at specific points in the processing of a print job

The three types of device control modules, distinguished by their placement in a print job, are:

Module Type	Placement
SETUP modules	Inserted at the beginning of a file.
PAGE_SETUP modules	Inserted at the beginning of each page.
RESET modules	Inserted at the end of each job. Use reset modules to reset a printer to a known state at the end of a job.

13.8.14 Specifying Device Control Library Options

To specify a setup or page setup module for a queue, use forms. Create one or more forms with the DEFINE/FORM command, and specify the /SETUP=*module* or /PAGE_SETUP=*module* qualifier to specify the module to be used. When the form is used with a print job, the module will automatically be sent to the printer.

To specify a reset module, use the /SEPARATE=RESET=*module* qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. The module will be automatically sent to the printer at the end of every job.

To use device control library options, perform the following steps:

1. Create a library and insert modules.
2. Assign the device control library to a queue. (This step is not necessary if you use the default library name SYSDEVCTL.TLB.)
3. Create one or more forms with setup or page setup modules.
4. Assign reset modules to a queue.

For more information on creating and managing device control libraries, see Section 13.9.5.

13.8.15 Understanding the Order of Device Control Module Output

The following list shows the order in which device control modules are sent to the printer within a print job:

1. Reset modules assigned to the queue. (Reset modules are only used at this point for the first job printed after a queue is started.)
2. Setup modules specified in the form definition.
3. Page setup modules specified in the form definition.
4. Setup modules specified with the PRINT command
5. Page 1 of file 1.
6. Page setup modules specified in the form definition.
7. Page 2 of file 1, and so forth.
8. Page setup modules specified in the form definition.
9. Last page of file 1.
10. Setup modules specified in the form definition.
11. Page setup modules specified in the form definition.

12. Setup modules specified with the PRINT command.
13. Page 1 of file 2.
14. Page setup modules specified in the form definition.
15. Page 2 of file 2, and so forth.
16. Page setup modules specified in the form definition.
17. Last page of file 2.
18. Reset modules assigned to the queue.

13.9 Maintaining the Queues

Once you set up your queues, you must monitor and modify them according to the needs of your site. Setting up queues is not restricted to startup time. During normal operation, you can enter the commands to create and start queues as needs dictate.

For example, you might perform the following types of maintenance on your queues:

Maintenance Task	Section
Managing queues	Section 13.9.1
Managing characteristics	Section 13.9.2
Managing forms	Section 13.9.4
Managing device control libraries	Section 13.9.5
Managing jobs in queues	Section 13.9.6
Solving queue problems	Section 13.10

13.9.1 Managing Queues

Once you set up queues, you must monitor, control, and, if necessary, change them. The following sections introduce the basic DCL commands used to perform routine queue management operations.

The following tables list the basic commands for creating and controlling queues. Table 13–2 lists commands that have the same effect on all queues. Table 13–3 lists commands that have a different effect on autostart and nonautostart queues.

Table 13–2 Effects of Queue Commands on All Queues

Command	Description
ASSIGN/MERGE	Moves jobs from one queue to another.
ASSIGN/QUEUE	Redirects jobs in a logical queue to an execution queue.
DELETE/QUEUE	Deletes a queue.
SET QUEUE	Modifies a queue.
SHOW QUEUE	Displays information about a queue.

(continued on next page)

Table 13–2 (Cont.) Effects of Queue Commands on All Queues

Command	Description
STOP/QUEUE	Pauses a queue.
STOP/QUEUES/ON_NODE	Aborts current jobs and stops all queues on a node without stopping the queue manager.

Table 13–3 Effects of Queue Commands on Autostart and Nonautostart Queues

Command	Effect on Autostart Queues	Effect on Nonautostart Queues
DISABLE AUTOSTART/QUEUES	After allowing jobs to complete, fails over all autostart queues to the next available node in each queue's node list. If a queue has no list specified, the queue is stopped.	No effect.
ENABLE AUTOSTART/QUEUES	Starts all stopped, active autostart queues capable of running on the node.	No effect.
INITIALIZE/QUEUE	Creates the queue. The /AUTOSTART_ON qualifier specifies one or more nodes or nodes and devices on which the queue can run.	Creates the queue. The /ON qualifier specifies a single node or node and device on which the queue is to run.
INITIALIZE/QUEUE/START	Creates the queue and activates it for autostart. The /AUTOSTART_ON qualifier specifies one or more nodes or nodes and devices on which the queue can run.	Creates and starts the queue. The /ON qualifier specifies a single node or node and device on which the queue is to run.
START/QUEUE	Activates the queue for autostart.	Starts the queue.
STOP/QUEUE/NEXT	Stops a queue after allowing the current jobs to complete, and deactivates the queue for autostart.	Stops a queue after allowing the current jobs to complete.
STOP/QUEUE/RESET	Stops a queue abruptly and deactivates the queue for autostart.	Stops a queue abruptly.

The following sections describe tasks for managing queues.

Task	For More Information
Initializing Queues	Section 13.9.1.1
Starting Nonautostart Queues	Section 13.9.1.2
Enabling Autostart on a Node to Start Autostart Queues	Section 13.9.1.3
Activating an Autostart Queue	Section 13.9.1.4
Monitoring Queue Information	Section 13.9.1.5
Modifying a Queue	Section 13.9.1.6

Task	For More Information
Pausing a Queue	Section 13.9.1.7
Closing a Queue	Section 13.9.1.8
Stopping a Queue	Section 13.9.1.9
Preventing Autostart Queues from Starting	Section 13.9.1.10
Disabling Autostart on a Node	Section 13.9.1.11
Stopping All Queues on a Node	Section 13.9.1.12
Stopping Queues Before Shutting Down a System	Section 13.9.1.13
Assigning a Logical Queue	Section 13.9.1.14
Moving All Jobs from One Queue to Another	Section 13.9.1.15
Deleting a Queue	Section 13.9.1.16

13.9.1.1 Initializing Queues

Typically, you create your queues when you first set up your system. However, you might want to create queues later as the need arises.

Use the INITIALIZE/QUEUE command to create a queue, as explained in Section 13.7.2.

If you create additional output queues at a later time, make sure you add commands to set device characteristics to the startup command procedure on the node on which the device is located. For devices attached to a LAT port, add commands to the startup procedures on all nodes with queues for the device.

If you create any additional nonautostart queues at a later time, you must also add appropriate START/QUEUE commands to the startup command procedure on the node on which the queue will run. If you do not add appropriate START /QUEUE commands, the queues will not be started when the system reboots.

13.9.1.2 Starting Nonautostart Queues

To start a queue that is not an autostart queue, use the START/QUEUE command as follows:

START/QUEUE[/qualifier,...] queue-name

Queue-name is the name of the queue you want to start. For information about the qualifiers you can specify, see the *OpenVMS DCL Dictionary*.

13.9.1.3 Enabling Autostart on a Node to Start Autostart Queues

You must enable the autostart feature on a node to start autostart queues.

Enabling autostart on a node notifies the queue manager to automatically do the following:

- Start all active and valid autostart queues on the node
- Start any active autostart queue that fails over to the node

To enable autostart on a node, enter the following command:

```
$ ENABLE AUTOSTART/QUEUES
```

By default, the command affects the node from which it is entered. However, you can specify the /ON_NODE qualifier to enable autostart on a different node. For more information, see the *OpenVMS DCL Dictionary* description of the ENABLE AUTOSTART/QUEUES command.

Note

The ENABLE AUTOSTART/QUEUES command starts only valid, *active* autostart queues capable of running on a node. If an autostart queue does not start when you enter this command, the queue might not be active for autostart. You must initially activate autostart queues, as explained in Section 13.9.1.4.

When a node boots, autostart is disabled until you enter the ENABLE AUTOSTART/QUEUES command. Typically, you should add this command to your site-specific startup command procedure or your queue startup command procedure to start a node's autostart queues each time the node boots.

13.9.1.4 Activating an Autostart Queue

You must initially activate an autostart queue in one of the following ways:

- Specify the /START qualifier in the INITIALIZE/QUEUE command used to create the queue, as follows:

  ```
  INITIALIZE/QUEUE/START[/qualifiers,...] queue-name
  ```

- Enter a START/QUEUE command after you create the queue, as follows:

  ```
  START/QUEUE[/qualifiers,...] queue-name
  ```

Once an autostart queue is activated, it remains active unless the queue is stopped with the STOP/QUEUE/NEXT or STOP/QUEUE/RESET command. Shutting down a node does not deactivate autostart queues on the node.

To start an autostart queue that has been deactivated by the STOP/QUEUE /NEXT or STOP/QUEUE/RESET command, enter the START/QUEUE command. The queue will then be automatically started by the queue manager. The queue will be started immediately if a node on which it can run is enabled for autostart. Otherwise, the queue will be started as soon as a node on which it can run is enabled for autostart.

13.9.1.5 Monitoring Queue Information

You use the DCL command SHOW QUEUE to monitor the status of queues. To display queue information, enter the SHOW QUEUE command in the following format:

```
SHOW QUEUE [/qualifier,...] [queue-name]
```

If you do not specify a qualifier or a queue name, the system displays the status of all queues on the system and all jobs owned by you. The SHOW QUEUE qualifiers let you select the type of queue and the amount of information you want to display.

Use the following qualifiers to select the information you want to display:

Qualifier	Description
/BY_JOB_STATUS[=*keyword*,...]	Displays queues that contain jobs of the specified status. You can specify one or more of the following keywords: executing holding pending retained timed_release If no keyword is specified, by default the jobs of all status are displayed. For more information about job status, see Table 13–5.
/BATCH	Displays batch execution queues.
/DEVICE[=*keyword*,...]	Displays output execution queues. You can select a specific type of execution queue by entering one or more of the following keywords: printer terminal server If no keywords are specified, all types of output queue are displayed.
/GENERIC	Displays the status of generic queues.

Use the following qualifiers to select the amount of information you want to display:

Qualifier	Description
/ALL_JOBS	Displays information about all jobs for the selected queues.
/BRIEF	Displays a brief listing of information about job entries in the queue. The brief listing is the default when no qualifier is specified with the SHOW QUEUE command.
/FILES	Adds a list of files associated with each job to the display.
/FULL	Displays complete queue and job information (also displays any ACLs set for the queues).
/SUMMARY	Displays the total number of executing, pending, holding, retained, and timed release jobs. The jobs themselves are not displayed.

You can also combine certain qualifiers to further delineate the queue information you want to display.

Table 13–4 defines queue statuses returned by the SHOW QUEUE command.

Table 13–4 Queue Statuses Displayed in the SHOW QUEUE Command

Queue Status	Description
Aligning	The queue manager is processing a START/QUEUE/ALIGN command.
Autostart inactive	Queue was stopped and needs to be activated. For more information, see Section 13.10.4.

(continued on next page)

Table 13–4 (Cont.) Queue Statuses Displayed in the SHOW QUEUE Command

Queue Status	Description
Available	Queue is processing at least one job but is capable of processing additional concurrent jobs.
Busy	Queue cannot process additional jobs because of one or more jobs in progress.
Closed	Queue is closed and will not accept new jobs until it is set open. For more information, see Section 13.9.1.8.
Device unavailable	Device to which the queue is assigned is not available.
Idle	Queue is not processing any jobs and is capable of doing so.
Paused	A STOP/QUEUE command has been executed.
Pausing	The queue manager is processing a STOP/QUEUE command.
Remote	Queue is assigned to a physical device that is not connected to the local system.
Resuming	The queue manager is processing a START/QUEUE command on a paused queue.
Server	Queue processing is directed to a server symbiont.
Stalled	Symbiont processing temporarily halted due to device-related problem.
Starting	Queue has been started, but the symbiont process is not yet active.
Stopped	The queue is stopped and will not process work until started.
Stop pending	Queue will be in the stopped state when current jobs have finished executing.
Stopping	The queue is being stopped.

To display the forms or characteristics available on a system, use the DCL command SHOW QUEUE/FORM or SHOW QUEUE/CHARACTERISTIC.

You can further customize the type of queue information you want to monitor by writing a command procedure that uses the F$GETQUI lexical function. F$GETQUI invokes the $GETQUI system service to return information stored in the queue database.

You can use the F$GETQUI lexical function to obtain information about the following types of objects:

Characteristics
Forms
Queues
Jobs contained in queues
Files of jobs contained in queues

For example, you could write a command procedure to display the total number of blocks of jobs in a pending state in all printer queues. You must have read access to the job or SYSPRV or OPER privilege to obtain job and file information.

See the *OpenVMS DCL Dictionary* for detailed information about the SHOW QUEUE command and the F$GETQUI lexical function. For more information about the system service invoked by the F$GETQUI lexical function, refer to the description of the $GETQUI system service in the *OpenVMS System Services Reference Manual*.

Examples

1. The following example displays summary information for all printer and terminal queues:

```
$ SHOW QUEUE/SUMMARY/DEVICE=(PRINTER,TERMINAL)

Printer queue HERA_LPA0, busy, on HERA::LPA0, mounted form DEFAULT
  <Printer queue on node HERA for a line printer>

    Job summary:  1 executing

Printer queue HERA_LPB0, busy, on HERA::LPB0, mounted form DEFAULT
  <Printer queue on node HERA for a line printer>

    Job summary:  1 executing

Generic printer queue CLUSTER_PRINT
  <Generic printer queue for LPA0: and LPB0:>
    Job summary:  1 holding

Terminal queue LQ_PRINT, stopped, on HERA::TXA7:,
  <Letter quality printer in Bob's office>
mounted form PORTRAIT_INDENTED (stock=DEFAULT)

    Job summary:  2 pending (445 blocks),  1 holding
```

2. The following example displays the full status and options of all executing jobs:

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

Batch queue HERA_BATCH, available, on HERA::
  /AUTOSTART_ON=(HERA::) /BASE_PRIORITY=3 /JOB_LIMIT=25 /OWNER=[SYSTEM]
  /PROTECTION=(S:E,O:D,G:R,W:W)

  Entry  Jobname         Username             Status
  -----  -------         --------             ------
    700  VUE             SMITH                Executing
         Submitted 25-FEB-1995 14:46 /KEEP /NOLOG /NOPRINT /PRIORITY=100
         File: _$333$DISK1:[SMITH.COM]VUE.COM;19 (executing)


Batch queue ZZ_BATCH, available, on ZZ::
  /AUTOSTART_ON=(ZZ::) /BASE_PRIORITY=3 /JOB_LIMIT=25 /OWNER=[SYSTEM]
  /PROTECTION=(S:E,O:D,G:R,W:W)

  Entry  Jobname         Username             Status
  -----  -------         --------             ------
    874  PIPE            FITZGERALD           Executing
         Submitted 26-FEB-1995 11:25 /KEEP /NOTIFY /NOPRINT /PRIORITY=100
         /RESTART=CLUSTER_BATCH /RETAIN=UNTIL="0 01:00"
         File: _$333$DISK1:[FITZGERALD]PIPE.COM;2 (executing)

Server queue NM$QUE01, available, on HERA::, mounted form DEFAULT
  /BASE_PRIORITY=4 /DEFAULT=(FEED,FORM=DEFAULT) /OWNER=[DOC,SMITH]
  /PROCESSOR=NM$DAEMON /PROTECTION=(S:E,O:D,G:R,W:R) /RETAIN=ERROR

  Entry  Jobname         Username     Blocks  Status
  -----  -------         --------     ------  ------
    236  NM              ROSENBERG        12  Processing
         Submitted 23-FEB-1995 08:42 /FORM=DEFAULT /PRIORITY=100
         File: _$5$DISK3:[FOLK$.NM]NM$J1991072308340647.WRK;1
```

13.9.1.6 Modifying a Queue

You can use the INITIALIZE/QUEUE, START/QUEUE, and SET QUEUE commands to change queue options; as you change queue options, information about the queue in the queue database is updated. You can use the INITIALIZE and START commands only on stopped queues.

The SET QUEUE command lets you change many queue options without having to stop the queue, initialize it, and restart it. For example, the following command modifies the running batch queue, SYS$BATCH:

```
$ SET QUEUE/JOB_LIMIT=4/DISABLE_SWAPPING  SYS$BATCH
```

The command in this example changes the job limit for the queue and disables swapping for all jobs processed in SYS$BATCH. All other options of the queue remain the same. The changed options do not affect the execution of current jobs; however, all subsequent jobs are executed with the new options in effect.

How to Perform This Task

To change queue options that cannot be altered with the SET QUEUE command, use the following procedure:

1. Stop the queue with the STOP/QUEUE/NEXT command.
2. Restart the queue with the START/QUEUE or the INITIALIZE/QUEUE /START command, specifying the appropriate qualifiers for the options you desire.

   Any qualifiers that you do not specify remain as they were when the queue was previously initialized, started, or set.

Note that initializing an existing queue does not delete any current jobs in that queue. Any new queue settings established by the new INITIALIZE/QUEUE command affect all jobs waiting in the queue or subsequently entering the queue.

See Table 13–1 for a list of the options that can be used for batch and output queues.

13.9.1.7 Pausing a Queue

The DCL command STOP/QUEUE (when used without qualifiers) temporarily suspends the execution of all current jobs in the queue and places the queue in a paused state. Pausing an output queue lets you enter print job positioning and alignment commands to the print symbiont. (See Section 13.9.6.9 for more information about using the STOP/QUEUE command to control print jobs.)

To resume the execution of a paused queue, enter the DCL command START /QUEUE.

13.9.1.8 Closing a Queue

When a queue is not available for an extended period of time (for example, when a printer needs servicing), you can prevent new jobs from entering the queue by specifying the /CLOSE qualifier with the SET QUEUE, INITIALIZE/QUEUE, or START/QUEUE command. The /CLOSE qualifier prevents users from entering jobs in the queue with PRINT or SUBMIT commands. When a user attempts to print or submit a job to a closed queue, the job is rejected, and the user is notified that the queue is closed. For example:

```
$ PRINT/QUE=$PRINTER_1 REPORT.TXT;
%PRINT-F-CREJOB, error creating job
-JBC-E-QUE_CLOSED, queue closed, jobs not accepted
```

Jobs currently in the queue are not affected.

When the queue is available again, use the /OPEN qualifier to open the queue for incoming jobs.

13.9.1.9 Stopping a Queue

To stop a queue, enter one of the following commands, according to the desired stop queue operation:

Command	Description
STOP/QUEUE/NEXT	Lets all currently executing jobs complete and then stops the queue. Once you enter this command, all new jobs are prevented from executing.
STOP/QUEUE/RESET	Abruptly stops the queue and returns control to the system. Any jobs that are currently executing are stopped immediately. If the queue is not set to retain jobs completed with an error status, you might want to use the SET QUEUE /RETAIN=ERROR command to do so before stopping the queue with the STOP/QUEUE/RESET command. This causes the queue to retain information about aborted jobs. For print jobs retained on error, you can use the SET ENTRY/RELEASE/NOCHECKPOINT command to restart the interrupted jobs from the beginning. Print jobs are restartable by default; batch jobs are not restartable unless submitted with the /RESTART qualifier.

For autostart queues, these commands deactivate a queue for autostart as explained in Section 13.9.1.10. To restart a stopped nonautostart queue or to reactivate a deactivated autostart queue, enter the START/QUEUE command.

13.9.1.10 Preventing Autostart Queues from Starting

With autostart queues, the STOP/QUEUE/NEXT or STOP/QUEUE/RESET command stops a queue and marks it inactive for autostart until the START /QUEUE command is entered. The STOP/QUEUE/NEXT or STOP/QUEUE /RESET command prevents an autostart queue from being automatically restarted.

You might use this feature to prevent an autostart output queue from accidentally restarting when a printer is being serviced.

13.9.1.11 Disabling Autostart on a Node

The DISABLE AUTOSTART/QUEUES command notifies the queue manager to perform the following tasks on the affected node:

- Prevent autostart queues from failing over to the node.
- Mark all autostart queues on the node as ‘‘stop pending’’ in preparation for a planned shutdown. This lets jobs currently executing on the queues complete.
- Upon completion of any jobs currently executing on one of the node’s autostart queues, force the queue to fail over to the next available node in the queue’s failover list on which autostart is enabled. (An autostart queue can only fail over if you have set it up to run on more than one node.)

The DISABLE AUTOSTART/QUEUES command affects only autostart queues. By default, the command affects the node from which it is entered. However, you can specify the /ON_NODE qualifier to disable autostart on a different node.

The DISABLE AUTOSTART/QUEUES command should be used prior to shutting down a node. For more information, on disabling autostart before shutting down a system, see Section 13.9.1.13.

For more information on the DISABLE AUTOSTART/QUEUES command, see the *OpenVMS DCL Dictionary*.

13.9.1.12 Stopping All Queues on a Node

To stop all queues on a node without stopping the queue manager, enter the DCL command STOP/QUEUES/ON_NODE. By default, this command affects the node on which the command is entered. To stop queues on a different node, specify the name of the node on which queues are to be stopped as follows:

```
STOP/QUEUES/ON_NODE=node-name
```

When you enter the STOP/QUEUES/ON_NODE command, nonautostart queues and autostart queues without a failover list are stopped. Autostart queues created or started with a failover list fail over to the next available node in that list that has autostart enabled. In all cases, currently executing jobs are aborted.

However, you can allow jobs executing on autostart queues to complete by entering the DISABLE AUTOSTART/QUEUES command and waiting for jobs to complete before entering the STOP/QUEUES/ON_NODE command. For more information, see Section 13.9.1.13.

The STOP/QUEUES/ON_NODE command is included in the shutdown command procedure SYS$SYSTEM:SHUTDOWN.COM. For more information, see Section 13.9.1.13.

13.9.1.13 Stopping Queues Before Shutting Down a System

The following commands are included in the shutdown command procedure SYS$SYSTEM:SHUTDOWN.COM and are automatically executed when you shut down a node using SHUTDOWN.COM:

- DISABLE AUTOSTART/QUEUES
- STOP/QUEUES/ON_NODE

Allowing Jobs to Complete Before Stopping Autostart Queues

The STOP/QUEUES/ON_NODE command aborts jobs and stops all queues on a node; the DISABLE AUTOSTART command allows jobs on autostart queues to finish processing before failing over or stopping autostart queues. If your configuration uses autostart queues, you might want to allow jobs on those queues to complete before stopping your queues.

In SHUTDOWN.COM, the STOP/QUEUES/ON_NODE command is executed shortly before the node is shut down. When using SHUTDOWN.COM, you can ensure that jobs on autostart queues have time to complete before the queues are stopped by specifying the time interval between the DISABLE AUTOSTART /QUEUES command and the shutdown.

Use one of the following methods:

Timing	Method
Before executing SHUTDOWN.COM	Define the logical name SHUTDOWN$DISABLE_AUTOSTART to be the number of minutes as follows: DEFINE/SYSTEM/EXECUTIVE_MODE SHUTDOWN$DISABLE_ AUTOSTART number-of-minutes
While executing SHUTDOWN.COM	Specify the number of minutes as a shutdown option as follows: `Shutdown options [NONE]:DISABLE_AUTOSTART=number-of-minutes`

Determine an appropriate number of minutes for your configuration, based on the number and type of jobs in the autostart queues.

If you shut down a node without using SHUTDOWN.COM, you might want to enter the DISABLE AUTOSTART/QUEUES command and wait a few minutes to allow jobs on autostart queues to finish processing before you enter the STOP /QUEUES/ON_NODE command.

13.9.1.14 Assigning a Logical Queue

When a problem occurs with a print device, you can reroute the queue associated with that device to another queue associated with a functioning device. By creating a logical queue out of a malfunctioning execution queue, all of the jobs submitted to that queue will execute in another queue. Use the following procedure to create a logical queue that redirects its jobs to another queue:

1. Stop the queue associated with the malfunctioning print device by entering a command in the following format:

   STOP/QUEUE/NEXT source_queue

   This command inhibits new jobs from processing but lets the current job finish processing, unless the print device is inoperable. If the device is inoperable, use the STOP/QUEUE/RESET command to halt the queue and immediately cancel all output from the device.

2. Take the device off line.

3. Reroute existing jobs from the malfunctioning print device to another print device by entering a command in the following format:

   ASSIGN/QUEUE target_queue source_queue

   Check to be sure that the options of the new print device are appropriate for processing the new jobs.

To deassign the logical queue, enter the following command:

DEASSIGN/QUEUE source_queue

13.9.1.15 Moving All Jobs from One Queue to Another

Before you delete a queue, you might want to requeue all jobs in the queue to another queue. To do so, enter a command in the following format:

```
ASSIGN/MERGE target-queue source-queue
```

Target-queue is the queue to which you are moving the jobs. *Source-queue* is the queue to be deleted.

The ASSIGN/MERGE command moves all jobs currently in the source queue. If new jobs are entered into the source queue before it is deleted, those new jobs remain in the source queue, and are not transferred to the target queue. You might want to close the queue to prevent new jobs from being entered in the queue, as explained in Section 13.9.1.8, before entering the ASSIGN/MERGE command.

For ongoing redirection of jobs, use the ASSIGN/QUEUE command as explained in Section 13.9.1.14.

13.9.1.16 Deleting a Queue

Perform the following steps to delete a queue:

1. Stop the queue by entering the STOP/QUEUE/NEXT command. (Use STOP /QUEUE/RESET if you want to abort all executing jobs.)
2. Wait for executing jobs to complete.
3. Requeue the entries still pending in the queue. If you do not perform this step, jobs will be deleted along with the queue.
4. Remove all references to the queue from generic queues or jobs. See Section 13.10.5 for more information about removing references to queues.
5. Delete the queue by entering the DELETE/QUEUE command.

For more information, see the DELETE/QUEUE command in the *OpenVMS DCL Dictionary*. See Section 13.10.5 for help solving problems deleting a queue.

13.9.2 Managing Characteristics

A characteristic is any attribute of a print or batch job that is relevant to your environment. For example, characteristics for a printer could refer to the color of the ink, the type of paper, or the location of the printer. Once you define the characteristics for a queue, users can specify the characteristics they want to associate with their job when they enter the PRINT command.

For example, you could define the characteristics NORTH, SOUTH, EAST, and WEST on a queue that sends jobs to printers located in four different areas of an office building. Users in the east section of the building could then specify the EAST characteristic to send their jobs to the printer located in their area of the building.

Use the following commands when working with characteristics:

Command	Description
DEFINE/CHARACTERISTIC	Creates a characteristic and assigns a name and number.
DELETE/CHARACTERISTIC	Deletes a characteristic.

Command	Description
SHOW QUEUE/CHARACTERISTIC	Displays information about characteristics defined for the system.
INITIALIZE/QUEUE/CHARACTERISTICS SET QUEUE/CHARACTERISTICS START/QUEUE/CHARACTERISTICS	Specifies one or more characteristics for processing jobs on a queue.
SHOW QUEUE/FULL	Displays information about a queue, including any characteristics assigned to the queue.
PRINT/CHARACTERISTIC SUBMIT/CHARACTERISTIC SET ENTRY/CHARACTERISTIC	Specifies the name or number of one or more characteristics to be associated with the job.

13.9.2.1 Defining Characteristics

No characteristics are defined by default. To define a characteristic, use the DEFINE/CHARACTERISTIC command in the following format:

DEFINE/CHARACTERISTIC characteristic-name characteristic-number

For example:

```
$ DEFINE/CHARACTERISTIC EAST 1
```

You cannot define more than one characteristic name to a number.

If your queue configuration requires you to have more than one characteristic name for a single number, you can define logical names to achieve the same result. For example, you might enter the following commands:

```
$ DEFINE/CHARACTERISTIC SECOND_FLOOR 2
$ DEFINE/SYSTEM/EXECUTIVE_MODE SALES_FLOOR SECOND_FLOOR
$ DEFINE/SYSTEM/EXECUTIVE_MODE SALES_DEPT SECOND_FLOOR
```

In this example, the characteristic name SECOND_FLOOR is assigned to the characteristic number 2. The logical names SALES_FLOOR and SALES_DEPT are defined as equivalent to the characteristic name SECOND_FLOOR. As a result, the logical names SALES_FLOOR or SALES_DEPT are equivalent to the characteristic name SECOND_FLOOR and the characteristic number 2. These logical names can be specified as the *characteristic-name* value for any /CHARACTERISTIC=*characteristic-name* qualifier.

In a VAXcluster environment, you must define the logical names on every node that requires them.

For more information, see the DEFINE/CHARACTERISTIC command in the *OpenVMS DCL Dictionary*.

Note

If you want to define a characteristic name that is also an existing logical name, make sure you read the documentation of logical names in the *OpenVMS User's Manual*.

13.9.2.2 Displaying Characteristics Defined on a System

To see the characteristics defined on a system, enter the SHOW QUEUE /CHARACTERISTIC command. For example:

```
$ SHOW QUEUE/CHARACTERISTIC

Characteristic name                 Number
-------------------                 ------
EAST                                     1
WEST                                     2
NORTH                                    3
SOUTH                                    4
```

13.9.2.3 Assigning Characteristics to a Queue

No characteristics are assigned to a queue by default. To assign characteristics to a queue, include the /CHARACTERISTICS qualifier with the INITIALIZE /QUEUE, START/QUEUE, or SET QUEUE command as follows:

SET QUEUE/CHARACTERISTICS=characteristic[,...] queue-name

For example:

```
$ SET QUEUE/CHARACTERISTICS=(EAST) LN03_1
```

For more information, see the /CHARACTERISTICS qualifier of the INITIALIZE /QUEUE command in the *OpenVMS DCL Dictionary*.

13.9.2.4 Displaying Characteristics Assigned to a Queue

To determine the characteristics defined for a queue, enter the DCL command SHOW QUEUE/FULL. For example:

```
$ SHOW QUEUE/FULL LN03_1
Printer queue LN03_1, idle, on HERA::TTA3, mounted form DEFAULT
  <Printer queue on node HERA for an LN03 printer>
  /BASE_PRIORITY=4 /CHAR=(1) /DEFAULT=(FLAG=ONE,FORM=LN03$PORTRAIT
  (stock=DEFAULT)) /LIBRARY=LN03LIBRARY Lowercase
 /OWNER=[SYSTEM] /PROCESSOR=LATSYM /PROTECTION=(S:E,O:D,G:R,W:W)
 /SEPARATE=(RESET=(ANSI$RESET))
```

13.9.2.5 Canceling Characteristics Assigned to a Queue

To cancel characteristics assigned to a queue, specify the /NOCHARACTERISTICS qualifier with the INITIALIZE/QUEUE, START /QUEUE, or SET QUEUE command. For example:

```
$ SET QUEUE/NOCHARACTERISTICS
```

13.9.2.6 Deleting Characteristics

To delete a characteristic definition, enter the DCL command DELETE /CHARACTERISTIC. For example:

```
$ DELETE/CHARACTERISTIC EAST
```

You must specify the characteristic *name* with the DELETE/CHARACTERISTIC command. If you know the number assigned to the characteristic but do not know the name, enter the SHOW QUEUE/CHARACTERISTIC command to display the names and numbers assigned to characteristics on the system.

For more information about the DELETE/CHARACTERISTIC command, see the *OpenVMS DCL Dictionary*.

If the system displays the following messages, a queue or job refers to the characteristic:

```
%DELETE-E-NOTDELETED, error deleting characteristic
-JBC-E-REFERENCED, existing references prevent deletion
```

You must remove all references to the characteristic before you can delete the characteristic. For information about removing references to a characteristic, see Section 13.10.5.

13.9.3 Managing Banner Pages

For a detailed description of banner pages, see Section 13.8.7. Use the following commands when working with banner pages:

Command	Description
INITIALIZE/QUEUE/SEPARATE=*option* START/QUEUE/SEPARATE=*option* SET QUEUE/SEPARATE=*option*	Specifies one or more of the following job banner page options: [NO]BURST [NO]FLAG [NO]TRAILER The job banner pages you specify for a queue cannot be overridden by the user.
INITIALIZE/QUEUE/DEFAULT=*option=keyword* START/QUEUE/DEFAULT=*option=keyword* SET QUEUE/DEFAULT=*option=keyword*	Specifies one or more of the following file banner page options: [NO]BURST [NO]FLAG [NO]TRAILER *Keyword* can be either ALL or ONE. Users can override the file banner page settings you specify for a queue by specifying the BURST, FLAG, and TRAILER qualifiers with the PRINT command.
PRINT/BURST[=*keyword*]	Specifies the ALL or ONE keyword to override, for the job, the file burst pages specified for the queue.
PRINT/FLAG[=*keyword*]	Specifies the ALL or ONE keyword to override, for the job, the file flag pages specified for the queue.
PRINT/TRAILER[=*keyword*]	Specifies the ALL or ONE keyword to override, for the job, the file trailer pages specified for the queue.

13.9.4 Managing Forms

To use forms with your queues, take the following steps:

1. Define a form. For more information, see Section 13.9.4.1.
2. If desired, assign a default form for each output execution queue. For more information, see Section 13.9.4.4.

If you do not assign a default form to a queue, the queue uses the systemwide default form, DEFAULT. If you want to change the DEFAULT form, you must do so *before* creating any queues that reference the DEFAULT form. For more information, see Section 13.9.4.3.

3. Inform users of the available forms and the queues with which they should be used. In addition, you can create symbols to automatically include the form with the PRINT command as follows:

```
$ PRINT_REPORT :== PRINT/FORM=REPORT
```

4. If you see a print job pending because of a stock mismatch, change the stock of the printer to the requested stock and mount the form associated with the requested stock on the queue, as explained in Section 13.9.4.5, or perform one of the other steps explained in Section 13.10.2.1.

Use the following commands when working with forms:

Command	Description
DEFINE/FORM	Creates a form and assigns a name and number.
SHOW QUEUE/FORM/FULL	Displays information about forms available on a system.
DELETE/FORM	Deletes a form.
INITIALIZE/QUEUE/DEFAULT=FORM START/QUEUE/DEFAULT=FORM SET QUEUE/DEFAULT=FORM	Specifies the name or number of the default form for an output execution queue.
PRINT/FORM SET ENTRY/FORM	Specifies the name or number of the form to be associated with a print job.
INITIALIZE/QUEUE/FORM_MOUNTED START/QUEUE/FORM_MOUNTED SET QUEUE/FORM_MOUNTED	Specifies the name or number of the mounted form for an output execution queue.
SHOW QUEUE/FULL	Displays information about a queue, including the default form for the queue and the form mounted on the queue.

13.9.4.1 Defining a Form

To create a form, enter the DEFINE/FORM command as follows:

DEFINE/FORM form-name form-number [/qualifiers]

You can specify any of the following qualifiers:

Qualifier	Purpose
/WIDTH=*n*	Specifies the physical width of the paper in characters.
/LENGTH=*n*	Specifies the physical length of a form page in lines.
/[NO]TRUNCATE	Specifies that characters exceeding the line length specified by /WIDTH and /MARGIN be discarded. See Section 13.8.11.

Qualifier	Purpose
/[NO]WRAP	Specifies that characters exceeding the line length specified by /WIDTH and /MARGIN be wrapped to the next line. See Section 13.8.11.
/MARGIN=(*option*=*n*[,...])	Specifies the number of blank spaces for one or more of the four margin options: BOTTOM, LEFT, RIGHT, and TOP.
/[NO]PAGE_SETUP=(*module*[,...])	Specifies one or more device control modules that set up a device at the start of each page. See Section 13.9.5.3.
/SETUP=(*module*[,...])	Specifies one or more device control modules that set up the device at the start of each file. See Section 13.9.5.3.
/[NO]SHEET_FEED	Specifies that print jobs pause at the end of every physical page so that a new sheet of paper can be inserted.
/STOCK=*string*	Specifies the type of paper stock to be associated with the form.
/DESCRIPTION=*string*	Specifies a string used to provide information about the form.

If you create forms only to provide different formatting options (and not to specify paper stock), specify the same stock type for each of the different forms. That way, jobs requesting any of these forms will print on the same queue without requiring you to enter any additional commands to modify the queue. Unless you specify the /STOCK qualifier, the form's stock is the same as the name of the form.

Note

If you want to define a form name that is also an existing logical name, make sure you read the documentation of logical names in the *OpenVMS User's Manual*.

For more information about the DEFINE/FORM command, see the *OpenVMS DCL Dictionary*.

Example

The command in the following example defines the form MEMO as the number 3 and defines formatting options for the form:

```
$ DEFINE/FORM MEMO 3/STOCK=DEFAULT -
_$ /MARGIN=(TOP=2,BOTTOM=2,LEFT=6)/WIDTH=80/LENGTH=66/TRUNCATE -
_$ /DESCRIPTION="LN03 indented memo format"
```

13.9.4.2 Displaying Forms Defined on a System

To display forms defined on a system, enter the SHOW QUEUE/FORM/FULL command. If you know the name of the form you want to display, you can specify the form name as a parameter to the command. For example:

```
$ SHOW QUEUE/FORM/FULL MEMO
Form name                           Number   Description
---------                           ------   -----------
MEMO (stock=DEFAULT)                   110   LN03 indented memo format
    /LENGTH=66 /MARGIN=(TOP=2,BOTTOM=2,LEFT=5) /STOCK=DEFAULT /TRUNCATE
    /WIDTH=80
```

For more information, see the SHOW QUEUE/FORM command in the *OpenVMS DCL Dictionary*.

13.9.4.3 Changing the Systemwide Default Form

The operating system includes a systemwide default form, named DEFAULT, that corresponds to the form number 0 and has a stock named DEFAULT. A queue initialized without the /DEFAULT=FORM qualifier uses the systemwide default form to process print jobs not explicitly associated with a form definition. The systemwide DEFAULT form uses the following options:

```
Form name                    Number      Description
---------                    ------      -----------
DEFAULT                          0       System-defined default
    /MARGIN=(BOTTOM=6) /STOCK=DEFAULT /TRUNCATE /WIDTH=132
    /LENGTH=66
```

To change the systemwide default form, enter the DEFINE/FORM command in the following format:

```
DEFINE/FORM DEFAULT 0 /qualifier[s]
```

For example, to change the systemwide default form's bottom margin from 6 to 4, and the page length from 66 to 55, enter this command:

```
$ DEFINE/FORM DEFAULT 0/MARGIN=(BOTTOM=4)/LENGTH=55
```

Note

Once a queue or job references a form, you cannot change the stock for that form. Therefore, if you want to change the stock of the DEFAULT form, you should do so before any queues are created.

13.9.4.4 Assigning a Default Form for a Queue

If a user does not specify the /FORM qualifier when submitting a job with the PRINT command, the job uses the default form for the execution queue on which the job is printed.

To assign a default form for an output execution queue, specify the /DEFAULT qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command as follows:

```
INITIALIZE/QUEUE/DEFAULT=FORM=form-name-or-number
```

Note

The queue's default form is associated with a print job at the time the job is processed unless a specific form is requested by the user when the job is submitted. Therefore, if a user submits a job to a generic queue without specifying the /FORM qualifier, no form is associated with the job until it is transferred to an execution queue.

If you do not establish a default form for a queue, the queue uses the systemwide default form, DEFAULT.

13.9.4.5 Mounting a Form on a Queue

Mounting a form on a queue associates the stock of the form with the queue. The stock of the form mounted on a queue should match the stock of the paper physically loaded in the queue's printer. A queue's mounted form affects the scheduling of all jobs in the queue. If the stock of a job does not match the stock of the form currently mounted, the job is held pending because of stock mismatch and is not scheduled. When this happens, you can mount a different form on the queue by using the /FORM_MOUNTED qualifier with the SET QUEUE command.

You mount a form on a queue by specifying the /FORM_MOUNTED qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command as follows:

```
INITIALIZE/QUEUE/FORM_MOUNTED=form-name-or-number
```

See Section 13.10.2.1 for other steps you can take to fix jobs that are pending because of stock mismatch. See the *OpenVMS DCL Dictionary* for more information about the /FORM_MOUNTED qualifier.

13.9.4.6 Displaying the Form Assigned to a Queue

To find the default form for a queue, enter the DCL command SHOW QUEUE /FULL. For example:

```
$ SHOW QUEUE/FULL JEAN_PRINT
Printer queue JEAN_PRINT, idle, on BAY::TTA3:, mounted form 8_5x11
  <Queue for printer in Jean's office>
  /BASE_PRIORITY=4 /DEFAULT=(FEED,FORM=REPORT (stock=8_5X11)) /OWNER=[SYSTEM]
  /PROTECTION=(S:E,O:D,G:R,W:W)
```

In this example, the default form is REPORT and its stock is 8_5x11. All jobs processed on this queue that are not associated with an explicit form definition in the PRINT command have the default form REPORT. As long as the stock of the mounted form matches the stock of the default form, all jobs submitted to this queue without an explicit form definition will be scheduled to print.

13.9.4.7 Deleting a Form

To delete a defined form, enter the DCL command DELETE/FORM. For example:

```
$ DELETE/FORM MEMO
```

You must specify the form *name* with the DELETE/FORM command (not the form *number*). If you know the number assigned to the form but do not know the name, enter the SHOW QUEUE/FORM command to display the names and numbers assigned to forms on the system.

For information about the DELETE/FORM command, see the *OpenVMS DCL Dictionary*.

If the system displays the following messages, a queue or job exists with a reference to the form:

```
%DELETE-E-NOTDELETED, error deleting form-name
-JBC-E-REFERENCED, existing references prevent deletion
```

You must remove all references to the form before you can delete the form. For information about removing references to a form, see Section 13.10.5.

13.9.5 Managing Device Control Libraries

To use a device control library module, perform the following steps:

1. Create a library and insert modules. For more information, see Section 13.9.5.1.
2. Assign the device control library to a queue. (This step is not necessary if you use the default library SYSDEVCTL.TLB.) For more information, see Section 13.9.5.2.
3. Create one or more forms with setup or page setup modules. This lets users request the form (and the accompanying setup or page setup modules) when they enter the PRINT/FORM command. For more information, see Section 13.9.5.3.
4. Assign a reset module to a queue. This specifies the reset module to follow each job printed on a queue. For more information, see Section 13.9.5.4.

Use the following DCL commands to set up device control library modules for processing print jobs:

Command	Description
DEFINE/FORM/SETUP=*module*[,...]	Specifies one or more modules that set up the device before each file of a job that is printed when the form is mounted.
DEFINE/FORM/[NO]PAGE_SETUP=*module*[,...]	Specifies one or more modules that set up the device before each page of a job that is printed when the form is mounted.
INITIALIZE/QUEUE/LIBRARY=*filename* START/QUEUE/LIBRARY=*filename*	Specifies the file name of the device control library. The default library is SYS$LIBRARY:SYSDEVCTL.TLB. The value specified as *filename* must be a file name only, without a type or a directory. The file must be found in the SYS$LIBRARY directory.
INITIALIZE/QUEUE/SEPARATE=[NO]RESET=*module*[,...] START/QUEUE/SEPARATE=[NO]RESET=*module*[,...] SET QUEUE/SEPARATE=[NO]RESET=*module*[,...]	Assigns one or more modules to the queue. This causes the modules to be extracted from the device control library and copied to the device at the end of each job. The module can be used to restore the device to a known state at the end of each job. The default is NORESET.
PRINT/FORM=*form-name*	Calls for a print job to be processed according to the specified form. You can create a form to contain setup or page setup modules to be to be extracted from the device control library and copied to the printer at the start of each file or page.

The following sections describe how to manage device control libraries.

13.9.5.1 Creating a Device Control Library and Inserting Modules

To create a device control library and insert modules, perform the following steps:

1. Create a device control library by entering a command in the following format:

   ```
   LIBRARY/CREATE/TEXT SYS$COMMON:[SYSLIB] filename.TLB
   ```

2. Determine the contents of the module—either the text to be inserted or the escape sequences needed for the desired printer setup. To determine the proper escape sequences for a printer option, refer to the operation guide for the specific printer.
3. Create a module file and enter the appropriate escape sequences, carriage control characters, or text. You create and edit a module file as you would any other text file. Make sure the text editor you use does not insert a carriage return or line feed at the end of your file. This will affect your printer output.
4. Insert the module into the device control library by entering the following command:

   LIBRARY/INSERT/TEXT library-file module-file

Note

To add a module to or delete it from a library, you must stop all output queues assigned to that library.

See the *OpenVMS Command Definition, Librarian, and Message Utilities Manual* for more information about creating libraries and inserting modules.

13.9.5.2 Assigning a Library to a Queue

You assign a device control library to an output queue by specifying the /LIBRARY qualifier with the INITIALIZE/QUEUE or START/QUEUE command in the following format:

INITIALIZE/QUEUE/START/LIBRARY=filename queue-name

The file name is the name of the library file that contains the desired modules. Libraries must be in SYS$LIBRARY and must have the file type .TLB. If you do not specify the /LIBRARY qualifier, the default library is SYS$LIBRARY:SYSDEVCTL.TLB. You can use the /LIBRARY qualifier to specify an alternate device control library. For example:

```
$ INITIALIZE/QUEUE/LIBRARY=LN03DEVCTL LN03_A_QUE
```

Note

If you specify a value for the /LIBRARY qualifier, do *not* include the directory, file type, or version number. The system assumes that the file is in SYS$LIBRARY and has the type .TLB. If you copy a library file from another node, be sure that the new library has a unique file name.

Operations that request a particular device control library module use the module from the library specified for the queue. If you have a small configuration of printers and normally use only a few modules, you usually store all modules in a single library and assign that same library to each printer queue.

For sites with a large number of different printers, you usually create and assign a separate device control library for each type of printer. If you have a separate device control library for each type of printer, Digital recommends you give modules that perform the same function an identical name in all libraries, even though the modules contain escape sequences unique to the specific type of printer. For example, if three libraries contained modules that set up a printer for landscape orientation, these modules might be named LANDSCAPE in all

three libraries. This allows you to use the same form on any queue for which a library contains a module of the specified name, even though the modules might contain different device-specific sequences.

If you use a single library to store modules for different types of printers, make sure that each module has a unique name.

Use the following command format to display a listing of all modules contained within a specified library:

LIBRARY/LIST/FULL SYS$LIBRARY:library-name.TLB

13.9.5.3 Creating Forms for Setup and Page Setup Modules

To use setup or page setup modules with print jobs, Digital recommends you create forms that include these modules. Use DEFINE/FORM command with the /SETUP=*module* or /PAGE_SETUP=*module* qualifiers. The modules you specify with the /SETUP qualifier are sent to the printer when the form is mounted, before each file of a job is printed. Similarly, the modules you specify with /PAGE_SETUP are sent to the printer before each page of a job.

Once you have properly created a form that includes a setup or page setup module, users can request the module by including the /FORM qualifier with the PRINT command as follows:

PRINT/FORM=form-name file-name /QUEUE=queue-name

Users can also request a setup module by specifying the /SETUP=*module* qualifier with the PRINT command. Note, however, that the PRINT/FORM command has an advantage over the PRINT/SETUP command: When a user enters the PRINT /FORM command, the form name is checked for validity when the command is entered. With the PRINT/SETUP command, however, the module names are not checked until the job attempts to print. If a user makes an error while entering a module name, the job will not print, and the user will not be notified unless he or she also specified the /NOTIFY qualifier.

13.9.5.4 Assigning a Reset Module to a Queue

To assign a module to an output execution queue to reset the printer to a known state at the end of each job, you use the /SEPARATE=RESET=*module* qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. For example:

```
$ INITIALIZE/QUEUE/LIBRARY=MYDEVCTL/SEPARATE=RESET=MODULE2 PDQ_QUE
```

The reset sequence contained in the module resets the printer at the end of each job. It also resets the printer when the queue is started to ensure that the first job prints correctly.

Because the /SEPARATE qualifier specifies mandatory queue options, the RESET module you specify is sent to the queue at the end of every job. The user cannot override this option.

Example

The following example uses device control library modules to process a print job.

```
$ LIBRARY/CREATE/TEXT SYS$LIBRARY:MYDEVCTL.TLB
$ EDIT MODULE1.TXT

    !enter printer escape sequences or text for module1

$ EDIT MODULE2.TXT

    !enter printer escape sequences or text for module2
```

```
$ LIBRARY/INSERT SYS$LIBRARY:MYDEVCTL.TLB/TEXT MODULE1
$ LIBRARY/INSERT SYS$LIBRARY:MYDEVCTL.TLB/TEXT MODULE2
$ INITIALIZE/QUEUE/START/ON=TTA9:/LIBRARY=MYDEVCTL PDQ_QUE
$ SET QUEUE/SEPARATE=RESET=MODULE2 PDQ_QUE
$ SHOW QUEUE/FULL PDQ_QUE

Terminal queue PDQ_QUE, idle on TOAD::TTA9, mounted form DEFAULT
   /BASE_PRIORITY=4/DEFAULT=(FEED,FORM=DEFAULT)/LIBRARY=MYDEVCTL
   /OWNER=[1,4]/PROTECTION=(S:E,O:D,G:R,W:W)/SEPARATE=(RESET=(MODULE2))

$ DEFINE/FORM/SETUP=MODULE1/STOCK=DEFAULT FORM1 1

$ PRINT/FORM=FORM1 REPORT.TXT,MEMO.TXT/QUEUE=PDQ_QUE

Job REPORT (Queue PDQ_QUE, entry 619) started on PDQ_QUE
```

In this example, two device control modules are created and inserted into the library file MYDEVCTL.TLB. The escape sequence or text in the setup module named MODULE1 is sent to the printer to set up the printer before REPORT.TXT is printed and again before MEMO.TXT is printed. The escape sequence or text in the reset module named MODULE2 is sent to the printer only once after both of the files in job REPORT have printed.

13.9.6 Managing Jobs

As system manager, you are responsible for controlling the flow of batch and print jobs and for maintaining efficient job processing performance. Some of the routine tasks for managing jobs include the following:

Task	Section
Monitoring jobs	Section 13.9.6.1
Modifying job processing options	Section 13.9.6.2
Holding and releasing a job	Section 13.9.6.3
Changing the scheduling priority of a job	Section 13.9.6.4
Requeuing an executing job	Section 13.9.6.5
Requeuing a pending job	Section 13.9.6.6
Retaining jobs in a queue	Section 13.9.6.7
Deleting a job	Section 13.9.6.8
Specifying the position at which a job should begin printing	Section 13.9.6.9.1
Aligning print forms	Section 13.9.6.9.2

13.9.6.1 Monitoring Jobs

You use the DCL command SHOW ENTRY to monitor the status of batch and print jobs. (For information about job status, see Table 13–5.) Use the following format to specify the SHOW ENTRY command:

SHOW ENTRY [entry-number[,...]], [job-name[,...]]

If you do not specify an entry number or job name, the system displays all jobs owned by you or by the user specified with the /USER_NAME qualifier. If you specify a job name, the system displays all jobs owned by you or by the user specified with /USER_NAME that match the specified character string. You can also display a group of jobs by entering a list of entry numbers or job names, or both, on the command line.

Specify qualifiers with the SHOW ENTRY command to specify the type of job information you want to display. For more information, see the *OpenVMS DCL Dictionary*.

Table 13–5 describes the job statuses returned by the SHOW ENTRY command.

Table 13–5 Job Statuses Returned by SHOW ENTRY

Status	Description
Aborting	Executing job is terminating abnormally.
Executing	Job is executing from a batch queue.
Holding	Job is being held until explicitly released. For more information, see Section 13.9.6.3.
Holding until	Job is being held until a specified time. For more information, see Section 13.9.6.3.
Pending	Job is waiting to execute. For more information, see Section 13.10.2.
Printing	Job is executing from a printer or terminal queue.
Processing	Job is executing from a server queue.
Retained on Completion	Job remains in the queue upon completion. For more information, see Section 13.9.6.7.
Retained on Error	Job remains in the queue upon encountering an error. For more information, see Section 13.9.6.7.
Retained Until	Job remains in the queue until the specified time. For more information, see Section 13.9.6.7.
Stalled	Job is stalled because the physical device on which the queue is located is stalled.
Waiting	Symbiont refuses the job.

Examples

1. The following command displays jobs owned by user GARDNER:

```
$ SHOW ENTRY/USER_NAME=GARDNER

  Entry  Jobname         Username     Blocks  Status
  -----  -------         --------     ------  ------
      4  TEST            GARDNER              Holding
         On available batch queue OPAL_BATCH

    611  SET             GARDNER         140  Pending
         On stopped printer queue LQPRINT
```

2. In the following example, the /FULL qualifier displays job status information, the time the job was submitted, the file specification, and the job processing options:

```
$ SHOW ENTRY/FULL 4,611

  Entry  Jobname         Username     Blocks  Status
  -----  -------         --------     ------  ------
      4  TEST            GARDNER              Holding
         On available batch queue OPAL_BATCH
         Submitted 15-JAN-1995 16:12 /LOG=_$5$DUA1:[GARDNER]TEST.LOG;
         /PRIORITY=100
         File: _$5$DUA1:[GARDNER]TEST.COM;8

    611  SET             GARDNER         140  Pending (queue stopped)
         On stopped printer queue LQPRINT
         Submitted 21-JAN-1995 16:23 /FORM=DEFAULT /PRIORITY=200
         File: _$5$DUA1:[GARDNER]SET.TXT;5
         File: _$5$DUA1:[GARDNER]WAIT.TXT;1
```

13.9.6.2 Modifying Job Processing Options

You can modify many job processing options by specifying qualifiers with a command in the following format:

```
SET ENTRY/qualifier[,...] entry-number
```

Table 13–6 lists some qualifiers that are frequently used to change jobs. For a list of all the job processing options you can change with the SET ENTRY command, see the *OpenVMS DCL Dictionary*.

Table 13–6 SET ENTRY Qualifiers for Changing Jobs

Qualifier	Description
/[NO]AFTER=*time*	Controls whether a job is held until after a specified time. For more information, see Section 13.9.6.3.
/CHARACTERISTICS=(*characteristic*[,...])	Specifies the name or number of one or more characteristics associated with a batch or print job. For more information, see Section 13.8.3.
/FORM=*form-name-or-number*	Specifies the name or number of the form to be associated with a print job. For more information, see Section 13.8.9.
/[NO]HOLD	Controls whether a job is available for immediate processing or held until it is released for processing. For more information, see Section 13.9.6.3.
/PRIORITY=*n*	Specifies the scheduling priority of the job. For more information, see Section 13.9.6.4.
/RELEASE	Releases a previously held job. For more information, see Section 13.9.6.3.
/REQUEUE=*queue-name*[:]	Requests that the job be moved from the original queue to the specified queue; this qualifier can also be used with the STOP/QUEUE/ENTRY command. For more information, see Section 13.9.6.6.
/RESTART	Specifies whether a batch or print job is restarted after a system failure or a STOP/QUEUE/REQUEUE command. Print jobs are restartable by default. Batch jobs are restartable only if submitted or modified with the /RESTART qualifier.

13.9.6.3 Holding and Releasing a Job

Users can specify that a job be held in a queue before processing by specifying one of the following qualifiers with the PRINT, SUBMIT, or SET ENTRY command:

- /AFTER=*time-value* holds a job until the specified time.
- /HOLD holds a job indefinitely.

You can use the following commands to hold and release jobs:

Command	Purpose
SET ENTRY/HOLD	Holds a job in a queue indefinitely before processing.
SET ENTRY/AFTER=*time-value*	Holds a job in a queue for processing after a specified time.
SET ENTRY/NOHOLD	Releases a job that is held in a queue for any of the following reasons: • Releases a job from Holding status. • Releases a job from Retained state (a job held in a queue by the /RETAIN qualifier). For more information, see Section 13.8.2 and Section 13.9.6.7. • Releases a job that was refused by a user-written symbiont.
SET ENTRY/NOAFTER	Releases a job with from a status of ''Holding until *time-value*,'' and does not wait until the specified time to schedule the job.
SET ENTRY/RELEASE	Releases a job that is held in a queue for any of the following reasons: • Releases a job from Holding status. • Releases a job from a status of ''Holding until *time-value*,'' and does not wait until the specified time to schedule the job. • Releases a job from a Retained state (a job held in a queue by the /RETAIN qualifier). For more information, see Section 13.8.2 and Section 13.9.6.7. • Releases a job that was refused by a user-written symbiont.

Examples

1. The following example holds a job until the specified time and subsequently releases the job before that time:

```
$ SET ENTRY 1121/AFTER=12-FEB-1995:17:30
$ SET ENTRY/NOAFTER
```

2. The following example holds a job until the end of the current day (00:00:00 0 o'clock) and subsequently releases the job before that time:

```
$ SET ENTRY 1121/AFTER=TODAY
$ SET ENTRY/NOAFTER
```

3. The following example holds a job indefinitely and subsequently releases it:

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

13.9.6.4 Changing the Scheduling Priority of a Job

The queue manager uses the following criteria to determine the scheduling order for batch and print jobs that are eligible for processing:

1. Job scheduling priority

   The queue manager checks the job's scheduling priority. The job with the highest scheduling priority value is processed first. The job scheduling priority is different than the base process priority or current process priority. The user can specify job scheduling priority with the /PRIORITY qualifier of the PRINT or SUBMIT command.

2. Size (optional, and applies only to output jobs)

   By default, the job size of an output job is checked. Among jobs of identical scheduling priority, the smallest sized job is processed first.

   The job size is not checked if the queue has been created, started, or modified to use the /SCHEDULE=NOSIZE option. For more information, see Section 13.8.6.

3. Submission time

   If the jobs' scheduling priorities are identical, the job that was submitted first is processed first.

If a batch or print job cannot be processed, it is placed in a pending state and is not processed until the cause of the pending state is resolved. For more information, see Section 13.10.2. You can change the scheduling priority of a job by using the /PRIORITY qualifier with the SET ENTRY command as follows:

```
SET ENTRY/PRIORITY=n entry-number
```

Do not confuse the job scheduling priority with the process base priority on a queue. The job scheduling priority value must be in a range of 0 through 255, where 0 is the lowest priority and 255 is the highest. The default value for /PRIORITY is the value of the system parameter DEFQUEPRI (usually set at 100). The following example changes the scheduling priority of a job to 50:

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

13.9.6.5 Requeuing an Executing Job

To stop and requeue an executing print job, enter the STOP/QUEUE/REQUEUE command. This command suspends a currently executing job and requeues it to the specified queue. Other jobs remain pending in the queue until they are processed.

Note

The STOP/QUEUE/REQUEUE command stops only the job currently executing in the queue. The queue is not stopped.

For example, suppose a job is executing in output execution queue BETA_LPB0 when the printer on which the queue is running jams. If no other jobs are pending in the queue, you might want to stop and requeue the job to a queue running on another printer. Because the printer in this example is jammed, you might also want to stop the queue. To do so, you would enter commands similar to the following:

```
$ STOP/QUEUE/REQUEUE=BETA_LPA0 BETA_LPB0
$ STOP/QUEUE/RESET BETA_LPB0
```

The first command causes the executing print job on BETA_LPB0 to be stopped and requeued to BETA_LPA0. The second command stops queue BETA_LPB0.

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

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

To hold an aborted job, specify the /HOLD qualifier as follows:

STOP/QUEUE/REQUEUE[=queue-name]/HOLD[/ENTRY=entry-number] queue-name

When you specify /HOLD, the aborted job is placed in a hold state for later release with the SET ENTRY/RELEASE or SET ENTRY/NOHOLD command.

To change the scheduling priority of the aborted job, specify the /PRIORITY qualifier as follows:

STOP/QUEUE/REQUEUE[=queue-name]/PRIORITY=n[/ENTRY=entry-number] queue-name

Specify the new priority as *n*.

For more information about the STOP/QUEUE/REQUEUE command, see the *OpenVMS DCL Dictionary*.

13.9.6.6 Requeuing a Pending Job

To requeue a job that is pending in a queue, enter the SET ENTRY/REQUEUE command. For example:

```
$ SET ENTRY/REQUEUE=LN03$PRINT 196
```

This command moves job 196 to the queue LN03$PRINT.

For more information about the SET ENTRY/REQUEUE command, see the *OpenVMS DCL Dictionary*.

13.9.6.7 Retaining Jobs in a Queue

Users can specify a retention option for a batch or print job by specifying the /RETAIN=*option* qualifier with the SUBMIT or PRINT command.

To change the user-specified retention policy for a job, specify the /RETAIN=*option* qualifier with the SET ENTRY command in the following format:

SET ENTRY/RETAIN=option entry-number

Option is one of the following:

Option	Description
ALWAYS	Holds the job in the queue regardless of the job's completion status.
DEFAULT	Holds the job in the queue as specified by the queue's retention policy. For more information, see Section 13.8.2. If no policy has been set on the queue, the job is not retained.
ERROR	Holds the job in the queue only if the job completes unsuccessfully.

Option	Description
UNTIL=*time-value*	Holds the job in the queue for a specified length of time, regardless of the job's completion status. This lets you retain the job in the queue only as long as the job is needed and eliminates the need to delete the job from the queue later. The time value you specify is interpreted first as a delta time, then as a combination time, and finally as an absolute time. For information about specifying time values, see the *OpenVMS User's Manual*.

For example, the following command retains job 172 in the queue until 3 hours after the job completes. At that time, the job will automatically be deleted from the queue:

```
$ SET ENTRY/RETAIN=UNTIL="+3:00" 172
```

For more information about user-specified job retention, see the /RETAIN qualifier for the PRINT, SUBMIT, or SET ENTRY command in the *OpenVMS DCL Dictionary*.

You can set a job retention policy on a queue by specifying the /RETAIN qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command. For information, see Section 13.8.2.

13.9.6.8 Deleting a Job

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

Follow this procedure to delete either a pending or an executing job:

1. Determine the entry number of the job by entering one of the following DCL commands:

   SHOW ENTRY/USER_NAME=username [job-name]

   SHOW QUEUE/ALL [queue-name]

   If you do not know the job name, user name, or queue name, enter the following command:

   SHOW QUEUE/BATCH/ALL/BY_JOB_STATUS=EXECUTING

2. Delete the job by entering a command in the following format:

   DELETE/ENTRY=(entry-number)[,...]

Example

Suppose a user has noticed that a job is processing in an endless loop. The user is not the owner of the job and lacks sufficient privilege to stop it. The user enlists your aid as the system manager. You might enter the following command:

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

Batch queue JADE_BATCH, available, on JADE::

  Entry  Jobname         Username             Status
  -----  -------         --------             ------
    312  ARTWORK         HUNTER               Executing

Batch queue OPAL_BATCH, available, on OPAL::
```

```
  Entry  Jobname         Username             Status
  -----  -------         --------             ------
    317  STOCKS          CHANDLER             Executing

Batch queue RUBY_BATCH, available, on RUBY::

  Entry  Jobname         Username             Status
  -----  -------         --------             ------
    888  TEMPO           ENGLISH              Executing

$ DELETE/ENTRY=317
```

13.9.6.9 Pausing an Output Queue to Control Print Job Position and Alignment

Pausing an output queue lets you communicate with the print symbiont interactively. You enter the STOP/QUEUE command (without any qualifiers) to pause a queue. Once a queue is paused, you can perform operations such as the following:

- Specify the position at which a suspended job is to resume printing. For example, suppose a printer has a paper jam and the first several pages of a print job are destroyed. You can pause the queue and restart the job, resuming printing at the beginning of the file.

  For more information, see Section 13.9.6.9.1.

- Specify the number of pages and the type of data for aligning printer forms. For example, suppose a printer uses a paper stock that is a preprinted continuous-form paper. When you begin printing a job, you notice the paper is not correctly aligned, so output does not print in the correct space on the preprinted form. You could pause the queue and print sample data to help you correct the paper alignment.

  For more information, see Section 13.9.6.9.2.

Note

In order to perform these tasks, you must pause the queue with the STOP /QUEUE command *after* the job has begun printing.

13.9.6.9.1 Specifying the Position of Print By default, when you pause a queue and restart it, printing resumes in the current job at a checkpoint near where it left off. To specify the position at which the current job is to resume printing, pause the queue, then enter the START/QUEUE command with any of the following qualifiers:

Qualifier	Description
/[NO]BACKWARD[=*pages*]	Specifies the number of pages the file is to be backspaced before printing is resumed. A value of 1 indicates that printing is to resume at the top of the current page. If you omit the page value, the file is backspaced one page.
/[NO]FORWARD[=*pages*]	Specifies the number of pages the file is to be advanced before printing is resumed. A value of 1 indicates that printing is to resume at the top of the next page. If you omit the page value, the file is advanced one page.

Qualifier	Description
/[NO]SEARCH=*string*	Specifies that printing is to resume at the page containing a particular search string. The search for the string moves forward, beginning on the page following the current page. During the search, consecutive tabs and spaces are treated as a single space, and character case is ignored.
/TOP_OF_FILE	Specifies that printing is to resume at the beginning of the file.

When you must use more than one positioning qualifier with the same START /QUEUE command, file positioning is performed in the following order:

1. /TOP_OF_FILE
2. /FORWARD
3. /BACKWARD
4. /SEARCH

Examples

1. In the following example, the STOP/QUEUE command suspends the job that is currently printing on the printer queue JADE_PRINT and places that queue in the paused state. The START/QUEUE command releases the queue from the paused state. The /TOP_OF_FILE qualifier causes the job that was suspended to resume printing at the beginning of the file rather than at where it was interrupted.

```
$ STOP/QUEUE JADE_PRINT
$ START/QUEUE/TOP_OF_FILE JADE_PRINT
```

2. In the following example, the START/QUEUE command resumes output on printer SYS_LPA0 after advancing 15 pages from the beginning of the file:

```
$ START/QUEUE/TOP/FORWARD=15 SYS_LPA0
```

13.9.6.9.2 Aligning Print Forms To print alignment data to aid in aligning printer forms, pause the queue, then enter the START/QUEUE command along with the /ALIGN qualifier in the following format:

START/QUEUE/ALIGN[=(option[,...])]

The following options control the number of alignment pages and type of alignment data:

Option	Description
MASK	Specifies that input data is masked by replacing alphabetic characters with the character X and numeric characters with the number 9. Mask characters let you prevent the printing of sensitive information. If you omit the MASK option, data is printed unaltered.
n	A decimal number in the range 1 through 20 that specifies the number of alignment pages to print. By default, one page of alignment data is printed.

You can use the /ALIGN qualifier along with any of the file positioning qualifiers described in the previous section. File positioning is performed before alignment data is printed. After the alignment is complete, the queue enters a paused state until you restart it by reentering the START/QUEUE command. Printing

resumes from the point that alignment data started; that is, the task is backspaced over the pages printed for alignment.

Example

The command in the following example requests masked alignment for four pages of output:

```
$ START/QUEUE/BACKWARD=2/ALIGN=(MASK,4) SYS_LPA0
```

In this example, the file for the job that was being printed when the queue was paused is backspaced two pages before alignment is performed. Four pages of alignment mask characters are printed. Then the output for the current job is positioned backward four pages, and the queue pauses.

13.10 Solving Queue Problems

The following sections can help you solve common queue problems:

Problem	Section
General printer problems	Section 13.10.1
Pending jobs	Section 13.10.2
Stock mismatch problems	Section 13.10.2.1
Characteristics mismatch problems	Section 13.10.2.2
Stalled output queues	Section 13.10.3
Autostart queues that will not start	Section 13.10.4
Problems deleting queues, forms, and characteristics	Section 13.10.5
Problems adding or deleting a module from a device control library	Section 13.10.6
Queue is disabled	Section 13.10.7

13.10.1 Determining the Cause of General Printer Problems

The following steps can help you determine the cause of general printer problems:

1. Enter a command in the following format to determine the status of the queue with which the printer is associated:

   SHOW QUEUE/FULL queue name

   Table 13–4 lists and describes queue statuses.

2. Enter the command SHOW LOGICAL/FULL SMBSRVSHR to determine whether the logical name SMBSRVSHR is assigned and, if it is assigned, its access mode (SUPERVISOR_MODE or EXECUTIVE_MODE). In most cases, this logical name should not be defined. However, if SMBSRVSHR is assigned, deassign it by entering the command DEASSIGN/access_mode SMBSRVSHR. (For more information about the commands SHOW LOGICAL /FULL and DEASSIGN, see the *OpenVMS DCL Dictionary*.)

3. If a print request returns a fatal error or does not print, perform the following steps:

   a. Stop the queue by entering the STOP/QUEUE/RESET command.

   b. If the output device is spooled, despool it using the SET DEVICE /NOSPOOLED command.

c. Copy the file you are attempting to print by entering the COPY command in the following format:

COPY filename output-device:

Be sure to include a colon (:) after the device name.

If the COPY command does not produce output, a PRINT request will not work.

4. If the problem is on a queue using the LATSYM symbiont, try using the default symbiont, PRTSMB, and see if the problem persists. You can use the PRTSMB symbiont for printers on LAT ports. However, only one queue at a time will be able to send jobs to the printer.

   To determine whether a queue is using the LATSYM symbiont, enter the SHOW QUEUE/FULL command. If the queue is using the LATSYM symbiont, the display shows the following words: /PROCESSOR=LATSYM.

   To change the queue's symbiont to PRTSMB, perform the following steps:

   a. Stop the queue by entering the STOP/QUEUE/RESET command.

   b. Restart the queue using the START/QUEUE/NOPROCESSOR command.

   If a PRINT request succeeds when the queue is using PRTSMB, the problem is with LATSYM or with the LAT driver LTDRIVER.

5. Attach a terminal at the end of the cable that is being used for the printer. If the data shows up on the terminal, the Hold Screen key works, and no data is lost, the problem is probably with the printer (an incorrect setting, for example). If the problem exists with the terminal, it might be caused by the cable, the hardware interface port, or a hardware port setting.

13.10.2 Making Pending Jobs Eligible for Scheduling

If a job does not execute when expected, the job might have a pending or holding status. The SHOW QUEUE/FULL/ALL_JOBS command displays the status for all jobs on a queue.

If the job is in a holding status, see Section 13.9.6.3 for information about holding and releasing a job.

If the job is in the pending state, the /FULL qualifier enables you to see the reason why the job is ineligible to execute. (Use a 132-character wide display to make sure that all the information is displayed.)

For example:

```
$ SHOW QUEUE/FULL/ALL_JOBS/BY_JOB_STATUS=PENDING

Generic printer queue REG$GENERIC
 /GENERIC=(REG$Q1,REG$Q2,REG$Q3)/OWNER=[SYSTEM]/PROTECTION=(S:E,O:D,G:R,W:W)

  Entry  Jobname         Username     Blocks  Status
  -----  -------         --------     ------  ------
    684  PROBLEMS         CHURCHILL     3118  Pending (check execution queues)
         Submitted  7-MAR-1995 17:49 /FORM=DEFAULT /NOTIFY /PRIORITY=100
         File: _$5$DUA174:[CHURCHILL]PROBLEMS.TXT;2

Printer queue REG$Q1, stopped, on LONDON::NPA1, mounted form DEFAULT
 /BASE_PRIORITY=4/DEFAULT=(FEED,FORM=DEFAULT)/OWNER=[SYSTEM]
 /PROTECTION=(S:E,O:D,G:R,W:W)
```

```
Entry  Jobname         Username     Blocks  Status
-----  -------         --------     ------  ------
  687  PM$SPEECH       CHURCHILL      3558  Pending (queue stopped)
       Submitted  7-MAR-1995 17:51 /FORM=DEFAULT /NOTIFY /PRIORITY=100
       File: _$5$DUA174:[CHURCHILL]PM$SPEECH.TXT;1 (checkpointed)
```

A job enters the pending status whenever the job is not eligible to execute. Table 13–7 lists common causes and solutions for a pending status.

Table 13–7 Common Causes and Solutions for Pending Job Status

Problem	Solution
The queue is currently processing as many jobs as it can.	Wait until intervening jobs complete.
The queue is stopped or stalled.	If a SHOW QUEUE/FULL command shows the queue status as stopped or stalled, determine why the queue is stopped or stalled. If the queue is stopped, start it with the START/QUEUE command. If the stopped queue is an autostart queue, use START /QUEUE to activate the queue and ENABLE AUTOSTART/QUEUES to start the queue. If the queue is stalled, follow the steps in Section 13.10.3.
The stock of the print job's form does not match the stock of the queue's mounted form.	Perform the steps in Section 13.10.2.1.
The job was submitted or modified with characteristics not associated with the queue.	Perform the steps in Section 13.10.2.2.
The queue has a block limit set and the size of the print job does not fall within the specified range.	Use the SET ENTRY/REQUEUE command to move the job to another queue, or use the SET QUEUE/[NO]BLOCK_LIMIT command to change or remove the block limit on the queue.
The owner of the job does not have write access to the execution queue.	Use to SET ENTRY/REQUEUE command to move the job to another queue, or change the access to the queue as explained in Section 13.8.1.
The print job is in a logical queue that is assigned to a stopped execution queue.	Use the SET ENTRY/REQUEUE command to requeue the job to another queue or start the execution queue to which the logical queue is assigned.
An output job requires an output device that is enabled for lowercase printing.	If the printer supports lowercase printing, use the /LOWERCASE qualifier with the SET PRINTER or SET TERMINAL command to enable lowercase printing on the device. Otherwise, use the SET ENTRY/REQUEUE command to move the job to an execution queue that sends its output to a printer with lowercase printing enabled.

13.10.2.1 Fixing Print Jobs That Are Pending Due to Stock Mismatch

When monitoring jobs, you might see a print job that is pending in a queue because the stock does not match that of the mounted form. For example, you might see a SHOW ENTRY display similar to the following:

```
$ SHOW ENTRY 133/FULL

  Entry  Jobname         Username     Blocks  Status
  -----  -------         --------     ------  ------
    133  SET             RANDOM           74  Pending (stock type mismatch)
         On idle printer queue SUE$PRINT
         Submitted 21-JAN-1995 16:14 /FORM=MANUAL (stock=HQ)  /PRIORITY=100
         File: _$5$DUA1:[RANDOM]SET.TXT;5
```

To fix jobs that are pending because of a stock mismatch, do one or more of the following:

- Make the stocks match by mounting a different form on the queue (for example, using the SET QUEUE/FORM_MOUNTED command) or by specifying a different form with the job (for example, using the SET ENTRY /FORM command).
- Move the job to a queue on which the stock of the mounted form matches the stock of the job's form (for example, using the SET ENTRY/REQUEUE command).
- Delete the job (for example, using the DELETE/ENTRY command).

See Section 13.8.9 for more information about forms.

13.10.2.2 Fixing Jobs That Are Pending Because of Characteristics Mismatch

When monitoring jobs, you might see a batch or print job that is pending in a queue because the characteristics do not match those assigned to the queue. For example, you might see a SHOW ENTRY display similar to the following:

```
$ SHOW ENTRY 882/FULL

  Entry  Jobname         Username     Blocks  Status
  -----  -------         --------     ------  ------
    882  SETHOST         RANDOM            5  Pending (characteristics mismatch)
         On idle printer queue $PRINTER_1
         Submitted 28-MAR-1995 15:21 /CHAR=(5) /FORM=DEFAULT /PRIORITY=100
         File: _$5$DUA1:[RANDOM]SETHOST.LOG;5
```

To fix jobs that are pending because of a characteristics mismatch, do one or more of the following:

- Move the job to a queue on which the characteristics match those of the job (for example, using the SET ENTRY/REQUEUE command).
- Delete the job (for example, using the DELETE/ENTRY command).
- Make the characteristics match by assigning new characteristics to the queue (for example, using the SET QUEUE/CHARACTERISTIC command).

See Section 13.8.3 for more information about characteristics.

13.10.3 Fixing a Stalled Output Queue

If an output queue is in the stalled state, the device on which the queue is running is malfunctioning. Check the device and fix the problem. Once the problem is fixed, the queue will leave the stalled state.

If you cannot fix the problem immediately, stop the queue by entering the command STOP/QUEUE/RESET. While the queue is stopped, you might want to reroute the jobs in the queue to a functioning queue, as explained in Section 13.9.1.14. When the problem is fixed, deassign the logical queue and start the queue by entering the START/QUEUE command.

13.10.4 Determining Why an Autostart Queue Will Not Start

If you attempt to start an autostart queue with the ENABLE AUTOSTART /QUEUES command and the queue does not start, the queue might not be active for autostart. The ENABLE AUTOSTART/QUEUES command starts only *active* autostart queues capable of running on a node. To activate an autostart queue, you must include the /START qualifier with the INITIALIZE/QUEUE command or enter the START/QUEUE command. For example:

```
$ ENABLE AUTOSTART/QUEUES/ON_NODE=KATY:: ❶
$ SHOW QUEUE KATY_BATCH
Batch queue KATY_BATCH, stopped, autostart inactive, on KATY:: ❷
$ START/QUEUE KATY_BATCH ❸
$ SHOW QUEUE KATY_BATCH/ALL
Batch queue KATY_BATCH, idle, on KATY:: ❹
```

❶ The ENABLE AUTOSTART/QUEUES command attempts to start autostart queues on node KATY.

❷ The SHOW QUEUE display shows that the autostart batch queue KATY_ BATCH did not start. The display also reveals that the queue is inactive for autostart; the queue was either not activated initially or it was deactivated with a STOP/QUEUE/NEXT or STOP/QUEUE/RESET command.

❸ The START/QUEUE command activates the queue for autostart.

❹ The SHOW QUEUE command informs you that the queue is started.

13.10.5 Solving Problems Deleting a Queue, Form, or Characteristic

If you are having problems deleting a queue, form, or characteristic, make sure you have met the following requirements:

- When deleting a form or characteristic with the DELETE/FORM or DELETE /CHARACTERISTIC command, you must specify the *name* assigned to the form or characteristic. (The form or characteristic *number* cannot be used with the DELETE command.) To determine the name of a form or characteristic, enter the SHOW QUEUE/FORM or SHOW QUEUE /CHARACTERISTIC command.
- A queue must be stopped before being deleted.
- All references to a queue, form, or characteristic must be removed before you can delete the queue, form, or characteristic.

If you see a message similar to the following, a reference to the queue, form, or characteristic still exists:

```
%DELETE-E-NOTDELETED, error deleting object-name
-JBC-E-REFERENCED, existing references prevent deletion
```

For example, the queue you are attempting to delete might be named as a target for a generic queue, or the form you are attempting to delete might be specified for a print job. All references to a queue, form, or characteristic must be removed before you can delete the queue, form, or characteristic.

How to Perform This Task

Perform the following steps to find and remove references to a queue, form, or characteristic:

1. Enter the command SHOW QUEUE/FULL/ALL/OUTPUT=*filename*, where *filename* is a name of a file to which the display output of the command is to be sent.
2. Use the SEARCH command to search the output file for the *name* of the form or queue, or the *number* of the characteristic to be deleted. The result of your search will include all references to the queue, form, or characteristic.
3. If the SEARCH command reveals any queues with references to the queue, form, or characteristic you are trying to delete, perform the following steps:
   a. Use the STOP/QUEUE/NEXT command to stop each queue with an offending reference.
   b. Use the START/QUEUE command with the appropriate qualifiers to restart each queue without the reference.
4. If the SEARCH command reveals any jobs with references to the queue, form, or characteristic you are trying to delete, perform *one* of the following steps to eliminate the job reference:
   - Wait for the job to complete. (You might want to raise the priority of the job as explained in Section 13.9.6.4, so the job will be scheduled for processing sooner.)
   - Delete the job as explained in Section 13.9.6.8.
   - Modify the options for the job as explained in Section 13.9.6.2, or ask the owner of the job to do so.

Example

The following example includes several commands used to fix problems preventing the deletion of a queue:

```
$ ❶ DELETE/QUEUE JADE_BATCH
❷ %DELETE-E-NOTDELETED, error deleting JADE_BATCH
-JBC-E-QUENOTSTOP, queue must be stopped to perform operation
$ ❸ STOP/QUEUE/NEXT JADE_BATCH
$ ❹ DELETE/QUEUE JADE_BATCH
❺ %DELETE-E-NOTDELETED, error deleting JADE_BATCH
-JBC-E-REFERENCED, existing references prevent deletion
$ SHOW QUEUE/FULL
   .
   .
   .
❻ Generic batch queue CLUSTER_BATCH
  /GENERIC=(JADE_BATCH,RUBY_BATCH,OPAL_BATCH) /OWNER=[SYSTEM]
  /PROTECTION=(S:E,O:D,G:R,W:W)
   .
   .
   .
```

```
$ ❼ STOP/QUEUE/NEXT CLUSTER_BATCH
$ ❽ START/QUEUE CLUSTER_BATCH/GENERIC=(RUBY_BATCH,OPAL_BATCH)
$ ❾ DELETE/QUEUE JADE_BATCH
```

❶ The DELETE/QUEUE command attempts to delete the queue.

❷ The message indicates that the queue is not stopped.

❸ The STOP/QUEUE/NEXT command stops the queue after allowing the current job to complete.

❹ The DELETE/QUEUE command again attempts to delete the queue.

❺ This time, the message indicates that references to the queue exist.

❻ The SHOW QUEUE/FULL command shows information about all queues. However, the only reference to JADE_BATCH names the queue as a target queue for the generic queue CLUSTER_BATCH.

❼ The STOP/QUEUE/NEXT command stops the generic queue that targets JADE_BATCH.

❽ The START/QUEUE command eliminates the reference to JADE_BATCH by restarting the generic queue without specifying JADE_BATCH as a target.

❾ The DELETE/QUEUE command successfully deletes the queue.

13.10.6 Adding or Deleting a Device Control Library Module

When attempting to add or delete a device control library module, you might see the following message:

```
$LIBRAR-F-OPENIN, error opening module-name
-RMS-E-FLK, file currently locked by another user
```

To add or delete a library module, you must stop all output queues to which the library is assigned. To determine which queues the library is assigned to, perform the following steps:

1. Enter a command similar to the following, where *filename* is the name of a file to which the display output of the command is to be sent:

   SHOW QUEUE/FULL/OUTPUT=filename

2. Use the SEARCH command to search the output file for the name of the library.

The result of your search will include all queues to which that library is assigned. Stop the queues and reenter the command to add or delete the library module.

Note

The SHOW QUEUE/FULL display shows the library assigned to a queue only if you explicitly assigned a library for the queue by including the /LIBRARY qualifier with the INITIALIZE/QUEUE or START/QUEUE command. If you do not explicitly assign a library to a queue, the default library, SYSDEVCTL, is used.

If the module you are trying to delete is in the default library, SYSDEVCTL, you must stop all queues for which the SHOW QUEUE /FULL command displays no library. To make sure the SYSDEVCTL library appears in the SHOW QUEUE/FULL display in the future, specify /LIBRARY=SYSDEVCTL when you restart the queue.

If you cannot stop the queues immediately, perform the following steps:

1. Use the COPY command to copy the library to be modified into your own directory.
2. Add the module to the copy, or delete the module from the copy.
3. Use the COPY command to copy the library back to the directory SYS$COMMON:[SYSLIB]. As long as you use the same name for the modified library, the library will be picked up by each queue when the queue is stopped and restarted.

If your site has a large number of different printers, you can help prevent this problem by using more libraries, so that each library is assigned to fewer queues. For example, you should create and assign a different library for each type of printer, as explained in Section 13.9.5.2.

13.10.7 Fixing a Disabled Queue

The queue manager attempts to correct any kind of corruption detected. If the queue manager detects corruption in a queue record, it might disable a queue to isolate the corruption. When a queue is disabled, the following message is written on the console and in the operator log file:

```
%QMAN-I-QUEDISCOR, queue 'queue_name' has been
disabled due to database corruption
```

When a queue is disabled, any attempt to modify or submit a job to it returns the following message:

```
%JBC-E-QUEDISABLED, disabled queue cannot be modified, nor can a job be
submitted to it
```

If you see either of the previous messages, do the following:

1. Submit a Software Performance Report (SPR) and include copies of the queue file and master file of the queue database as explained in Section 12.14.
2. Delete the queue and create a new queue to replace it.

13.10.8 Reporting Queue Problems to Digital

If you encounter a problem with your queues, and you want to report it to Digital, please provide as much information as possible. Section 12.14 specifies the information that is most useful to Digital in diagnosing your queuing system problems.

Index

A

Aborting job status, *Sys Mgr Man: Essentials*, 13–69
Access control entries
  See ACEs
Access control lists
  See ACLs
Accessibility field
  tape file system checks, *Sys Mgr Man: Essentials*, 9–17
Access types
  abbreviations of, *Sys Mgr Man: Essentials*, 11–8
  checking when writing files to tape volumes, *Sys Mgr Man: Essentials*, 9–18
  protecting disk directory files, *Sys Mgr Man: Essentials*, 9–10
  protecting disk files, *Sys Mgr Man: Essentials*, 9–6
  protection codes and, *Sys Mgr Man: Essentials*, 11–8
  read, *Sys Mgr Man: Essentials*, 9–13
    continuation volumes, *Sys Mgr Man: Essentials*, 8–42
  write
    continuation volumes, *Sys Mgr Man: Essentials*, 8–41
Account expiration, *Sys Mgr Man: Essentials*, 6–43
ACCOUNTING command, *Sys Mgr Man: Tuning*, 19–4
Accounting groups
  setting up, *Sys Mgr Man: Tuning*, 19–5
Accounting utility (ACCOUNTING), *Sys Mgr Man: Tuning*, 19–4
ACCOUNTNG.DAT file, *Sys Mgr Man: Tuning*, 19–3
ACCOUNTNG logical name, *Sys Mgr Man: Tuning*, 19–4
Accounts
  access, *Sys Mgr Man: Essentials*, 6–16
  adding, *Sys Mgr Man: Essentials*, 6–17, 6–18
    with ADDUSER.COM, *Sys Mgr Man: Essentials*, 6–18
  adding proxy logins, *Sys Mgr Man: Essentials*, 6–36
Accounts (cont'd)
  automatic login, *Sys Mgr Man: Essentials*, 6–32
  captive, *Sys Mgr Man: Essentials*, 6–13
  deleting, *Sys Mgr Man: Essentials*, 6–25
  directory, *Sys Mgr Man: Essentials*, 6–15
  disabling, *Sys Mgr Man: Essentials*, 6–27
  MAIL, *Sys Mgr Man: Essentials*, 6–38
  maintaining, *Sys Mgr Man: Essentials*, 6–24
  network proxy, *Sys Mgr Man: Essentials*, 6–34
  project, *Sys Mgr Man: Essentials*, 6–33
  restricted, *Sys Mgr Man: Essentials*, 6–13
  restricting use, *Sys Mgr Man: Essentials*, 6–27
  security, *Sys Mgr Man: Essentials*, 6–16
  using ADDUSER.COM, *Sys Mgr Man: Essentials*, 6–18
ACEs (access control entries)
  adding to ACL after file is created, *Sys Mgr Man: Essentials*, 9–7
  Creator ACEs, *Sys Mgr Man: Essentials*, 11–9
  Default Protection ACEs, *Sys Mgr Man: Essentials*, 11–9
  Identifier ACEs, *Sys Mgr Man: Essentials*, 11–9
  none for subdirectories, *Sys Mgr Man: Essentials*, 9–12
  Security Alarm, *Sys Mgr Man: Essentials*, 11–9
  Security Audit, *Sys Mgr Man: Essentials*, 11–10
  Subsystem ACEs, *Sys Mgr Man: Essentials*, 11–10
  using to override default UIC protection, *Sys Mgr Man: Essentials*, 9–12
ACL editor
  invoking, *Sys Mgr Man: Essentials*, 11–11
ACLs (access control lists), *Sys Mgr Man: Essentials*, 6–16, 6–33
  default protection, *Sys Mgr Man: Essentials*, 9–12
  on public volumes, *Sys Mgr Man: Essentials*, 8–14
  on queues, *Sys Mgr Man: Essentials*, 13–24
  on vector capability object, *Sys Mgr Man: Tuning*, 26–8
  SHOW ACL command, *Sys Mgr Man: Essentials*, 9–5

Activating an autostart queue, *Sys Mgr Man: Essentials*, 13–15, 13–16, 13–49
on LAT queues, *Sys Mgr Man: Essentials*, 13–4
relationship to starting an autostart queue, *Sys Mgr Man: Essentials*, 13–4
Active disks
backing up, *Sys Mgr Man: Essentials*, 10–63
Active sets, *Sys Mgr Man: Tuning*, 26–2
displaying, *Sys Mgr Man: Tuning*, 26–3
Active system parameters, *Sys Mgr Man: Tuning*, 14–3, 14–26
Adaptive routing
network, *Sys Mgr Man: Tuning*, 21–5
Adding comments to Digital messages in the Help Message database, *Sys Mgr Man: Essentials*, 5–30
Adding files to the system disk, *Sys Mgr Man: Essentials*, 5–1
Adding messages to the Help Message database, *Sys Mgr Man: Essentials*, 5–32
ADDUSER.COM command procedure, *Sys Mgr Man: Essentials*, 6–18
ADD_DUMPFILE symbol, *Sys Mgr Man: Tuning*, 15–24
ADD_PAGEFILE*n*_SIZE symbol, *Sys Mgr Man: Tuning*, 15–24
ADD_PAGEFILE symbol, *Sys Mgr Man: Tuning*, 15–24
ADD_ prefix for AUTOGEN, *Sys Mgr Man: Tuning*, 14–19
ADD_SWAPFILE*n*_SIZE symbol, *Sys Mgr Man: Tuning*, 15–24
ADD_SWAPFILE symbol, *Sys Mgr Man: Tuning*, 15–24
Adjacent nodes
definition, *Sys Mgr Man: Tuning*, 21–4
AGEN$FEEDBACK.DAT file
description, *Sys Mgr Man: Tuning*, 14–11
AGEN$FEEDBACK_REQ_TIME logical name, *Sys Mgr Man: Tuning*, 14–21
AGEN$PARAMS.REPORT file, *Sys Mgr Man: Tuning*, 14–11
sample, *Sys Mgr Man: Tuning*, 14–12
Alarms
security
enabling, *Sys Mgr Man: Tuning*, 18–29
security applications, *Sys Mgr Man: Essentials*, 11–13
ALF (automatic login facility)
See Automatic login facility
Alias file name
assigning, *Sys Mgr Man: Essentials*, 9–10
Alias VMScluster name, *Sys Mgr Man: Essentials*, 2–15
Aligning preprinted forms, *Sys Mgr Man: Essentials*, 13–75, 13–76
Aligning queue status, *Sys Mgr Man: Essentials*, 13–50
Alignment data, *Sys Mgr Man: Essentials*, 13–76
ALLOCATE command
allocating a particular type of device, *Sys Mgr Man: Essentials*, 8–9
tape drive, *Sys Mgr Man: Essentials*, 9–21, 9–22
to allocate device, *Sys Mgr Man: Essentials*, 8–9
Allocating
disk drives, *Sys Mgr Man: Essentials*, 8–9
allocating a particular type of device, *Sys Mgr Man: Essentials*, 8–9
space on disk volume, *Sys Mgr Man: Essentials*, 8–52
tape drives, *Sys Mgr Man: Essentials*, 8–9
ALPHAVMSSYS.PAR file, *Sys Mgr Man: Essentials*, 4–2; *Sys Mgr Man: Tuning*, 14–3
initializing parameters at boot time, *Sys Mgr Man: Tuning*, 14–36
Alternate Root
VMSINSTAL.COM option, *Sys Mgr Man: Essentials*, 3–18
restriction, *Sys Mgr Man: Essentials*, 3–18
specifying for software installations, *Sys Mgr Man: Essentials*, 3–12
Alternate root directory
adding to an existing system disk, *Sys Mgr Man: Essentials*, 2–31
Alternate startup command procedure
specifying, *Sys Mgr Man: Essentials*, 4–12
as the default, *Sys Mgr Man: Essentials*, 4–13
Alternate Working Device
VMSINSTAL.COM option, *Sys Mgr Man: Essentials*, 3–14
ANALYZE/AUDIT command
See also Audit Analysis utility
generating security reports, *Sys Mgr Man: Tuning*, 18–30
Analyze/Disk_Structure utility (ANALYZE/DISK_STRUCTURE)
builds BITMAP.SYS file, *Sys Mgr Man: Tuning*, A–8
checks validity of files, *Sys Mgr Man: Tuning*, A–8
commands, *Sys Mgr Man: Essentials*, 8–56
creates version of quota file, *Sys Mgr Man: Tuning*, A–9
creating disk usage file, *Sys Mgr Man: Essentials*, 8–56
directing output, *Sys Mgr Man: Essentials*, 8–56
files used by, *Sys Mgr Man: Tuning*, A–4

Analyze/Disk_Structure utility (ANALYZE/DISK_ STRUCTURE) (cont'd)
identification record, *Sys Mgr Man: Essentials*, 8–56
listing file information, *Sys Mgr Man: Essentials*, 8–56
recovering lost files, *Sys Mgr Man: Essentials*, 8–57
repairing disk errors, *Sys Mgr Man: Essentials*, 8–57
reporting disk errors, *Sys Mgr Man: Essentials*, 8–56
uses of, *Sys Mgr Man: Essentials*, 8–55
uses VOLSET.SYS to locate volumes in set, *Sys Mgr Man: Tuning*, A–9
verifies files in directory structure, *Sys Mgr Man: Tuning*, A–9
ANALYZE/ERROR_LOG command, *Sys Mgr Man: Tuning*, 18–5
Error Log utility, *Sys Mgr Man: Tuning*, 18–6
excluding unknown entries, *Sys Mgr Man: Tuning*, 18–8
specifying output, *Sys Mgr Man: Tuning*, 18–7
ANALYZE/MEDIA command
to invoke Bad Block Locator utility, *Sys Mgr Man: Essentials*, 8–64
Analyzing a crash dump, *Sys Mgr Man: Tuning*, 15–11
See also Crash dumps
See also System failures
in system startup, *Sys Mgr Man: Essentials*, 5–13; *Sys Mgr Man: Tuning*, 15–15
Announcements
creating systemwide, *Sys Mgr Man: Essentials*, 5–14
displaying system, *Sys Mgr Man: Essentials*, 5–15
Anonymous account, *Sys Mgr Man: Tuning*, 21–14
Answer file (for software installation), *Sys Mgr Man: Essentials*, 3–14
APB.EXE file, *Sys Mgr Man: Essentials*, 4–18
role in boot process, *Sys Mgr Man: Essentials*, 4–2
Application images
registering with the Image Registry facility, *Sys Mgr Man: Essentials*, 5–22
Applications
TCP/IP, *Sys Mgr Man: Tuning*, 21–13
Archive file
creating for each node in a cluster, *Sys Mgr Man: Tuning*, 18–27
Area routers
See Routers, Level 2
Area routing
network, *Sys Mgr Man: Tuning*, 21–6
Arrow keys
functions of, *Sys Mgr Man: Tuning*, 20–7
Assigning
a default form to a queue, *Sys Mgr Man: Essentials*, 13–63
a logical queue, *Sys Mgr Man: Essentials*, 13–56
characteristics to a queue, *Sys Mgr Man: Essentials*, 13–59
libraries to queues, *Sys Mgr Man: Essentials*, 13–66
reset modules to a queues, *Sys Mgr Man: Essentials*, 13–67
ASSIGN/MERGE command, *Sys Mgr Man: Essentials*, 13–57
Assisted merge and volume shadowing, *Sys Mgr Man: Essentials*, 10–41
Asterisk (*)
as wildcard character, *Sys Mgr Man: Essentials*, 9–16
ASTLM process limit, *Sys Mgr Man: Essentials*, 6–42
value for efficient backups, *Sys Mgr Man: Essentials*, 10–10
AST queue process limit, *Sys Mgr Man: Essentials*, 6–42
Asymmetric vector processing configuration, *Sys Mgr Man: Tuning*, 26–4
Asynchronous DECnet
using virtual terminals, *Sys Mgr Man: Essentials*, 7–11, 7–12
Attached processors, *Sys Mgr Man: Tuning*, 26–2
Audit Analysis utility (ANALYZE/AUDIT), *Sys Mgr Man: Essentials*, 11–14
See also ANALYZE/AUDIT command
generating security reports, *Sys Mgr Man: Tuning*, 18–30
Auditing
security, *Sys Mgr Man: Essentials*, 11–2, 11–13
See also Security audit log files
See Security auditing
displaying using SHOW AUDIT command, *Sys Mgr Man: Tuning*, 18–27
Audit log files, *Sys Mgr Man: Tuning*, 18–2
See also Security audit log files
Audit server processes
creation during system startup, *Sys Mgr Man: Essentials*, 5–5
Authentication of user identity, *Sys Mgr Man: Tuning*, 21–14
Authorization files, *Sys Mgr Man: Essentials*, 6–4
Authorize utility (AUTHORIZE)
ADD command, *Sys Mgr Man: Essentials*, 6–17
ADD/IDENTIFIER command, *Sys Mgr Man: Essentials*, 6–33

Authorize utility (AUTHORIZE) (cont'd)
adding a user account, *Sys Mgr Man: Essentials*, 6–17
checking UAF quotas for software installation, *Sys Mgr Man: Essentials*, 3–7
GRANT/IDENTIFIER command, *Sys Mgr Man: Essentials*, 6–33
limiting page file usage, *Sys Mgr Man: Tuning*, 15–10
listing user records, *Sys Mgr Man: Essentials*, 6–23
modifying a user account, *Sys Mgr Man: Essentials*, 6–23
modifying process limits for SYSTEM account, *Sys Mgr Man: Essentials*, 3–7
reducing process paging, *Sys Mgr Man: Tuning*, 26–7
restricting login hours with, *Sys Mgr Man: Tuning*, 16–4
restricting system use with, *Sys Mgr Man: Tuning*, 16–4
setting process quotas for efficient backups, *Sys Mgr Man: Essentials*, 10–9
Autoconfiguration
See also AUTOCONFIGURE command
benefits, *Sys Mgr Man: Essentials*, 7–6
definition, *Sys Mgr Man: Essentials*, 5–8, 7–6
in system startup, *Sys Mgr Man: Essentials*, 5–4
suppressing, *Sys Mgr Man: Essentials*, 5–8, 7–9
AUTOCONFIGURE command
See also Autoconfiguration
See also IO AUTOCONFIGURE command
in SYSGEN (VAX), *Sys Mgr Man: Essentials*, 7–6
in system startup, *Sys Mgr Man: Essentials*, 5–4, 5–7
suppressing, *Sys Mgr Man: Essentials*, 5–8, 7–9
AUTOGEN.COM command procedure, *Sys Mgr Man: Tuning*, 14–8
See also AUTOGEN feedback
See also MODPARAMS.DAT file
ADD_ prefix, *Sys Mgr Man: Tuning*, 14–19
AGEN$PARAMS.REPORT file, *Sys Mgr Man: Tuning*, 14–11
as recommended method for changing system parameters, *Sys Mgr Man: Tuning*, 14–5
automatic management of page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–22
calculation of page file size, *Sys Mgr Man: Tuning*, 15–4, 15–22, 15–23, 15–26
calculation of swap file size, *Sys Mgr Man: Tuning*, 15–5, 15–22, 15–23, 15–26
calculation of system dump file size, *Sys Mgr Man: Tuning*, 15–2, 15–22, 15–23, 15–26
AUTOGEN.COM command procedure (cont'd)
changing page, swap, and dump file sizes, *Sys Mgr Man: Tuning*, 15–22
changing system parameters, *Sys Mgr Man: Tuning*, 14–3, 14–4
controlling operations performed by, *Sys Mgr Man: Tuning*, 14–10
controlling system parameter values set by, *Sys Mgr Man: Tuning*, 14–18
converting system parameter values for use with, *Sys Mgr Man: Tuning*, 14–5
creating page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–17, 15–24
defining the number of VAXcluster nodes for, *Sys Mgr Man: Tuning*, 14–21
displaying page, swap, and dump file size calculations, *Sys Mgr Man: Tuning*, 15–18, 15–22
end phase
specifying when invoking, *Sys Mgr Man: Tuning*, 14–9
executing
in batch, *Sys Mgr Man: Tuning*, 14–22
interactively, *Sys Mgr Man: Tuning*, 14–18
execution mode
specifying when invoking, *Sys Mgr Man: Tuning*, 14–9
failure executing SYCONFIG.COM, *Sys Mgr Man: Essentials*, 7–10
feedback
See AUTOGEN feedback
functions of, *Sys Mgr Man: Tuning*, 14–9
installing page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–18, 15–22
instructions for using, *Sys Mgr Man: Tuning*, 14–18
invoking, *Sys Mgr Man: Tuning*, 14–9
MODPARAMS.DAT file
See MODPARAMS.DAT file
parameters to, *Sys Mgr Man: Tuning*, 14–9
performance tuning, *Sys Mgr Man: Tuning*, 16–5
phases
order of, *Sys Mgr Man: Tuning*, 14–16
specifying when invoking, *Sys Mgr Man: Tuning*, 14–9
recommendation
for using to size page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–22
restrictions
for changing file sizes, *Sys Mgr Man: Tuning*, 15–23
for specifying page and swap file sizes, *Sys Mgr Man: Tuning*, 15–23
reviewing calculations of, *Sys Mgr Man: Tuning*, 14–10, 14–18
SETPARAMS.DAT file, *Sys Mgr Man: Tuning*, 14–18

AUTOGEN.COM command procedure (cont'd)
specifying an alternate default startup command procedure, *Sys Mgr Man: Essentials*, 4–13
specifying location of page and swap files, *Sys Mgr Man: Tuning*, 15–24
specifying number of Ethernet adapters in, *Sys Mgr Man: Tuning*, 14–21
specifying size of page and swap files
of individual files, *Sys Mgr Man: Tuning*, 15–22, 15–24
of total file space, *Sys Mgr Man: Tuning*, 15–22, 15–23
specifying values for parameters not calculated by, *Sys Mgr Man: Tuning*, 14–20
start phase
specifying when invoking, *Sys Mgr Man: Tuning*, 14–9
system parameters affected by, *Sys Mgr Man: Tuning*, 14–10
timing of file size calculations, *Sys Mgr Man: Tuning*, 15–23
types of data collected by, *Sys Mgr Man: Tuning*, 14–9
when to run, *Sys Mgr Man: Tuning*, 14–8
AUTOGEN feedback, *Sys Mgr Man: Tuning*, 14–8 to 14–22
checks performed on, *Sys Mgr Man: Tuning*, 14–11
collection of, *Sys Mgr Man: Tuning*, 14–11
examining effect on parameters, *Sys Mgr Man: Tuning*, 14–11
file stored in, *Sys Mgr Man: Tuning*, 14–11
importance of system work load, *Sys Mgr Man: Tuning*, 14–11
improving system performance, *Sys Mgr Man: Tuning*, 14–10
maximum age, *Sys Mgr Man: Tuning*, 14–11
minimum age, *Sys Mgr Man: Tuning*, 14–11, 14–21
report file, *Sys Mgr Man: Tuning*, 14–11
sample, *Sys Mgr Man: Tuning*, 14–12
sending automatically, *Sys Mgr Man: Tuning*, 14–8, 14–22
resources affected by, *Sys Mgr Man: Tuning*, 14–10
saving during system shutdown, *Sys Mgr Man: Essentials*, 4–29
Automatic configurations
of DECnet node, *Sys Mgr Man: Tuning*, 21–12
of devices, *Sys Mgr Man: Essentials*, 7–6
Automatic login facility (ALF)
setting up an automatic login account, *Sys Mgr Man: Essentials*, 6–32
Automatic start
See also Autostart feature
of queue manager, *Sys Mgr Man: Essentials*, 12–3, 12–7, 12–9
Automatic volume switching, *Sys Mgr Man: Essentials*, 8–39
Autostart feature
See also Autostart queues
description, *Sys Mgr Man: Essentials*, 13–4
disabling, *Sys Mgr Man: Essentials*, 12–9, 13–54
before shutting down a node, *Sys Mgr Man: Essentials*, 13–55
enabling, *Sys Mgr Man: Essentials*, 13–4, 13–18, 13–48
on LAT queues, *Sys Mgr Man: Essentials*, 13–4, 13–10
recommended use, *Sys Mgr Man: Essentials*, 13–10
with LAT queues, *Sys Mgr Man: Essentials*, 13–4, 13–10
Autostart queues
See also Autostart feature
activating, *Sys Mgr Man: Essentials*, 13–4, 13–15, 13–16, 13–49
activating inactive queue status, *Sys Mgr Man: Essentials*, 13–50
creating, *Sys Mgr Man: Essentials*, 13–15, 13–16
preventing from starting, *Sys Mgr Man: Essentials*, 13–54
recommended use, *Sys Mgr Man: Essentials*, 13–10
relationship between activating and starting, *Sys Mgr Man: Essentials*, 13–4
starting, *Sys Mgr Man: Essentials*, 13–48
in startup command procedure, *Sys Mgr Man: Essentials*, 13–18
troubleshooting, *Sys Mgr Man: Essentials*, 13–81
with LAT printers, *Sys Mgr Man: Essentials*, 13–4, 13–10
AUTO_POSITIONING command
SHOW CLUSTER, *Sys Mgr Man: Tuning*, 20–12
Availability
of queue manager, *Sys Mgr Man: Essentials*, 12–9, 12–17
of queues, *Sys Mgr Man: Essentials*, 13–4, 13–15
Availability of devices
OPCOM message, *Sys Mgr Man: Essentials*, 8–59
Available queue status, *Sys Mgr Man: Essentials*, 13–51
Available sets, *Sys Mgr Man: Tuning*, 26–2

B

Backing up the system disk, *Sys Mgr Man: Essentials*, 5–50
  after installation, *Sys Mgr Man: Essentials*, 3–13
Backlinks
  definition, *Sys Mgr Man: Essentials*, 8–56
BACKUP.SYS file
  See Backup log file
BACKUP command
  and save sets, *Sys Mgr Man: Essentials*, 10–23
  /EXACT_ORDER qualifier, *Sys Mgr Man: Essentials*, 10–22
  for backing up directories, *Sys Mgr Man: Essentials*, 10–24
  for copying directories, *Sys Mgr Man: Essentials*, 10–23
  for copying files, *Sys Mgr Man: Essentials*, 10–23
  for image backups, *Sys Mgr Man: Essentials*, 10–31, 10–32
  for incremental backups, *Sys Mgr Man: Essentials*, 10–34, 10–35
  format, *Sys Mgr Man: Essentials*, 10–5
  /GROUP_SIZE qualifier, *Sys Mgr Man: Essentials*, 10–62
  /IGNORE=LABEL_PROCESSING qualifier, *Sys Mgr Man: Essentials*, 10–22, 10–65
  /IGNORE qualifier, *Sys Mgr Man: Essentials*, 10–30, 10–63
  /IMAGE qualifier, *Sys Mgr Man: Essentials*, 10–31, 10–32, 10–55
  /INITIALIZE qualifier, *Sys Mgr Man: Essentials*, 10–14
  in VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–13
  /JOURNAL qualifier, *Sys Mgr Man: Essentials*, 10–26
  /LABEL qualifier, *Sys Mgr Man: Essentials*, 10–13, 10–23, 10–34
  /LOG qualifier, *Sys Mgr Man: Essentials*, 10–63
  /PHYSICAL qualifier, *Sys Mgr Man: Essentials*, 10–55
  qualifiers, *Sys Mgr Man: Essentials*, 10–5
  /RECORD qualifier, *Sys Mgr Man: Essentials*, 10–31, 10–32, 10–34, 10–35
  /REWIND qualifier, *Sys Mgr Man: Essentials*, 10–13, 10–34
  /SAVE_SET qualifier, *Sys Mgr Man: Essentials*, 10–32, 10–35
  /SINCE qualifier, *Sys Mgr Man: Essentials*, 10–34, 10–35
  /VERIFY qualifier, *Sys Mgr Man: Essentials*, 10–63
BACKUP command (cont'd)
  with multiple output devices, *Sys Mgr Man: Essentials*, 10–24, 10–31, 10–33
Backup log file
  BACKUP.SYS, *Sys Mgr Man: Tuning*, A–9
  reserved file, *Sys Mgr Man: Tuning*, A–9
Backup Manager, *Sys Mgr Man: Essentials*, 10–5
  features, *Sys Mgr Man: Essentials*, 10–5
  getting started, *Sys Mgr Man: Essentials*, 10–6
  types of help available, *Sys Mgr Man: Essentials*, 10–5
BACKUP media
  Files–11 disk save set, *Sys Mgr Man: Essentials*, 10–7
  magnetic tape save set, *Sys Mgr Man: Essentials*, 10–6
  network save set, *Sys Mgr Man: Essentials*, 10–7
  sequential-disk save set, *Sys Mgr Man: Essentials*, 10–8
Backups
  image
    See Image backup
  incremental
    See Incremental backup
  performing in command procedures, *Sys Mgr Man: Essentials*, 8–46
  standalone
    See Standalone BACKUP
Backup utility (BACKUP)
  backing up active disks, *Sys Mgr Man: Essentials*, 10–63
  backing up shadow sets, *Sys Mgr Man: Essentials*, 10–40
  Backup Manager, *Sys Mgr Man: Essentials*, 10–4
  command format, *Sys Mgr Man: Essentials*, 10–5
  command procedures, *Sys Mgr Man: Essentials*, 10–36
  command qualifiers, *Sys Mgr Man: Essentials*, 10–5
  copying dump file, *Sys Mgr Man: Tuning*, 15–4, 15–14
  copying the queue database, *Sys Mgr Man: Essentials*, 12–12
  data integrity mechanisms, *Sys Mgr Man: Essentials*, 10–62
  InfoServer tapes, *Sys Mgr Man: Essentials*, 10–60
  interfaces to it, *Sys Mgr Man: Essentials*, 10–4
  open files during a backup, *Sys Mgr Man: Essentials*, 10–30, 10–63
  restoring shadow sets, *Sys Mgr Man: Essentials*, 10–47
  save set, *Sys Mgr Man: Essentials*, 10–23

Backup utility (BACKUP) (cont'd)
stores save-set file in MFD, *Sys Mgr Man: Tuning*, A–8
transferring information, *Sys Mgr Man: Essentials*, 9–19
use by VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–13
using on workstations, *Sys Mgr Man: Essentials*, 10–36
using to back up directories, *Sys Mgr Man: Essentials*, 10–24
using to copy directories, *Sys Mgr Man: Essentials*, 10–23
using to copy files, *Sys Mgr Man: Essentials*, 10–23
BACKUSER.COM command procedure, *Sys Mgr Man: Essentials*, 10–36
BADBLK.SYS file
See Bad block file
Bad block file
BADBLK.SYS, *Sys Mgr Man: Tuning*, A–8
definition, *Sys Mgr Man: Tuning*, A–8
reserved file, *Sys Mgr Man: Tuning*, A–8
Bad Block Locator utility (BAD)
analyzing media for bad blocks, *Sys Mgr Man: Essentials*, 8–65
detecting media errors, *Sys Mgr Man: Essentials*, 8–64
invoking with ANALYZE/MEDIA, *Sys Mgr Man: Essentials*, 8–65
BADLOG.SYS file
See Pending bad block log file
Banner pages
commands used with, *Sys Mgr Man: Essentials*, 13–60
definition, *Sys Mgr Man: Essentials*, 13–33
file, *Sys Mgr Man: Essentials*, 13–42, 13–60
job, *Sys Mgr Man: Essentials*, 13–42, 13–60
BASEENVIRON phase of system startup, *Sys Mgr Man: Essentials*, 5–4, 5–18
Base priority
See also Priority, base
Batch and print queuing system, *Sys Mgr Man: Essentials*, 12–2
See also Queue configurations
components, *Sys Mgr Man: Essentials*, 12–1
in VMScluster environments
with multiple system disk, *Sys Mgr Man: Essentials*, 12–6
queue database
location of files, *Sys Mgr Man: Essentials*, 12–5
mounting disk for, *Sys Mgr Man: Essentials*, 12–6
queuing process, *Sys Mgr Man: Essentials*, 13–2
Batch and print queuing system (cont'd)
sample configurations, *Sys Mgr Man: Essentials*, 13–4 to 13–14
starting in system startup, *Sys Mgr Man: Essentials*, 5–12
steps for setting up, *Sys Mgr Man: Essentials*, 13–14
Batch execution queues
See also Execution queues
description, *Sys Mgr Man: Essentials*, 13–3
Batch identifiers, *Sys Mgr Man: Essentials*, 11–11
Batch jobs
See also Batch processing environment
See also Batch queues
accessing devices, *Sys Mgr Man: Essentials*, 8–46
allowing to complete before stopping a queue, *Sys Mgr Man: Essentials*, 13–55
changing scheduling priority, *Sys Mgr Man: Essentials*, 13–72
completing before using VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–7
controlling, *Sys Mgr Man: Essentials*, 13–68
deleting, *Sys Mgr Man: Essentials*, 13–74
distributing system work load, *Sys Mgr Man: Tuning*, 16–3
executing, *Sys Mgr Man: Essentials*, 13–3
holding and releasing, *Sys Mgr Man: Essentials*, 13–70
job card, *Sys Mgr Man: Essentials*, 7–18
modifying, *Sys Mgr Man: Essentials*, 13–70
monitoring, *Sys Mgr Man: Essentials*, 13–68
requeuing
executing, *Sys Mgr Man: Essentials*, 13–72
pending, *Sys Mgr Man: Essentials*, 13–73
retaining in a queue, *Sys Mgr Man: Essentials*, 13–73
scheduling, *Sys Mgr Man: Essentials*, 13–72
status
See Job status
submitting at startup, *Sys Mgr Man: Essentials*, 5–14
Batch mode
as execution mode for a startup procedure, *Sys Mgr Man: Essentials*, 5–19
Batch processing environment
See also Batch jobs
See also Batch queues
generic queues in a VMScluster, *Sys Mgr Man: Essentials*, 13–7
on a standalone workstation, *Sys Mgr Man: Essentials*, 13–5
sample configurations, *Sys Mgr Man: Essentials*, 13–4 to 13–7
steps for setting up, *Sys Mgr Man: Essentials*, 13–14

Batch processing environment (cont'd)
with specialized queues, *Sys Mgr Man: Essentials*, 13–6
Batch queues
See also Batch jobs
See also Batch processing environment
allowing jobs to complete before stopping, *Sys Mgr Man: Essentials*, 13–55
commands for managing, *Sys Mgr Man: Essentials*, 13–46
creating, *Sys Mgr Man: Essentials*, 13–15
deleting, *Sys Mgr Man: Essentials*, 13–57
for memory constrained systems, *Sys Mgr Man: Essentials*, 13–31
on standalone workstations, *Sys Mgr Man: Essentials*, 13–5
optimizing for Sort/Merge utility, *Sys Mgr Man: Essentials*, 13–31
options, *Sys Mgr Man: Essentials*, 13–18
characteristics, *Sys Mgr Man: Essentials*, 13–28
controlling job performance and resources, *Sys Mgr Man: Essentials*, 13–28
qualifiers for specifying, *Sys Mgr Man: Essentials*, 13–19 to 13–20
restricting access, *Sys Mgr Man: Essentials*, 13–22
retaining jobs, *Sys Mgr Man: Essentials*, 13–26
pausing, *Sys Mgr Man: Essentials*, 13–53
setting up for vector processing, *Sys Mgr Man: Tuning*, 26–7
starting, *Sys Mgr Man: Essentials*, 13–15
status, *Sys Mgr Man: Essentials*, 13–50
stopping, *Sys Mgr Man: Essentials*, 13–54, 13–55
before shutting down a node, *Sys Mgr Man: Essentials*, 13–55
Battery-Backed Watch
setting time using, *Sys Mgr Man: Essentials*, 5–43
Binding volumes into a volume set, *Sys Mgr Man: Essentials*, 8–22
BIOLM process limit, *Sys Mgr Man: Essentials*, 6–42
value for efficient backups, *Sys Mgr Man: Essentials*, 10–10
BITMAP.SYS file
See Storage bit map file
Blocks
disk
definition, *Sys Mgr Man: Essentials*, 8–2
erasing, *Sys Mgr Man: Essentials*, 8–58
Files–11
definition, *Sys Mgr Man: Tuning*, A–1
tape
definition, *Sys Mgr Man: Essentials*, 8–6
Boot block
definition, *Sys Mgr Man: Essentials*, 4–17
in index file, *Sys Mgr Man: Tuning*, A–6
processors using, *Sys Mgr Man: Essentials*, 4–17
role in boot process, *Sys Mgr Man: Essentials*, 4–2
writing with Writeboot utility, *Sys Mgr Man: Essentials*, 4–17
Booting
automatically after shutting down, *Sys Mgr Man: Essentials*, 4–29
bootstrap image
Alpha, *Sys Mgr Man: Essentials*, 4–18
in index file, *Sys Mgr Man: Tuning*, A–6
VAX, *Sys Mgr Man: Essentials*, 4–17
conversationally
See Conversational boot
cross-architecture, *Sys Mgr Man: Tuning*, 22–23
definition, *Sys Mgr Man: Essentials*, 4–2
displaying startup commands, *Sys Mgr Man: Essentials*, 4–14
from an alternate system disk, *Sys Mgr Man: Essentials*, 4–3
in a multiprocessing system, *Sys Mgr Man: Tuning*, 26–2
in an emergency
with default system parameters, *Sys Mgr Man: Essentials*, 4–8
without startup and login procedures, *Sys Mgr Man: Essentials*, 4–8
without the user authorization file, *Sys Mgr Man: Essentials*, 4–10
installation of page and swap files, *Sys Mgr Man: Tuning*, 15–5
loading VVIEF code, *Sys Mgr Man: Tuning*, 26–5
location of computer-specific instructions, *Sys Mgr Man: Essentials*, 4–2
message
See Boot messages
nonstop, *Sys Mgr Man: Essentials*, 4–3
problems
fixing by booting with default parameter values, *Sys Mgr Man: Essentials*, 4–8
fixing by booting with minimum startup, *Sys Mgr Man: Essentials*, 4–14
hardware, *Sys Mgr Man: Essentials*, 4–16
invalid boot block, *Sys Mgr Man: Essentials*, 4–17
solving, *Sys Mgr Man: Essentials*, 4–16
use of boot block, *Sys Mgr Man: Essentials*, 4–17
with an alternate system parameter file, *Sys Mgr Man: Essentials*, 4–7

Booting (cont'd)
with controlled system startup, *Sys Mgr Man: Essentials*, 4–12
Boot messages
indicating execution of STARTUP.COM procedure, *Sys Mgr Man: Essentials*, 4–5
indicating execution of STARTUP_VMS.COM procedure, *Sys Mgr Man: Essentials*, 4–5
indicating success, *Sys Mgr Man: Essentials*, 4–5
indicating that login is possible, *Sys Mgr Man: Essentials*, 4–5
question mark (?), *Sys Mgr Man: Essentials*, 4–16
Bootstrapping
See Booting
BOT (beginning-of-tape)
See BOT markers
BOT markers, *Sys Mgr Man: Essentials*, 8–6
Break-ins
auditing attempts, *Sys Mgr Man: Tuning*, 18–26
Bridges
network, *Sys Mgr Man: Tuning*, 21–7
BROADCAST device setting, *Sys Mgr Man: Tuning*, 18–24
Buffered I/O
byte count limit, *Sys Mgr Man: Essentials*, 6–42
count limit, *Sys Mgr Man: Essentials*, 6–42
Buffers
using large buffers in LAT environment, *Sys Mgr Man: Tuning*, 24–13
Bugcheck message
during UETP, *Sys Mgr Man: Tuning*, 17–30
Burst bars, *Sys Mgr Man: Essentials*, 13–33
Burst pages, *Sys Mgr Man: Essentials*, 13–33
file, *Sys Mgr Man: Essentials*, 13–35
job, *Sys Mgr Man: Essentials*, 13–33
Busy queue status, *Sys Mgr Man: Essentials*, 13–51
BYTLM process limit, *Sys Mgr Man: Essentials*, 6–42
value for efficient backups/nomaster, *Sys Mgr Man: Essentials*, 10–10

C

C2-class system
installing OpenVMS to use, *Sys Mgr Man: Essentials*, 3–2
CACHE_SERVER process
creation during system startup, *Sys Mgr Man: Essentials*, 5–5
Caching
ACP system parameters, *Sys Mgr Man: Essentials*, 8–43
Canceling
characteristics on a queue, *Sys Mgr Man: Essentials*, 13–59
Capability of a vector
See Vector capability
Captive accounts, *Sys Mgr Man: Essentials*, 6–29
Card readers
operating, *Sys Mgr Man: Essentials*, 7–17
translation modes, *Sys Mgr Man: Essentials*, 7–19
Cards
decks, *Sys Mgr Man: Essentials*, 7–18
CD–ROM (compact disc read-only memory), *Sys Mgr Man: Essentials*, 8–4 to 8–5
accessing ISO 9660-formatted, *Sys Mgr Man: Essentials*, 8–34
accessing with the InfoServer Client for OpenVMS, *Sys Mgr Man: Tuning*, 23–10
characteristics of, *Sys Mgr Man: Essentials*, 8–4
data arrangement on, *Sys Mgr Man: Essentials*, 8–5
file structures, *Sys Mgr Man: Essentials*, 8–4
formats for automatic serving, *Sys Mgr Man: Tuning*, 23–2
High Sierra format, *Sys Mgr Man: Essentials*, 8–4
ISO 9660 format, *Sys Mgr Man: Essentials*, 8–4
media formats used, *Sys Mgr Man: Essentials*, 8–4
Changing Digital-supplied data in the Help Message database, *Sys Mgr Man: Essentials*, 5–31
Changing scheduling priority for a batch or print job, *Sys Mgr Man: Essentials*, 13–72
Changing size of page, swap, and dump files
recommended method, *Sys Mgr Man: Tuning*, 15–22
Changing system parameters
recommended method, *Sys Mgr Man: Tuning*, 14–4, 14–18
with AUTOGEN, *Sys Mgr Man: Tuning*, 14–18
with conversational boot, *Sys Mgr Man: Essentials*, 4–6; *Sys Mgr Man: Tuning*, 14–36
with SYSGEN, *Sys Mgr Man: Tuning*, 14–33, 14–34
with SYSMAN, *Sys Mgr Man: Tuning*, 14–29
Changing the DEFAULT form, *Sys Mgr Man: Essentials*, 13–63
Changing the system parameter file
with SYSGEN, *Sys Mgr Man: Tuning*, 14–33
with SYSMAN, *Sys Mgr Man: Tuning*, 14–28
Channels
network, *Sys Mgr Man: Tuning*, 21–5

CI (computer interconnect)
data link between cluster nodes, *Sys Mgr Man: Tuning*, 21–8
Circuits
network, *Sys Mgr Man: Tuning*, 21–5
definition, *Sys Mgr Man: Tuning*, 21–4
Classes
enabling and disabling, *Sys Mgr Man: Tuning*, 18–23
Class of data
in SHOW CLUSTER, *Sys Mgr Man: Tuning*, 20–4
removing, *Sys Mgr Man: Tuning*, 20–11
CLEAR command
NCP, *Sys Mgr Man: Tuning*, 21–11
Closed queue status, *Sys Mgr Man: Essentials*, 13–51
Closing a queue, *Sys Mgr Man: Essentials*, 13–53
Clusters
See also VAXcluster environments
See also VMScluster systems
See VAXcluster environments
See VMScluster systems
CLUSTER_CONFIG.COM command procedure, *Sys Mgr Man: Essentials*, 5–6; *Sys Mgr Man: Tuning*, 20–2
compared to VMSKITBLD.COM command procedure, *Sys Mgr Man: Essentials*, 2–31
creating SATELLITE_PAGE.COM command procedure, *Sys Mgr Man: Tuning*, 15–20
setting up LAN MOP, *Sys Mgr Man: Tuning*, 22–21
use in adding a system to a VMScluster, *Sys Mgr Man: Essentials*, 2–31
CLUSTER_SERVER process
creation during system startup, *Sys Mgr Man: Essentials*, 5–5
CLUSTER_SIZE attribute, *Sys Mgr Man: Essentials*, 10–57
Command files
LAN Control Program (LANCP) utility, *Sys Mgr Man: Tuning*, 22–9
Command formats
for backups, *Sys Mgr Man: Essentials*, 10–5
for image backups, *Sys Mgr Man: Essentials*, 10–31, 10–32
for incremental backups, *Sys Mgr Man: Essentials*, 10–34, 10–35
for multiple backup output devices, *Sys Mgr Man: Essentials*, 10–24, 10–31, 10–33
Command procedures
executing in SYSMAN, *Sys Mgr Man: Essentials*, 2–18
executing with SYSMAN DO command, *Sys Mgr Man: Essentials*, 2–18
for backups, *Sys Mgr Man: Essentials*, 10–36
Command procedures (cont'd)
for image backups, *Sys Mgr Man: Essentials*, 10–36
for incremental backups, *Sys Mgr Man: Essentials*, 10–37
for installing products, *Sys Mgr Man: Essentials*, 3–32
for interactive backups, *Sys Mgr Man: Essentials*, 10–39
for MONITOR
archiving recording and summary files, *Sys Mgr Man: Tuning*, 18–40
creating cluster summaries, *Sys Mgr Man: Tuning*, 18–41
creating summary file, *Sys Mgr Man: Tuning*, 18–40
generating clusterwide multifile summary reports, *Sys Mgr Man: Tuning*, 18–40
initiating continuous recording, *Sys Mgr Man: Tuning*, 18–42
invoking SUBMON.COM from startup, *Sys Mgr Man: Tuning*, 18–40
MONITOR.COM, *Sys Mgr Man: Tuning*, 18–40
MONSUM.COM, *Sys Mgr Man: Tuning*, 18–40
rerecording monitoring, *Sys Mgr Man: Tuning*, 18–40
starting MONITOR.COM as a detached process, *Sys Mgr Man: Tuning*, 18–40
SUBMON.COM, *Sys Mgr Man: Tuning*, 18–40
for POLYCENTER Software Installation utility, *Sys Mgr Man: Essentials*, 3–18
for SHOW CLUSTER, *Sys Mgr Man: Tuning*, 20–15
controlling output, *Sys Mgr Man: Tuning*, 20–5
default file type, *Sys Mgr Man: Tuning*, 20–16
description, *Sys Mgr Man: Tuning*, 20–16
formatting reports, *Sys Mgr Man: Tuning*, 20–10
initialization, *Sys Mgr Man: Tuning*, 20–14
SHOW_CLUSTER$INIT.COM, *Sys Mgr Man: Tuning*, 20–14
for system management (overview), *Sys Mgr Man: Essentials*, 2–5
for system startup, *Sys Mgr Man: Essentials*, 2–11
See also Startup command procedure
login
setting protection in, *Sys Mgr Man: Essentials*, 9–7
LOGIN.COM, *Sys Mgr Man: Essentials*, 2–16
setting up storage media, *Sys Mgr Man: Essentials*, 8–46

Command procedures
setting up storage media (cont'd)
disk volumes, *Sys Mgr Man: Essentials*, 8–46
tape volumes, *Sys Mgr Man: Essentials*, 8–47
testing a spooled printer, *Sys Mgr Man: Essentials*, 7–16
to configure a DECnet network node, *Sys Mgr Man: Tuning*, 21–11
to install software products
See VMSINSTAL.COM command procedure
to run AUTOGEN regularly, *Sys Mgr Man: Tuning*, 14–22
to specify default state of operator log files, *Sys Mgr Man: Tuning*, 18–23
VVIEF$INSTAL.COM procedure, *Sys Mgr Man: Tuning*, 26–5
Commands
See also DCL commands
executing on multiple nodes, *Sys Mgr Man: Essentials*, 2–18
executing on remote nodes, *Sys Mgr Man: Essentials*, 2–12
executing with SYSMAN DO command, *Sys Mgr Man: Essentials*, 2–18
Communicating
with operators, *Sys Mgr Man: Essentials*, 8–18
with users, *Sys Mgr Man: Essentials*, 8–18
creating systemwide announcements, *Sys Mgr Man: Essentials*, 5–14
Communications line
definition, *Sys Mgr Man: Tuning*, 21–12
network
definition, *Sys Mgr Man: Tuning*, 21–4
Compact disc drives
supported by UETP, *Sys Mgr Man: Tuning*, 17–7
Compact disc read-only memory
See CD–ROM
Compare operations
with BACKUP, *Sys Mgr Man: Essentials*, 10–25
Completion status
showing for batch and print jobs, *Sys Mgr Man: Essentials*, 13–26
Compressing the Help Message database after deletions, *Sys Mgr Man: Essentials*, 5–30
Computer interconnect
See CI
Configuration databases
building manually using NCP, *Sys Mgr Man: Tuning*, 21–11
changing location of file, *Sys Mgr Man: Tuning*, 21–10
contents of, *Sys Mgr Man: Tuning*, 21–10
Configuration databases (cont'd)
modifying using NCP, *Sys Mgr Man: Tuning*, 21–11
permanent database, *Sys Mgr Man: Tuning*, 21–11
stored in SYS$SYSTEM:NETNODE_REMOTE.DAT file, *Sys Mgr Man: Tuning*, 21–10
volatile database, *Sys Mgr Man: Tuning*, 21–11
Configurations
displaying LAN, *Sys Mgr Man: Tuning*, 22–9
for software product options, *Sys Mgr Man: Essentials*, 3–22
LAN, *Sys Mgr Man: Tuning*, 22–1
LAN firmware updates, *Sys Mgr Man: Tuning*, 22–14
network
DECnet support, *Sys Mgr Man: Tuning*, 21–6
local area LAN, *Sys Mgr Man: Tuning*, 21–6
multiple-area, *Sys Mgr Man: Tuning*, 21–6
single-area, *Sys Mgr Man: Tuning*, 21–6
queue
sample batch queuing system, *Sys Mgr Man: Essentials*, 13–4 to 13–7
sample print queuing system, *Sys Mgr Man: Essentials*, 13–8 to 13–14
CONFIGURATION SET command
in SYSMAN, *Sys Mgr Man: Tuning*, 20–16
CONFIGURATION SHOW command
in SYSMAN, *Sys Mgr Man: Tuning*, 20–16
CONFIGURE phase of system startup, *Sys Mgr Man: Essentials*, 5–4, 5–18, 7–9
definition, *Sys Mgr Man: Essentials*, 7–7
CONFIGURE process
starting, *Sys Mgr Man: Essentials*, 5–5
Configuring DECnet, *Sys Mgr Man: Tuning*, 21–7
configuration database, *Sys Mgr Man: Tuning*, 21–10
nodes
automatic, *Sys Mgr Man: Tuning*, 21–12
manual, *Sys Mgr Man: Tuning*, 21–12
planning network, *Sys Mgr Man: Tuning*, 21–12
Configuring devices
HSC
disabling during system startup, *Sys Mgr Man: Essentials*, 7–9
in system startup, *Sys Mgr Man: Essentials*, 7–7
in system startup, *Sys Mgr Man: Essentials*, 5–7, 7–6
CONINTERR.EXE driver, *Sys Mgr Man: Essentials*, 7–8
CONNECT command
See also IO CONNECT command

CONNECT command (cont'd)
in SYSGEN (VAX), *Sys Mgr Man: Essentials*, 7–7
for connecting the console device, *Sys Mgr Man: Essentials*, 7–8
in system startup, *Sys Mgr Man: Essentials*, 5–7
Connecting devices
automatically, *Sys Mgr Man: Essentials*, 7–6, 7–8
in system startup, *Sys Mgr Man: Essentials*, 5–4
manually, *Sys Mgr Man: Essentials*, 7–8
in system startup, *Sys Mgr Man: Essentials*, 5–7
on VAX, *Sys Mgr Man: Essentials*, 7–7
network communications device, *Sys Mgr Man: Essentials*, 7–9
the network communications device
on VAX, *Sys Mgr Man: Essentials*, 7–8
virtual terminals, *Sys Mgr Man: Essentials*, 7–11
CONSCOPY.COM command procedure, *Sys Mgr Man: Essentials*, 5–50
Console report during UETP, *Sys Mgr Man: Tuning*, 17–15
Console storage device
connecting (VAX), *Sys Mgr Man: Essentials*, 7–8
copying, *Sys Mgr Man: Essentials*, 5–50
use in booting, *Sys Mgr Man: Essentials*, 4–2
Console terminals, *Sys Mgr Man: Essentials*, 2–21, 2–23
message
indicating lack of installed page file, *Sys Mgr Man: Essentials*, 5–6
indicating site-independent startup, *Sys Mgr Man: Essentials*, 4–5
indicating site-specific startup, *Sys Mgr Man: Essentials*, 4–5
login welcome, *Sys Mgr Man: Essentials*, 2–10
Container files, *Sys Mgr Man: Essentials*, 3–20
CONTIN.SYS file
See Continuation file
Continuation file
CONTIN.SYS, *Sys Mgr Man: Tuning*, A–9
reserved file, *Sys Mgr Man: Tuning*, A–9
used as extension file identifier, *Sys Mgr Man: Tuning*, A–9
Continuation volumes
in volume set, *Sys Mgr Man: Essentials*, 8–40
mounting in tape volume sets, *Sys Mgr Man: Essentials*, 8–38
CONTINUE command
in conversational boot, *Sys Mgr Man: Essentials*, 4–6
Control access
for disk directory files, *Sys Mgr Man: Essentials*, 9–11
for disk files, *Sys Mgr Man: Essentials*, 9–6
Conversational boot
changing system parameters, *Sys Mgr Man: Essentials*, 4–6; *Sys Mgr Man: Tuning*, 14–36
CONTINUE command, *Sys Mgr Man: Essentials*, 4–6
location of computer-specific instructions, *Sys Mgr Man: Essentials*, 4–4
performing, *Sys Mgr Man: Essentials*, 4–3
SET command, *Sys Mgr Man: Essentials*, 4–6
SET/STARTUP command, *Sys Mgr Man: Essentials*, 4–12
SHOW command, *Sys Mgr Man: Essentials*, 4–6
showing system parameters, *Sys Mgr Man: Essentials*, 4–6; *Sys Mgr Man: Tuning*, 14–36
SHOW/STARTUP command, *Sys Mgr Man: Essentials*, 4–12
specifying an alternate startup command procedure, *Sys Mgr Man: Essentials*, 4–12
SYSBOOT prompt, *Sys Mgr Man: Essentials*, 4–3
tasks allowed in, *Sys Mgr Man: Essentials*, 4–3
USE command, *Sys Mgr Man: Essentials*, 4–7
uses of, *Sys Mgr Man: Essentials*, 4–3
with alternate system parameter file, *Sys Mgr Man: Essentials*, 4–7
with minimum startup, *Sys Mgr Man: Essentials*, 4–14
Convert utility (CONVERT)
saving the queue database, *Sys Mgr Man: Essentials*, 12–11
using to change organization of file, *Sys Mgr Man: Essentials*, 9–19
Coordinated Universal Time
See UTC service
COPY command, *Sys Mgr Man: Essentials*, 10–23; *Sys Mgr Man: Tuning*, 21–14
comparison with COPY command in System Dump Analyzer utility, *Sys Mgr Man: Tuning*, 15–15
disk volumes, *Sys Mgr Man: Essentials*, 9–19
in System Dump Analyzer utility, *Sys Mgr Man: Tuning*, 15–4, 15–15
restriction for copying dump files, *Sys Mgr Man: Tuning*, 15–15
sending message after file is copied, *Sys Mgr Man: Essentials*, 9–22
standard-labeled volumes
copying from, *Sys Mgr Man: Essentials*, 9–20
tape volumes

COPY command
tape volumes (cont'd)
copying files from, *Sys Mgr Man: Essentials*, 9–21
copying files to, *Sys Mgr Man: Essentials*, 9–20, 9–21
transferring information, *Sys Mgr Man: Essentials*, 9–19
Copying directories
with BACKUP, *Sys Mgr Man: Essentials*, 10–23
Copying files
dump files, *Sys Mgr Man: Tuning*, 15–14
from disk volumes, *Sys Mgr Man: Essentials*, 9–20
from tape volumes, *Sys Mgr Man: Essentials*, 9–20
methods for, *Sys Mgr Man: Essentials*, 9–19
to and from a remote host, *Sys Mgr Man: Tuning*, 21–14
to disk volumes, *Sys Mgr Man: Essentials*, 9–19
to tape volumes, *Sys Mgr Man: Essentials*, 9–21
using COPY command, *Sys Mgr Man: Essentials*, 9–19
using Exchange utility, *Sys Mgr Man: Essentials*, 9–24
with BACKUP, *Sys Mgr Man: Essentials*, 10–23
Copying software product release notes, *Sys Mgr Man: Essentials*, 3–29
Core image file
CORIMG.SYS, *Sys Mgr Man: Tuning*, A–9
not supported by OpenVMS, *Sys Mgr Man: Tuning*, A–9
CORIMG.SYS file
See Core image file
Counters
status of LAT node, *Sys Mgr Man: Tuning*, 24–15
CPUDEFAULT process limit
choosing a value for batch queues, *Sys Mgr Man: Essentials*, 13–31
specifying a value for batch queues, *Sys Mgr Man: Essentials*, 13–19, 13–29
CPU identification number, *Sys Mgr Man: Tuning*, 26–2
CPUMAXIMUM process limit
choosing a value for batch queues, *Sys Mgr Man: Essentials*, 13–31
specifying a value for batch queues, *Sys Mgr Man: Essentials*, 13–19, 13–29
CPU process limit, *Sys Mgr Man: Essentials*, 6–42
CRASH console command, *Sys Mgr Man: Essentials*, 4–27
Crash dumps, *Sys Mgr Man: Tuning*, 15–2
See also System dump files
See also System failures
comparison of physical and selective, *Sys Mgr Man: Tuning*, 15–3
freeing page file of, *Sys Mgr Man: Tuning*, 15–16
physical, *Sys Mgr Man: Tuning*, 15–3
releasing, *Sys Mgr Man: Tuning*, 15–16
requirements for saving, *Sys Mgr Man: Tuning*, 15–3
saving contents of page file on reboot, *Sys Mgr Man: Tuning*, 15–2
saving contents of system dump file on reboot, *Sys Mgr Man: Essentials*, 5–13
selective, *Sys Mgr Man: Tuning*, 15–3
Crash Log Utility Extractor (CLUE)
description, *Sys Mgr Man: Tuning*, 15–13
CREATE command
creating directories, *Sys Mgr Man: Essentials*, 9–20
limiting number of file versions, *Sys Mgr Man: Essentials*, 8–53
in SYSGEN
changing page, swap, and dump file sizes, *Sys Mgr Man: Tuning*, 15–26
creating page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–18
writing new file to tape volume, *Sys Mgr Man: Essentials*, 9–18
CREATE/DIRECTORY command
to specify UIC-based directory protection, *Sys Mgr Man: Essentials*, 9–12
Creating an additional queue manager, *Sys Mgr Man: Essentials*, 12–10
Creating a new system parameter file
with SYSGEN, *Sys Mgr Man: Tuning*, 14–35
Creating a PCF
before installation, *Sys Mgr Man: Essentials*, 3–22
during installation, *Sys Mgr Man: Essentials*, 3–30
Creating a queue database, *Sys Mgr Man: Essentials*, 12–7
Creating execution queues, *Sys Mgr Man: Essentials*, 13–15
autostart, *Sys Mgr Man: Essentials*, 13–15, 13–16
nonautostart, *Sys Mgr Man: Essentials*, 13–16
Creating generic queues, *Sys Mgr Man: Essentials*, 13–17
Creating log files
operator log file, *Sys Mgr Man: Tuning*, 18–22

Creating page, swap, and dump files
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–17, 15–24
with SYSGEN, *Sys Mgr Man: Tuning*, 15–18
Creating search path of Help Message database files, *Sys Mgr Man: Essentials*, 5–28
Cross-architecture booting, *Sys Mgr Man: Tuning*, 22–23
CTRLNAME logical name, *Sys Mgr Man: Tuning*, 17–32
Current accounting file
controlling which resources are tracked in, *Sys Mgr Man: Tuning*, 19–3
default file name, *Sys Mgr Man: Tuning*, 19–3
definition, *Sys Mgr Man: Tuning*, 19–1
finding out what resources are tracked in, *Sys Mgr Man: Tuning*, 19–2
moving, *Sys Mgr Man: Tuning*, 19–3
starting up a new version of, *Sys Mgr Man: Tuning*, 19–3
Current system parameters, *Sys Mgr Man: Tuning*, 14–3, 14–26
Customizing DECwindows Motif interface, *Sys Mgr Man: Essentials*, 3–39
Customizing the system
adding optional system files that have been removed from the system disk, *Sys Mgr Man: Essentials*, 5–1
backing up
console storage device, *Sys Mgr Man: Essentials*, 5–50
system disk, *Sys Mgr Man: Essentials*, 5–50
building standalone BACKUP, *Sys Mgr Man: Essentials*, 5–50
creating site-specific startup command procedures, *Sys Mgr Man: Essentials*, 5–2
creating systemwide announcements, *Sys Mgr Man: Essentials*, 5–14
duplicating the system disk, *Sys Mgr Man: Essentials*, 2–26
enabling autostart, *Sys Mgr Man: Essentials*, 5–12
installing known images, *Sys Mgr Man: Essentials*, 5–12
installing resident images (Alpha), *Sys Mgr Man: Essentials*, 5–13
limiting the number of interactive users, *Sys Mgr Man: Essentials*, 5–16
making remote InfoServer devices available, *Sys Mgr Man: Tuning*, 23–14
making remote InfoServer disks available, *Sys Mgr Man: Essentials*, 5–13
modifying login procedures, *Sys Mgr Man: Essentials*, 5–16
modifying site-specific startup command procedures, *Sys Mgr Man: Essentials*, 5–2
rules, *Sys Mgr Man: Essentials*, 5–4
Customizing the system
modifying site-specific startup command procedures (cont'd)
SYCONFIG.COM, *Sys Mgr Man: Essentials*, 5–7
SYLOGICALS.COM, *Sys Mgr Man: Essentials*, 5–8
SYPAGSWPFILES.COM, *Sys Mgr Man: Essentials*, 5–6
SYSECURITY.COM, *Sys Mgr Man: Essentials*, 5–9
SYSTARTUP_VMS.COM, *Sys Mgr Man: Essentials*, 5–10
removing optional system files from the system disk, *Sys Mgr Man: Essentials*, 5–1
setting up a LAT network, *Sys Mgr Man: Essentials*, 5–15
starting InfoServer Client for OpenVMS, *Sys Mgr Man: Essentials*, 5–13; *Sys Mgr Man: Tuning*, 23–10
starting queues, *Sys Mgr Man: Essentials*, 5–12
starting the DECnet network, *Sys Mgr Man: Essentials*, 5–15
submitting batch jobs at system startup, *Sys Mgr Man: Essentials*, 5–14
Cylinders
definition, *Sys Mgr Man: Tuning*, A–2

D

DAD virtual disk unit, *Sys Mgr Man: Tuning*, 23–12
Databases
See also Product database
configuration
See Configuration databases
LAT database, *Sys Mgr Man: Tuning*, 24–14
LMF
use in system startup, *Sys Mgr Man: Essentials*, 5–5
network
permanent, *Sys Mgr Man: Tuning*, 21–11
volatile, *Sys Mgr Man: Tuning*, 21–11
of software products, *Sys Mgr Man: Essentials*, 3–20
queue
See Queue database
startup
definition, *Sys Mgr Man: Essentials*, 5–18
layered product, *Sys Mgr Man: Essentials*, 5–18, 5–19
OpenVMS, *Sys Mgr Man: Essentials*, 5–18
order of startup events, *Sys Mgr Man: Essentials*, 5–4
Data blocks
partially recorded

Data blocks
partially recorded (cont'd)
ISO 9660 standard, *Sys Mgr Man: Essentials*, 8–5
Data card deck, *Sys Mgr Man: Essentials*, 7–18
Data interleaving
ISO 9660, *Sys Mgr Man: Essentials*, 8–6
Data loss
avoiding by dismounting volume, *Sys Mgr Man: Essentials*, 8–44
Date formats, *Sys Mgr Man: Essentials*, 5–43
predefined, *Sys Mgr Man: Essentials*, 5–48
specifying, *Sys Mgr Man: Essentials*, 5–46
DBBFs (detected bad block files), *Sys Mgr Man: Essentials*, 8–65
DCL commands
accessing disk and tape files, *Sys Mgr Man: Essentials*, 9–13
executing with SYSMAN DO command, *Sys Mgr Man: Essentials*, 2–18
file manipulation, *Sys Mgr Man: Essentials*, 9–1
for remote directory listings, *Sys Mgr Man: Tuning*, 21–14
for remote file access, *Sys Mgr Man: Tuning*, 21–14
for remote terminal service, *Sys Mgr Man: Tuning*, 21–13
for system management, *Sys Mgr Man: Essentials*, 2–5
retrieving file information, *Sys Mgr Man: Essentials*, 9–2
support for TCP/IP, *Sys Mgr Man: Tuning*, 21–13
with DO command in SYSMAN, *Sys Mgr Man: Tuning*, 20–20
DCL interface, *Sys Mgr Man: Essentials*, 3–18
DDCMP (Digital Data Communications Message Protocol)
for multipoint connections, *Sys Mgr Man: Tuning*, 21–9
for point-to-point connections, *Sys Mgr Man: Tuning*, 21–8
ddcu format
for device names, *Sys Mgr Man: Essentials*, 7–1
DEALLOCATE command, *Sys Mgr Man: Essentials*, 8–10
tape, *Sys Mgr Man: Essentials*, 9–23
Deallocating devices, *Sys Mgr Man: Essentials*, 8–10
Deallocating drives, *Sys Mgr Man: Essentials*, 8–10
DECdns naming service
storing full names in, *Sys Mgr Man: Tuning*, 21–2
DECdtm services
and managing transaction logs, *Sys Mgr Man: Tuning*, 25–1
disabling, *Sys Mgr Man: Tuning*, 25–20
enabling, *Sys Mgr Man: Tuning*, 25–21
DECevent utility
additional commands, *Sys Mgr Man: Tuning*, 18–11
description, *Sys Mgr Man: Tuning*, 18–2
DIAGNOSE command, *Sys Mgr Man: Tuning*, 18–2
exiting, *Sys Mgr Man: Tuning*, 18–9
introduction, *Sys Mgr Man: Tuning*, 18–8
invoking, *Sys Mgr Man: Tuning*, 18–9
privileges required, *Sys Mgr Man: Tuning*, 18–11
producing reports, *Sys Mgr Man: Tuning*, 18–11
qualifiers, *Sys Mgr Man: Tuning*, 18–10
report formats, *Sys Mgr Man: Tuning*, 18–9
brief, *Sys Mgr Man: Tuning*, 18–12
fast error (FSTERR), *Sys Mgr Man: Tuning*, 18–15
full, *Sys Mgr Man: Tuning*, 18–11
summary, *Sys Mgr Man: Tuning*, 18–14
terse, *Sys Mgr Man: Tuning*, 18–13
using to report on error log file, *Sys Mgr Man: Tuning*, 18–8
DECnet
See also Networks
adaptive routing, *Sys Mgr Man: Tuning*, 21–5
advantages, *Sys Mgr Man: Tuning*, 21–1
asynchronous
using virtual terminals, *Sys Mgr Man: Essentials*, 7–11, 7–12
circuit
definition, *Sys Mgr Man: Tuning*, 21–4
communications line
definition, *Sys Mgr Man: Tuning*, 21–4
configuration
automatic, *Sys Mgr Man: Tuning*, 21–12
manual, *Sys Mgr Man: Tuning*, 21–12
on an OpenVMS system, *Sys Mgr Man: Tuning*, 21–7
with bridge, *Sys Mgr Man: Tuning*, 21–7
configuration database, *Sys Mgr Man: Tuning*, 21–10
connecting with communications line, *Sys Mgr Man: Tuning*, 21–12
definition, *Sys Mgr Man: Tuning*, 21–4
end node, *Sys Mgr Man: Tuning*, 21–6
error message during UETP, *Sys Mgr Man: Tuning*, 17–28
establishing node in network, *Sys Mgr Man: Tuning*, 21–11
Ethernet, *Sys Mgr Man: Tuning*, 21–5
Event Logging facility

DECnet
Event Logging facility (cont'd)
to monitor network events, *Sys Mgr Man: Tuning*, 21–17
license, *Sys Mgr Man: Tuning*, 21–12
local area network (LAN) connections, *Sys Mgr Man: Tuning*, 21–8
logical link
definition, *Sys Mgr Man: Tuning*, 21–4
managing a network node, *Sys Mgr Man: Tuning*, 21–14
providing host services, *Sys Mgr Man: Tuning*, 21–15
managing remote nodes with, *Sys Mgr Man: Essentials*, 2–12
multiple-area network, *Sys Mgr Man: Tuning*, 21–6
network configurations, *Sys Mgr Man: Tuning*, 21–6
network connections, *Sys Mgr Man: Tuning*, 21–8
network interface for OpenVMS, *Sys Mgr Man: Tuning*, 21–7
network monitoring, *Sys Mgr Man: Tuning*, 21–14
See also Networks, monitoring
tools, *Sys Mgr Man: Tuning*, 21–15
nodes
definition, *Sys Mgr Man: Tuning*, 21–4
object definition, *Sys Mgr Man: Tuning*, 21–4
overriding AUTOGEN observations, *Sys Mgr Man: Tuning*, 14–21
PAK, *Sys Mgr Man: Tuning*, 21–12
Phase IV and Phase V compared, *Sys Mgr Man: Tuning*, 21–1
planning configuration, *Sys Mgr Man: Tuning*, 21–12
preparing for UETP, *Sys Mgr Man: Tuning*, 17–12
preparing system, *Sys Mgr Man: Tuning*, 21–12
router node, *Sys Mgr Man: Tuning*, 21–5
routing
definition, *Sys Mgr Man: Tuning*, 21–5
levels of, *Sys Mgr Man: Tuning*, 21–5
security, *Sys Mgr Man: Tuning*, 21–12
controlling access to node, *Sys Mgr Man: Tuning*, 21–12
protecting files, *Sys Mgr Man: Tuning*, 21–12
using proxy accounts, *Sys Mgr Man: Tuning*, 21–12
shutting down for software installation, *Sys Mgr Man: Essentials*, 3–7
specifying MAIL addresses, *Sys Mgr Man: Essentials*, 5–33
terminology, *Sys Mgr Man: Tuning*, 21–4
testing network, *Sys Mgr Man: Tuning*, 21–17
DECnet (cont'd)
tools
DTS/DTR, *Sys Mgr Man: Tuning*, 21–17
Monitor utility, *Sys Mgr Man: Tuning*, 21–17
NCP Ethernet configurator, *Sys Mgr Man: Tuning*, 21–17
UETP defaults for installation, *Sys Mgr Man: Tuning*, 17–30
UETP test phase, *Sys Mgr Man: Tuning*, 17–35, 17–36
using with EXCHANGE/NETWORK, *Sys Mgr Man: Essentials*, 9–24
verifying connection to the network, *Sys Mgr Man: Tuning*, 21–12
WAN (wide area network) connections, *Sys Mgr Man: Tuning*, 21–8
with VMScluster systems, *Sys Mgr Man: Tuning*, 21–6
worldwide connections, *Sys Mgr Man: Tuning*, 21–9
DECnet/OSI for OpenVMS, *Sys Mgr Man: Tuning*, 21–1
definition, *Sys Mgr Man: Tuning*, 21–1
distinguished from DECnet for OpenVMS, *Sys Mgr Man: Tuning*, 21–1
full names, *Sys Mgr Man: Tuning*, 21–2
starting, *Sys Mgr Man: Essentials*, 5–16
DECnet for OpenVMS, *Sys Mgr Man: Tuning*, 21–1
node names, *Sys Mgr Man: Tuning*, 21–2
starting, *Sys Mgr Man: Essentials*, 5–15
DECnet Test Sender/DECnet Test Receiver utility (DTS/DTR)
network monitoring tool, *Sys Mgr Man: Tuning*, 21–17
DECW$TAILOR
See Tailoring utilities
DECwindows
use of Snapshot facility with, *Sys Mgr Man: Essentials*, 4–21
DECwindows Motif interface, *Sys Mgr Man: Essentials*, 3–18
customizing your environment, *Sys Mgr Man: Essentials*, 3–39
POLYCENTER Software Installation utility, *Sys Mgr Man: Essentials*, 3–34
buttons, *Sys Mgr Man: Essentials*, 3–37, 3–39, 3–40
Deductible resource, *Sys Mgr Man: Essentials*, 6–3
DEFAULT account
in UAF, *Sys Mgr Man: Essentials*, 6–6
Default boot procedure, *Sys Mgr Man: Essentials*, 4–2

Default directories, *Sys Mgr Man: Essentials*, 6–15
Default form, *Sys Mgr Man: Essentials*, 13–63
DEFAULT form, *Sys Mgr Man: Essentials*, 13–63
Default protection
  UIC, *Sys Mgr Man: Essentials*, 9–7
Default system parameters
  booting with, *Sys Mgr Man: Essentials*, 4–8
DEFINE/CHARACTERISTIC command, *Sys Mgr Man: Essentials*, 13–58
DEFINE command
  NCP, *Sys Mgr Man: Tuning*, 21–11
DEFINE/FORM command, *Sys Mgr Man: Essentials*, 13–61
  for controlling line overflow, *Sys Mgr Man: Essentials*, 13–44
Defining a form, *Sys Mgr Man: Essentials*, 13–61
Defragmenting disks, *Sys Mgr Man: Essentials*, 10–47
Deinstalling layered software products, *Sys Mgr Man: Essentials*, 3–34
Delete access
  explicitly assigning, *Sys Mgr Man: Essentials*, 9–11
  for disk directory files, *Sys Mgr Man: Essentials*, 9–11
  for disk files, *Sys Mgr Man: Essentials*, 9–6
  granting through protection codes, *Sys Mgr Man: Essentials*, 11–8
DELETE/CHARACTERISTIC command, *Sys Mgr Man: Essentials*, 13–59
DELETE/ENTRY command, *Sys Mgr Man: Essentials*, 13–80
DELETE/FORM command, *Sys Mgr Man: Essentials*, 13–64
DELETE/QUEUE command, *Sys Mgr Man: Essentials*, 13–57
Deleting
  Digital messages from the Help Message database, *Sys Mgr Man: Essentials*, 5–29
  files from the system disk, *Sys Mgr Man: Essentials*, 5–1
  forms, *Sys Mgr Man: Essentials*, 13–64
    problems with, *Sys Mgr Man: Essentials*, 13–81
  jobs, *Sys Mgr Man: Essentials*, 13–74
  page, swap, and dump files
    after creating new version, *Sys Mgr Man: Tuning*, 15–28
    caution, *Sys Mgr Man: Tuning*, 15–21
  queue characteristics, *Sys Mgr Man: Essentials*, 13–59
    problems with, *Sys Mgr Man: Essentials*, 13–81
  queues, *Sys Mgr Man: Essentials*, 13–57
    problems with, *Sys Mgr Man: Essentials*, 13–81
Dependent software products, *Sys Mgr Man: Essentials*, 3–27
DESELECT command
  in SHOW CLUSTER, *Sys Mgr Man: Tuning*, 20–13
Despooling a spooled printer, *Sys Mgr Man: Essentials*, 7–16
Destination for installing software
  copying files, *Sys Mgr Man: Essentials*, 3–33
  PCSI$DESTINATION location, *Sys Mgr Man: Essentials*, 3–28
  specifying location, *Sys Mgr Man: Essentials*, 3–28
Destination parameter
  in VMSINSTAL.COM, *Sys Mgr Man: Essentials*, 3–12
Detected bad block files (DBBFs)
  See DBBFs
Device control libraries, *Sys Mgr Man: Essentials*, 13–44 to 13–46
  See also Device control modules
  assigning to a queue, *Sys Mgr Man: Essentials*, 13–66
  order of module output, *Sys Mgr Man: Essentials*, 13–45
  procedure for using, *Sys Mgr Man: Essentials*, 13–65
  sample commands, *Sys Mgr Man: Essentials*, 13–67
  setting up, *Sys Mgr Man: Essentials*, 13–44
Device control modules, *Sys Mgr Man: Essentials*, 13–45 to 13–83
  See also Device control libraries
  adding, *Sys Mgr Man: Essentials*, 13–66, 13–83
  creating, *Sys Mgr Man: Essentials*, 13–66
  deleting, *Sys Mgr Man: Essentials*, 13–66, 13–83
  forms, *Sys Mgr Man: Essentials*, 13–65, 13–67
  inserting into a library, *Sys Mgr Man: Essentials*, 13–66
  listing, *Sys Mgr Man: Essentials*, 13–67
  naming, *Sys Mgr Man: Essentials*, 13–66
  order of output, *Sys Mgr Man: Essentials*, 13–45
  page setup, *Sys Mgr Man: Essentials*, 13–45, 13–67
  requesting with PRINT command, *Sys Mgr Man: Essentials*, 13–67
  reset, *Sys Mgr Man: Essentials*, 13–67
    when queue is started, *Sys Mgr Man: Essentials*, 13–67
  sample commands, *Sys Mgr Man: Essentials*, 13–67
  setting up, *Sys Mgr Man: Essentials*, 13–65
  setup, *Sys Mgr Man: Essentials*, 13–45, 13–67
  specifying, *Sys Mgr Man: Essentials*, 13–45, 13–65

Device control modules (cont'd)
storing, *Sys Mgr Man: Essentials*, 13–66
troubleshooting, *Sys Mgr Man: Essentials*, 13–83
types, *Sys Mgr Man: Essentials*, 13–45
with forms, *Sys Mgr Man: Essentials*, 13–45
Device drivers
CONINTERR.EXE file, *Sys Mgr Man: Essentials*, 7–8
for event handling, *Sys Mgr Man: Essentials*, 7–8
loading
automatically, *Sys Mgr Man: Essentials*, 7–6
in system startup, *Sys Mgr Man: Essentials*, 5–4, 5–7
manually (Alpha), *Sys Mgr Man: Essentials*, 7–8
manually (VAX), *Sys Mgr Man: Essentials*, 7–7
not associated with a specific device, *Sys Mgr Man: Essentials*, 7–8
TTDRIVER, *Sys Mgr Man: Essentials*, 7–11
Device names, *Sys Mgr Man: Essentials*, 7–1
for virtual terminals, *Sys Mgr Man: Essentials*, 7–12
in a VMScluster environment, *Sys Mgr Man: Essentials*, 7–1
DEVICE phase of system startup, *Sys Mgr Man: Essentials*, 5–4, 5–18
Devices
accessing in batch job, *Sys Mgr Man: Essentials*, 8–46
allocating, *Sys Mgr Man: Essentials*, 8–9, 9–22
availability
OPCOM message, *Sys Mgr Man: Essentials*, 8–59
configuring
in system startup, *Sys Mgr Man: Essentials*, 5–7, 7–6
manually, *Sys Mgr Man: Essentials*, 5–7, 7–7
special devices, *Sys Mgr Man: Essentials*, 7–7, 7–8
connecting, *Sys Mgr Man: Essentials*, 7–7
determining available, *Sys Mgr Man: Essentials*, 7–2
Ethernet adapter
specifying number for AUTOGEN, *Sys Mgr Man: Tuning*, 14–21
getting information about, *Sys Mgr Man: Essentials*, 7–2, 7–17
ISO-9660
getting information about, *Sys Mgr Man: Essentials*, 7–5
LTA*n*, *Sys Mgr Man: Essentials*, 7–12
magnetic tape
See Tapes
Devices (cont'd)
managing, *Sys Mgr Man: Essentials*, 7–1
managing with LAN Control Program (LANCP) utility, *Sys Mgr Man: Tuning*, 22–9
manually configuring non-standard devices, *Sys Mgr Man: Essentials*, 5–7, 7–7
mounting volumes, *Sys Mgr Man: Essentials*, 8–18
network communications
connecting, *Sys Mgr Man: Essentials*, 7–9
connecting (VAX), *Sys Mgr Man: Essentials*, 7–8
not recognized by the system, *Sys Mgr Man: Essentials*, 7–7
OPA0:, *Sys Mgr Man: Essentials*, 2–23
printers
See Printers
protecting, *Sys Mgr Man: Essentials*, 7–5
requiring manual connecting, *Sys Mgr Man: Essentials*, 7–9
resetting default in SYSMAN, *Sys Mgr Man: Essentials*, 2–17
RTA*n*, *Sys Mgr Man: Essentials*, 7–12
setting characteristics, *Sys Mgr Man: Essentials*, 13–14
in system startup, *Sys Mgr Man: Essentials*, 13–14
special
connecting, *Sys Mgr Man: Essentials*, 7–8
spooled, *Sys Mgr Man: Essentials*, 13–14
status report on, *Sys Mgr Man: Tuning*, 18–17
suppressing autoconfiguration during system startup, *Sys Mgr Man: Essentials*, 5–8, 7–9
terminals
See Terminals
Device tests
running individually with UETP, *Sys Mgr Man: Tuning*, 17–32
Device unavailable queue status, *Sys Mgr Man: Essentials*, 13–51
DIAGNOSE command, *Sys Mgr Man: Tuning*, 18–2
Diagnostics
relationship to UETP, *Sys Mgr Man: Tuning*, 17–16
Dialup identifiers, *Sys Mgr Man: Essentials*, 11–11
DIBOL
starting the message manager, *Sys Mgr Man: Essentials*, 5–16
Digital Data Communications Message Protocol
See DDCMP
Digital System Identifier
See DSI

DIOLM process limit, *Sys Mgr Man: Essentials*, 6–42
value for efficient backups, *Sys Mgr Man: Essentials*, 10–10
Direct I/O count process limit, *Sys Mgr Man: Essentials*, 6–42
Direct mode
as execution mode for a startup procedure, *Sys Mgr Man: Essentials*, 5–19
Directories
backing up, *Sys Mgr Man: Essentials*, 10–24
backlink, *Sys Mgr Man: Essentials*, 8–56
copying with BACKUP, *Sys Mgr Man: Essentials*, 10–23
creating, *Sys Mgr Man: Essentials*, 9–20
destination
specifying in VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–13
for an interactive account, *Sys Mgr Man: Essentials*, 6–15
levels of access in restore operations, *Sys Mgr Man: Essentials*, 10–29
protecting, *Sys Mgr Man: Essentials*, 9–11, 9–12
resetting default in SYSMAN, *Sys Mgr Man: Essentials*, 2–17
temporary working
in VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–14
DIRECTORY command, *Sys Mgr Man: Essentials*, 9–5
checking number of user's disk blocks, *Sys Mgr Man: Essentials*, 8–49
retrieving file information, *Sys Mgr Man: Essentials*, 9–2
showing full information, *Sys Mgr Man: Essentials*, 9–17
to obtain product list, *Sys Mgr Man: Essentials*, 3–10
with tapes, *Sys Mgr Man: Essentials*, 9–16, 9–20
Directory trees
copying, *Sys Mgr Man: Essentials*, 10–23
DIR/FTP command, *Sys Mgr Man: Tuning*, 21–14
DISABLE AUTOSTART/QUEUES command, *Sys Mgr Man: Essentials*, 13–54, 13–55
entering before shutting down a system, *Sys Mgr Man: Essentials*, 13–55
relationship to STOP/QUEUES/ON_NODE command, *Sys Mgr Man: Essentials*, 13–55
Disabled queues, *Sys Mgr Man: Essentials*, 13–84
Disabling autostart, *Sys Mgr Man: Essentials*, 12–9, 13–54
before shutting down a node, *Sys Mgr Man: Essentials*, 13–55
Disabling operator terminals, *Sys Mgr Man: Essentials*, 2–24
Disabling spooling, *Sys Mgr Man: Essentials*, 7–16
Disabling user accounts, *Sys Mgr Man: Essentials*, 6–27
Disk
See System disks
See User disks
Disk files
accessing at file level, *Sys Mgr Man: Essentials*, 9–13
assigning an alias, *Sys Mgr Man: Essentials*, 9–10
copying
from disk volumes, *Sys Mgr Man: Essentials*, 9–19
to tapes, *Sys Mgr Man: Essentials*, 9–21
using COPY command, *Sys Mgr Man: Essentials*, 9–18
modifying characteristics, *Sys Mgr Man: Essentials*, 9–8, 9–10
DISKQUOTA commands
See also Disk Quota utility
DISKQUOTA commands in SYSMAN, *Sys Mgr Man: Essentials*, 2–11
DISKQUOTA/DISABLE command, *Sys Mgr Man: Essentials*, 8–52
DISKQUOTA/ENABLE command, *Sys Mgr Man: Essentials*, 8–52
Disk quotas, *Sys Mgr Man: Essentials*, 6–15
See also Disk Quota utility
creating, *Sys Mgr Man: Essentials*, 8–51
definition, *Sys Mgr Man: Essentials*, 8–49
disabling, *Sys Mgr Man: Essentials*, 8–53
displaying, *Sys Mgr Man: Essentials*, 8–49
ensuring accuracy with rebuild operation, *Sys Mgr Man: Essentials*, 8–50
establishing, *Sys Mgr Man: Essentials*, 8–50, 8–51
example, *Sys Mgr Man: Essentials*, 6–33
exceeding, *Sys Mgr Man: Essentials*, 8–50
file, *Sys Mgr Man: Essentials*, 8–49
maintaining, *Sys Mgr Man: Essentials*, 8–50
retrieving information, *Sys Mgr Man: Essentials*, 8–52
suspending operations, *Sys Mgr Man: Essentials*, 8–52
Disk Quota utility (DISKQUOTA), *Sys Mgr Man: Essentials*, 8–50
Disks
See also CD–ROMs
See also Disk commands
See also Disk files
See also Disk quotas
See also Disk volumes
See also System disks

Disks (cont'd)
allocating drives, *Sys Mgr Man: Essentials*, 8–9
allocating space on, *Sys Mgr Man: Essentials*, 8–52
backing up active, *Sys Mgr Man: Essentials*, 10–63
block
definition, *Sys Mgr Man: Essentials*, 8–2
grouped into cluster, *Sys Mgr Man: Tuning*, A–1
changing volume label, *Sys Mgr Man: Essentials*, 3–33
cluster, *Sys Mgr Man: Tuning*, A–1
definition, *Sys Mgr Man: Essentials*, 8–2
concepts, *Sys Mgr Man: Tuning*, A–1
cylinder
definition, *Sys Mgr Man: Tuning*, A–2
deallocating drives, *Sys Mgr Man: Essentials*, 8–10
default format, *Sys Mgr Man: Essentials*, 9–19
definition, *Sys Mgr Man: Essentials*, 8–2
dismounting, *Sys Mgr Man: Essentials*, 10–16
extents, *Sys Mgr Man: Tuning*, A–1
definition, *Sys Mgr Man: Essentials*, 8–2
file identification, *Sys Mgr Man: Tuning*, A–4
files
See Disk files
Files–11
directory hierarchy, *Sys Mgr Man: Tuning*, A–4
fragmentation of, *Sys Mgr Man: Essentials*, 10–47; *Sys Mgr Man: Tuning*, 15–27
I/O performance, *Sys Mgr Man: Tuning*, 16–8
initializing, *Sys Mgr Man: Essentials*, 10–14
mounting, *Sys Mgr Man: Essentials*, 8–28, 10–15
mounting in host-based shadow set, *Sys Mgr Man: Essentials*, 10–41
organization
logical, *Sys Mgr Man: Tuning*, A–1
physical, *Sys Mgr Man: Tuning*, A–2
protecting, *Sys Mgr Man: Essentials*, 8–14, 8–16
space
See Disk space
structure
See Disk structure
system
See System disks
terminology, *Sys Mgr Man: Essentials*, 8–2
track
definition, *Sys Mgr Man: Tuning*, A–2
usage, *Sys Mgr Man: Essentials*, 8–49
creating file, *Sys Mgr Man: Essentials*, 8–58
Disks
usage (cont'd)
UICs kept in quota file, *Sys Mgr Man: Tuning*, A–9
volumes
definitions, *Sys Mgr Man: Essentials*, 8–2
Disk space
See also Disk quotas
allocation by cluster, *Sys Mgr Man: Tuning*, A–1
managing, *Sys Mgr Man: Essentials*, 8–48 to 8–55
purging files, *Sys Mgr Man: Essentials*, 8–53
saving, *Sys Mgr Man: Essentials*, 8–53
by moving page and swap files off the system disk, *Sys Mgr Man: Tuning*, 15–5
by purging the operator log file, *Sys Mgr Man: Essentials*, 5–14
by removing optional system files, *Sys Mgr Man: Essentials*, 5–1
by storing minimal dump information, *Sys Mgr Man: Tuning*, 15–11
by using a selective dump, *Sys Mgr Man: Tuning*, 15–3, 15–4
tracking use of, *Sys Mgr Man: Tuning*, 19–6
Disk storage server, *Sys Mgr Man: Tuning*, 23–10
Disk structure
analyzing errors, *Sys Mgr Man: Essentials*, 8–55
Files–11, *Sys Mgr Man: Tuning*, A–3
reporting and repairing errors, *Sys Mgr Man: Essentials*, 8–56
Disk volumes
accessing files on, *Sys Mgr Man: Essentials*, 9–13
access to public, *Sys Mgr Man: Essentials*, 8–8
adding to an existing set, *Sys Mgr Man: Essentials*, 8–34
adding volumes to volume sets, *Sys Mgr Man: Essentials*, 8–34
analyzing disk structure errors, *Sys Mgr Man: Essentials*, 8–55
analyzing media errors, *Sys Mgr Man: Essentials*, 8–64
assigning logical name, *Sys Mgr Man: Essentials*, 5–11
assigning volume label, *Sys Mgr Man: Essentials*, 5–11
binding into volume sets, *Sys Mgr Man: Essentials*, 8–30
characteristics
modifying, *Sys Mgr Man: Essentials*, 8–29
console, *Sys Mgr Man: Essentials*, 9–24
copying files from, *Sys Mgr Man: Essentials*, 9–20
copying files to and from foreign volumes, *Sys Mgr Man: Essentials*, 9–24

Disk volumes (cont'd)
copying files to tape volumes, *Sys Mgr Man: Essentials*, 9–22
creating Files–11 structure, *Sys Mgr Man: Essentials*, 8–10
creating shadowed volume sets, *Sys Mgr Man: Essentials*, 8–33
creating volume sets from, *Sys Mgr Man: Essentials*, 8–32
definition, *Sys Mgr Man: Essentials*, 8–2; *Sys Mgr Man: Tuning*, A–1
disk quota operations, *Sys Mgr Man: Essentials*, 8–50
dismounting, *Sys Mgr Man: Essentials*, 8–43
file expiration dates
setting, *Sys Mgr Man: Essentials*, 8–54
file-structured, *Sys Mgr Man: Essentials*, 8–20
foreign, *Sys Mgr Man: Essentials*, 8–20
handling error conditions, *Sys Mgr Man: Essentials*, 8–55
initializing, *Sys Mgr Man: Essentials*, 8–10; *Sys Mgr Man: Tuning*, A–1
guidelines, *Sys Mgr Man: Essentials*, 8–13
load balancing, *Sys Mgr Man: Essentials*, 8–9
modifying characteristics, *Sys Mgr Man: Essentials*, 8–29
mounting, *Sys Mgr Man: Essentials*, 8–21, 8–27
for page and swap files, *Sys Mgr Man: Tuning*, 15–20
for queue database files, *Sys Mgr Man: Essentials*, 12–6
in system startup, *Sys Mgr Man: Essentials*, 5–11
early, *Sys Mgr Man: Essentials*, 5–12
for page and swap files, *Sys Mgr Man: Essentials*, 5–6
InfoServer, *Sys Mgr Man: Tuning*, 23–14
MOUNT/ASSIST command, *Sys Mgr Man: Essentials*, 5–11
special consideration about operator assistance, *Sys Mgr Man: Essentials*, 5–11
mount verification, *Sys Mgr Man: Essentials*, 8–58
performance, *Sys Mgr Man: Essentials*, 8–28
physical loading, *Sys Mgr Man: Essentials*, 8–21
private, *Sys Mgr Man: Essentials*, 8–9
protecting, *Sys Mgr Man: Essentials*, 8–14
public
See Public volumes
reading files from, *Sys Mgr Man: Essentials*, 9–13
rebuilding, *Sys Mgr Man: Essentials*, 8–50
removing before dismounting, *Sys Mgr Man: Essentials*, 8–43
Disk volumes (cont'd)
space
conserving, *Sys Mgr Man: Essentials*, 8–48
verification, *Sys Mgr Man: Essentials*, 8–58
write-locked
dismounting, *Sys Mgr Man: Essentials*, 8–43
write-locking, *Sys Mgr Man: Essentials*, 8–60
writing files to, *Sys Mgr Man: Essentials*, 9–20
DISMOUNT command
See also Dismounting
canceling mount verification, *Sys Mgr Man: Essentials*, 8–62
dismounting single volume in volume set, *Sys Mgr Man: Essentials*, 8–45
for a single tape volume, *Sys Mgr Man: Essentials*, 8–44
for foreign volumes, *Sys Mgr Man: Essentials*, 8–45
for single volume or volume set, *Sys Mgr Man: Essentials*, 8–42
overriding automatic unloading of volume, *Sys Mgr Man: Essentials*, 8–45
tape, *Sys Mgr Man: Essentials*, 9–23
Dismounting
See also DISMOUNT command
a backup volume, *Sys Mgr Man: Essentials*, 10–16
a disk with DECdtm transaction logs, *Sys Mgr Man: Tuning*, 25–14
system disk, *Sys Mgr Man: Essentials*, 8–43
volumes
conditions preventing, *Sys Mgr Man: Essentials*, 8–43
foreign, *Sys Mgr Man: Essentials*, 8–45
in a VMScluster system, *Sys Mgr Man: Essentials*, 8–45
with cached information, *Sys Mgr Man: Essentials*, 8–43
with open files, *Sys Mgr Man: Essentials*, 8–43
volume sets, *Sys Mgr Man: Essentials*, 8–45
Displaying
characteristics assigned to a queue, *Sys Mgr Man: Essentials*, 13–59
defined characteristics, *Sys Mgr Man: Essentials*, 13–59
defined forms, *Sys Mgr Man: Essentials*, 13–62
forms assigned to queues, *Sys Mgr Man: Essentials*, 13–64
information about queues, *Sys Mgr Man: Essentials*, 13–49
information about the queue manager, *Sys Mgr Man: Essentials*, 12–11
system parameters
See Showing system parameters

Distributed Queuing System
See DQS
Distributing system work load, *Sys Mgr Man: Tuning*, 16–3
Distribution kit
startup files included on, *Sys Mgr Man: Essentials*, 5–2
DO command
for managing a VMScluster system, *Sys Mgr Man: Tuning*, 20–20
in SYSMAN, *Sys Mgr Man: Essentials*, 2–18
DOS–11 tape volumes
file transfers with, *Sys Mgr Man: Essentials*, 9–24
format conversions for, *Sys Mgr Man: Essentials*, 9–24
Downline loading, *Sys Mgr Man: Tuning*, 23–2
DQS (Distributed Queuing System)
distributed printing, *Sys Mgr Man: Essentials*, 13–13
Drivers
See Device drivers
DS3 connection, *Sys Mgr Man: Tuning*, 21–9
DSA device naming, *Sys Mgr Man: Essentials*, 7–1
DSI (Digital System Identifier)
ISO 9660 media protection, *Sys Mgr Man: Essentials*, 8–18
mount option, *Sys Mgr Man: Essentials*, 8–17
DSI keyword
with MOUNT/PROTECTION command, *Sys Mgr Man: Essentials*, 8–23
DTS/DTR
See DECnet Test Sender/DECnet Test Receiver utility
Dual-architecture VMScluster systems
installing images, *Sys Mgr Man: Tuning*, 20–21
example, *Sys Mgr Man: Tuning*, 20–21
DUMPBUG system parameter, *Sys Mgr Man: Tuning*, 15–3
Dump file information
saving automatically, *Sys Mgr Man: Tuning*, 15–12
Dump file off system disk, *Sys Mgr Man: Tuning*, 15–14
requirements, *Sys Mgr Man: Tuning*, 15–14
Dump files
changing sizes
with SWAPFILES.COM, *Sys Mgr Man: Tuning*, 15–25
system
See System dump files
DUMPFILE symbol, *Sys Mgr Man: Tuning*, 15–24
DUMPSTYLE system parameter, *Sys Mgr Man: Tuning*, 15–3, 15–7, 15–11
Dynamic load balancing, *Sys Mgr Man: Tuning*, 26–2
Dynamic system parameters, *Sys Mgr Man: Tuning*, 14–2, 14–3
See also System parameters

E

EDIT keypad function, *Sys Mgr Man: Tuning*, 20–7
Emergency system shutdown
with console commands, *Sys Mgr Man: Essentials*, 4–27
with OPCCRASH, *Sys Mgr Man: Essentials*, 4–27, 4–36
Emergency system startup
with default system parameters, *Sys Mgr Man: Essentials*, 4–8
without startup and login procedures, *Sys Mgr Man: Essentials*, 4–8
without the UAF, *Sys Mgr Man: Essentials*, 4–10
ENABLE AUTOSTART/QUEUES command, *Sys Mgr Man: Essentials*, 13–18, 13–48
in startup command procedure, *Sys Mgr Man: Essentials*, 13–16
recommended use, *Sys Mgr Man: Essentials*, 13–18
Enabling autostart, *Sys Mgr Man: Essentials*, 13–4, 13–18, 13–48
in startup command procedure, *Sys Mgr Man: Essentials*, 13–16
End nodes
network, *Sys Mgr Man: Tuning*, 21–6
END phase
of system startup, *Sys Mgr Man: Essentials*, 5–19
ENQLM process limit, *Sys Mgr Man: Essentials*, 6–43
EOT markers, *Sys Mgr Man: Essentials*, 8–6
continuing to copy after, *Sys Mgr Man: Essentials*, 9–24
Erasing blocks, *Sys Mgr Man: Essentials*, 8–58
ERLBUFFERPAGES system parameter, *Sys Mgr Man: Tuning*, 15–6
ERRFMT
See Error Formatter
ERRFMT process, *Sys Mgr Man: Tuning*, 18–2
See also Error log files
See also Error logging
See also Error Log utility
creation during system startup, *Sys Mgr Man: Essentials*, 5–5
restarting, *Sys Mgr Man: Tuning*, 18–3

ERRFMT process (cont'd)
sending mail when it is deleted, *Sys Mgr Man: Tuning*, 18–4
writes to ERRLOG.SYS file, *Sys Mgr Man: Tuning*, 18–2
Error checking
in SYSTARTUP_VMS.COM, *Sys Mgr Man: Essentials*, 5–10
in system parameter files, *Sys Mgr Man: Tuning*, 14–19
Error Formatter
changing mail username, *Sys Mgr Man: Tuning*, 18–5
description, *Sys Mgr Man: Tuning*, 18–2
disabling mail, *Sys Mgr Man: Tuning*, 18–4, 18–5
enabling mail, *Sys Mgr Man: Tuning*, 18–5
notifying user with mail message, *Sys Mgr Man: Tuning*, 18–4
using to send mail, *Sys Mgr Man: Tuning*, 18–4
ERRORLOGBUFFERS system parameter, *Sys Mgr Man: Tuning*, 15–6
Error log files
created by ERRFMT process, *Sys Mgr Man: Tuning*, 18–2
events reported in, *Sys Mgr Man: Tuning*, 18–5
logical name defining location, *Sys Mgr Man: Essentials*, 5–8
maintaining, *Sys Mgr Man: Tuning*, 18–3
moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
producing full report, *Sys Mgr Man: Tuning*, 18–6
reporting on using DECevent, *Sys Mgr Man: Tuning*, 18–8
SYSPRV privilege to access, *Sys Mgr Man: Tuning*, 18–6
Error logging
See also ERRFMT process
See also Error log files
See also Error Log utility
description, *Sys Mgr Man: Tuning*, 18–2
reports produced, *Sys Mgr Man: Tuning*, 18–3
using the Error Log utility, *Sys Mgr Man: Tuning*, 18–2
Error logging facility, *Sys Mgr Man: Tuning*, 18–2
Error Log Report Formatter (ERF), *Sys Mgr Man: Tuning*, 18–2
Error log reports
See Error Log utility, reports
Error Log utility (ERROR LOG)
See also ERRFMT process
See also Error log files
See also Error logging
Error Log utility (ERROR LOG) (cont'd)
ANALYZE/ERROR_LOG command, *Sys Mgr Man: Tuning*, 18–6
definition, *Sys Mgr Man: Tuning*, 18–5
relationship to UETP, *Sys Mgr Man: Tuning*, 17–3, 17–17, 17–29
reporting on error log file, *Sys Mgr Man: Tuning*, 18–5
reports
excluding unknown entries, *Sys Mgr Man: Tuning*, 18–8
formats, *Sys Mgr Man: Tuning*, 18–7
printing, *Sys Mgr Man: Tuning*, 18–7
privileges required, *Sys Mgr Man: Tuning*, 18–7
specifying display device, *Sys Mgr Man: Tuning*, 18–8
specifying events and times, *Sys Mgr Man: Tuning*, 18–8
uses, *Sys Mgr Man: Tuning*, 18–5
report types, *Sys Mgr Man: Tuning*, 18–5
Error messages, *Sys Mgr Man: Essentials*, 2–5
See also Help Message utility
See also Messages
Error options
for fatal BACKUP errors, *Sys Mgr Man: Essentials*, 10–64
Errors
analyzing disk structure, *Sys Mgr Man: Essentials*, 8–55
analyzing error reports, *Sys Mgr Man: Tuning*, 18–2
analyzing media, *Sys Mgr Man: Essentials*, 8–64
disk read
if returned when booting, *Sys Mgr Man: Essentials*, 4–16
disk structure
repairing, *Sys Mgr Man: Essentials*, 8–57
reporting, *Sys Mgr Man: Essentials*, 8–57
error log file, *Sys Mgr Man: Tuning*, 18–2
error logging facility, *Sys Mgr Man: Tuning*, 18–2
handling on disk volumes, *Sys Mgr Man: Essentials*, 8–55
machine check
if returned when booting, *Sys Mgr Man: Essentials*, 4–16
mounting disk, *Sys Mgr Man: Essentials*, 8–22
Errors during UETP, *Sys Mgr Man: Tuning*, 17–21
diagnosing, *Sys Mgr Man: Tuning*, 17–16
sources of, *Sys Mgr Man: Tuning*, 17–17
ESS$LASTDRIVER device driver, *Sys Mgr Man: Tuning*, 23–9, 23–13
controlling and diagnosing, *Sys Mgr Man: Tuning*, 23–9, 23–10

ESS$STARTUP.COM command procedure, *Sys Mgr Man: Tuning*, 23–10, 23–13
invoking in system startup, *Sys Mgr Man: Tuning*, 23–10
Ethernet
adapters
specifying number of in MODPARAMS.DAT file, *Sys Mgr Man: Tuning*, 14–21
configurator
network monitoring tool, *Sys Mgr Man: Tuning*, 21–17
connecting, *Sys Mgr Man: Tuning*, 21–7
defining a remote node for UETP, *Sys Mgr Man: Tuning*, 17–19
definition, *Sys Mgr Man: Tuning*, 21–5
device drivers, *Sys Mgr Man: Tuning*, 22–2
in local area network, *Sys Mgr Man: Tuning*, 21–8
linking with bridge, *Sys Mgr Man: Tuning*, 21–7
preparing for UETP, *Sys Mgr Man: Tuning*, 17–8
Event classes
displaying those being audited, *Sys Mgr Man: Tuning*, 18–27
Event handling
device driver used in, *Sys Mgr Man: Essentials*, 7–8
EXCHANGE/NETWORK command
using to transfer files, *Sys Mgr Man: Essentials*, 9–19, 9–24
Exchange utility (EXCHANGE)
using to copy files, *Sys Mgr Man: Essentials*, 9–19
using to transfer information, *Sys Mgr Man: Essentials*, 9–24
Executable images, *Sys Mgr Man: Tuning*, 16–9, 16–14, 16–15
Execute access
for disk directory files, *Sys Mgr Man: Essentials*, 9–11
for disk files, *Sys Mgr Man: Essentials*, 9–6
granting through protection codes, *Sys Mgr Man: Essentials*, 11–8
Execute procedure (@) command, *Sys Mgr Man: Tuning*, 20–16
Executing job status, *Sys Mgr Man: Essentials*, 13–69, 13–74
Execution modes
startup procedures, *Sys Mgr Man: Essentials*, 5–19
BATCH, *Sys Mgr Man: Essentials*, 5–19
changing, *Sys Mgr Man: Essentials*, 5–21
DIRECT, *Sys Mgr Man: Essentials*, 5–19
SPAWN, *Sys Mgr Man: Essentials*, 5–19
specifying, *Sys Mgr Man: Essentials*, 5–20
Execution queues
activating autostart, *Sys Mgr Man: Essentials*, 13–4, 13–15, 13–49
creating, *Sys Mgr Man: Essentials*, 13–15
relationship to generic queues, *Sys Mgr Man: Essentials*, 13–2
starting
autostart, *Sys Mgr Man: Essentials*, 13–4, 13–18, 13–48
in system startup, *Sys Mgr Man: Essentials*, 5–12
nonautostart, *Sys Mgr Man: Essentials*, 13–16, 13–18, 13–48
Executive mode
calling images running in, *Sys Mgr Man: Tuning*, 16–11, 16–14
logical names, *Sys Mgr Man: Tuning*, 16–15
recommended use for logical names, *Sys Mgr Man: Essentials*, 5–9
Expiration date, *Sys Mgr Man: Essentials*, 6–43
field, *Sys Mgr Man: Essentials*, 9–14
checking, *Sys Mgr Man: Essentials*, 9–18
file, *Sys Mgr Man: Essentials*, 8–54
tape file system checks, *Sys Mgr Man: Essentials*, 9–17
Expiration time, *Sys Mgr Man: Essentials*, 6–43
Extended attribute records
See XARs
Extensions
file
See File extensions
Extents
definition, *Sys Mgr Man: Tuning*, A–1
disk
definition, *Sys Mgr Man: Essentials*, 8–2
index file
definition, *Sys Mgr Man: Tuning*, A–7
Extracting software product release notes, *Sys Mgr Man: Essentials*, 3–29

F

F$GETJPI lexical function
getting information about vector processing, *Sys Mgr Man: Tuning*, 26–9
F$GETQUI lexical function, *Sys Mgr Man: Essentials*, 13–51
F$GETSYI lexical function
getting information about vector processing, *Sys Mgr Man: Tuning*, 26–9
Failover list
for an autostart queue
specifying, *Sys Mgr Man: Essentials*, 13–15
for queue manager, *Sys Mgr Man: Essentials*, 12–3, 12–9
insufficient, *Sys Mgr Man: Essentials*, 12–17

Failover list
for queue manager (cont'd)
specifying, *Sys Mgr Man: Essentials*, 12–9
Failovers
See also Failover list
of queue manager, *Sys Mgr Man: Essentials*, 12–3, 12–17
forcing, *Sys Mgr Man: Essentials*, 12–9
of queues, *Sys Mgr Man: Essentials*, 13–4, 13–15
FDDI (Fiber Distributed Data Interface)
device drivers, *Sys Mgr Man: Tuning*, 22–2
multiaccess communications channel, *Sys Mgr Man: Tuning*, 21–8
supports VAXcluster technology, *Sys Mgr Man: Tuning*, 21–8
Feedback
See AUTOGEN feedback
Fiber Distributed Data Interface
See FDDI
FID (file identification)
See File identification
FIELD account
initial modification, *Sys Mgr Man: Essentials*, 6–9
in UAF, *Sys Mgr Man: Essentials*, 6–7
Field of data
in SHOW CLUSTER, *Sys Mgr Man: Tuning*, 20–4
removing, *Sys Mgr Man: Tuning*, 20–11
Field Service accounts
in UAF, *Sys Mgr Man: Essentials*, 6–7
File access
disk, *Sys Mgr Man: Essentials*, 9–13
levels allowed in restore operations, *Sys Mgr Man: Essentials*, 10–29
listing number of concurrent, *Sys Mgr Man: Tuning*, 16–7
of a remote host, *Sys Mgr Man: Tuning*, 21–14
tape, *Sys Mgr Man: Essentials*, 9–13, 9–14
File banner pages, *Sys Mgr Man: Essentials*, 13–42, 13–60
See also Job banner pages
File extensions
effect on system performance, *Sys Mgr Man: Tuning*, 16–7
specifying size, *Sys Mgr Man: Essentials*, 8–22; *Sys Mgr Man: Tuning*, 16–7
system parameter controlling, *Sys Mgr Man: Tuning*, 16–7
File formats
use with BACKUP, *Sys Mgr Man: Essentials*, 10–9
File fragmentation
of page and swap files, *Sys Mgr Man: Tuning*, 15–27
File headers
index file, *Sys Mgr Man: Tuning*, A–7
contents, *Sys Mgr Man: Tuning*, A–7
extension, *Sys Mgr Man: Tuning*, A–8
primary, *Sys Mgr Man: Tuning*, A–8
File identification
file number, *Sys Mgr Man: Tuning*, A–4
Files–11, *Sys Mgr Man: Tuning*, A–4
file sequence number (SEQ), *Sys Mgr Man: Tuning*, A–4
relative volume number (RVN), *Sys Mgr Man: Tuning*, A–4
File Log
VMSINSTAL.COM option, *Sys Mgr Man: Essentials*, 3–17
File names
OpenVMS extended, *Sys Mgr Man: Essentials*, 9–16
standard, *Sys Mgr Man: Essentials*, 9–16
File protection, *Sys Mgr Man: Essentials*, 9–4
SYSDUMP.DMP file, *Sys Mgr Man: Tuning*, 15–4
Files
See also Files–11 On-Disk Structure
See also Parameter files
accessing
See File access
attributes
accessing, *Sys Mgr Man: Essentials*, 9–17
backing up, *Sys Mgr Man: Essentials*, 10–23
comparing with BACKUP, *Sys Mgr Man: Essentials*, 10–25
copying
from disk to standard-labeled volumes, *Sys Mgr Man: Essentials*, 9–20
from disk volumes, *Sys Mgr Man: Essentials*, 9–19
remotely, *Sys Mgr Man: Tuning*, 21–14
to tape, *Sys Mgr Man: Essentials*, 9–21
to tape volumes, *Sys Mgr Man: Essentials*, 9–21
copying with BACKUP, *Sys Mgr Man: Essentials*, 10–23
creating, *Sys Mgr Man: Essentials*, 8–3
detected bad block (DBBF), *Sys Mgr Man: Essentials*, 8–65
expiration dates on, *Sys Mgr Man: Essentials*, 8–54
for AUTOGEN feedback, *Sys Mgr Man: Tuning*, 14–11
limiting number of versions, *Sys Mgr Man: Essentials*, 8–53
logging activity during installation, *Sys Mgr Man: Essentials*, 3–17
lost
recovering, *Sys Mgr Man: Essentials*, 8–57
modifying characteristics, *Sys Mgr Man: Essentials*, 9–8

Files (cont'd)
naming
on Files–11 volume, *Sys Mgr Man: Tuning*, A–7
nonstandard format
DCL commands with, *Sys Mgr Man: Essentials*, 9–2, 9–13
on public volumes, *Sys Mgr Man: Essentials*, 8–8
open during backup, *Sys Mgr Man: Essentials*, 10–30, 10–63
overwriting, *Sys Mgr Man: Essentials*, 9–14
PCF, *Sys Mgr Man: Essentials*, 3–22
private volumes, *Sys Mgr Man: Essentials*, 8–9
privileges, *Sys Mgr Man: Essentials*, 9–4
public, *Sys Mgr Man: Essentials*, 8–8
purging to save disk space, *Sys Mgr Man: Essentials*, 8–53
recovering lost, *Sys Mgr Man: Essentials*, 8–57
remote access, *Sys Mgr Man: Tuning*, 21–14
reserved, *Sys Mgr Man: Tuning*, A–4
list of, *Sys Mgr Man: Essentials*, 8–4
restoring from image backup, *Sys Mgr Man: Essentials*, 10–42
restoring from incremental backup, *Sys Mgr Man: Essentials*, 10–44
restoring with BACKUP, *Sys Mgr Man: Essentials*, 10–28
retrieving information from, *Sys Mgr Man: Essentials*, 9–2
security
using protection codes, *Sys Mgr Man: Essentials*, 9–6
system
moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
tape
See Tape files
tape volumes, *Sys Mgr Man: Essentials*, 9–5
writing to files on, *Sys Mgr Man: Essentials*, 9–18
transferring across network, *Sys Mgr Man: Essentials*, 9–24
using COPY, *Sys Mgr Man: Tuning*, 21–14
versions
limiting number of, *Sys Mgr Man: Essentials*, 8–53, 9–12
VMSMAIL_PROFILE.DATA, *Sys Mgr Man: Essentials*, 6–38
Files–11 On-Disk Structure
block
definition, *Sys Mgr Man: Tuning*, A–1
CD on OpenVMS, *Sys Mgr Man: Essentials*, 8–5
comparison of ODS Level 1 and Level 2, *Sys Mgr Man: Tuning*, A–10
creating structure, *Sys Mgr Man: Essentials*, 8–10
Files–11 On-Disk Structure (cont'd)
disk save set, *Sys Mgr Man: Essentials*, 10–7
file identification, *Sys Mgr Man: Tuning*, A–4
master file directory (MFD), *Sys Mgr Man: Tuning*, A–4
ODS Level 1, *Sys Mgr Man: Essentials*, 8–3
directory hierarchy, *Sys Mgr Man: Tuning*, A–4
ODS Level 2, *Sys Mgr Man: Essentials*, 8–3
assigning disk quotas, *Sys Mgr Man: Essentials*, 8–50
definition, *Sys Mgr Man: Essentials*, 8–4
directory hierarchy, *Sys Mgr Man: Tuning*, A–4
sector, *Sys Mgr Man: Tuning*, A–2
structure, *Sys Mgr Man: Essentials*, 8–3; *Sys Mgr Man: Tuning*, A–3
Level 1, *Sys Mgr Man: Essentials*, 9–19; *Sys Mgr Man: Tuning*, A–4
Level 2, *Sys Mgr Man: Essentials*, 9–19; *Sys Mgr Man: Tuning*, A–4
reserved files, *Sys Mgr Man: Essentials*, 8–4
terminology, *Sys Mgr Man: Tuning*, A–2
UIC, *Sys Mgr Man: Tuning*, A–4
using with Exchange utility (EXCHANGE)
to transfer data, *Sys Mgr Man: Essentials*, 9–24
File specifications
ANSI, *Sys Mgr Man: Essentials*, 9–16
for installing images, *Sys Mgr Man: Tuning*, 16–16
File structures, *Sys Mgr Man: Essentials*, 8–3
File Transfer Protocol
See FTP
File windows
mapping pointers for, *Sys Mgr Man: Essentials*, 8–24
FILLM process limit, *Sys Mgr Man: Essentials*, 6–43
value for efficient backups, *Sys Mgr Man: Essentials*, 10–10
Firmware
LAN updates, *Sys Mgr Man: Tuning*, 22–14
Flag pages, *Sys Mgr Man: Essentials*, 13–33
file, *Sys Mgr Man: Essentials*, 13–35
job, *Sys Mgr Man: Essentials*, 13–33
Foreign volumes
See also Volumes, foreign
Formats
CD–ROM on disc, *Sys Mgr Man: Essentials*, 8–4
converting software kits, *Sys Mgr Man: Essentials*, 3–33
High Sierra, *Sys Mgr Man: Essentials*, 8–4

Formatting volumes, *Sys Mgr Man: Essentials*, 8–11
Form feed
  suppressing, *Sys Mgr Man: Essentials*, 13–44
Form feeds
  controlling page overflow with, *Sys Mgr Man: Essentials*, 13–44
  inserting automatically in output jobs, *Sys Mgr Man: Essentials*, 13–19
Forms
  assigning default for a queue, *Sys Mgr Man: Essentials*, 13–63
  associating with jobs and queues, *Sys Mgr Man: Essentials*, 13–42
  changing DEFAULT, *Sys Mgr Man: Essentials*, 13–63
  commands used with, *Sys Mgr Man: Essentials*, 13–61
  controlling line overflow with, *Sys Mgr Man: Essentials*, 13–42, 13–43
  controlling page width, length and margin size with, *Sys Mgr Man: Essentials*, 13–42
  controlling paper stock with, *Sys Mgr Man: Essentials*, 13–42
  default, *Sys Mgr Man: Essentials*, 13–63
  DEFAULT, *Sys Mgr Man: Essentials*, 13–63
  defining, *Sys Mgr Man: Essentials*, 13–61
  deleting, *Sys Mgr Man: Essentials*, 13–64
    problems, *Sys Mgr Man: Essentials*, 13–81
  description, *Sys Mgr Man: Essentials*, 13–42
  displaying
    defined forms, *Sys Mgr Man: Essentials*, 13–62
    forms assigned to queues, *Sys Mgr Man: Essentials*, 13–64
  formatting jobs with, *Sys Mgr Man: Essentials*, 13–42
  mounting, *Sys Mgr Man: Essentials*, 13–64
  preprinted
    aligning, *Sys Mgr Man: Essentials*, 13–75, 13–76
  procedure for using, *Sys Mgr Man: Essentials*, 13–43, 13–60
  specifying setup modules with, *Sys Mgr Man: Essentials*, 13–42
  specifying sheet-feed paper with, *Sys Mgr Man: Essentials*, 13–42
Fragmentation of disks, *Sys Mgr Man: Essentials*, 10–47
FTP (File Transfer Protocol), *Sys Mgr Man: Tuning*, 21–14
Full backup
  See Image backup
Full names
  DECnet/OSI
    assigning, *Sys Mgr Man: Tuning*, 21–3
    syntax, *Sys Mgr Man: Tuning*, 21–2
Full names (cont'd)
  definition, *Sys Mgr Man: Tuning*, 21–2

G

GBLPAGES system parameter, *Sys Mgr Man: Tuning*, 16–12
GBLSECTIONS system parameter, *Sys Mgr Man: Tuning*, 16–12
Generic queues
  batch, *Sys Mgr Man: Essentials*, 13–3
    recommended use, *Sys Mgr Man: Essentials*, 13–7
  creating, *Sys Mgr Man: Essentials*, 13–17
  description, *Sys Mgr Man: Essentials*, 13–2, 13–3
  in a VMScluster environment, *Sys Mgr Man: Essentials*, 13–2
  output, *Sys Mgr Man: Essentials*, 13–3, 13–11, 13–12
    recommended use, *Sys Mgr Man: Essentials*, 13–11
  relationship to execution queues, *Sys Mgr Man: Essentials*, 13–2
  specifying target execution queues, *Sys Mgr Man: Essentials*, 13–17
Get Save Set
  VMSINSTAL.COM option, *Sys Mgr Man: Essentials*, 3–15, 3–16
Getting help, *Sys Mgr Man: Essentials*, 3–30
GHRs (granularity hint regions)
  slicing shareable images, *Sys Mgr Man: Tuning*, 16–12
Global pages, *Sys Mgr Man: Tuning*, 16–12
Global sections, *Sys Mgr Man: Tuning*, 16–12
Granularity hint regions
  See GHRs
Group numbers
  modifying, *Sys Mgr Man: Tuning*, 20–17
Groups
  accounting, *Sys Mgr Man: Tuning*, 19–5
Group users (security category), *Sys Mgr Man: Essentials*, 11–8
Group volumes
  definition, *Sys Mgr Man: Essentials*, 8–8
GRPPRV privilege
  giving rights of system user, *Sys Mgr Man: Essentials*, 11–8

H

Halting system
  waiting until after system dump file is written, *Sys Mgr Man: Tuning*, 15–2
Hardware
  booting problem, *Sys Mgr Man: Essentials*, 4–16

Hardware (cont'd)
importance of sufficient capacity for system performance, *Sys Mgr Man: Tuning*, 16–6
Header labels
on tape files, *Sys Mgr Man: Essentials*, 8–7, 9–15
reading attributes of, *Sys Mgr Man: Essentials*, 9–17
Header resident images, *Sys Mgr Man: Tuning*, 16–10, 16–12
Help
POLYCENTER Software Installation utility, *Sys Mgr Man: Essentials*, 3–30, 3–35
Help Message utility (MSGHLP), *Sys Mgr Man: Essentials*, 5–25
accessing $STATUS values for uninstalled messages, *Sys Mgr Man: Essentials*, 5–26
adding .MSGHLP$DATA files to the database, *Sys Mgr Man: Essentials*, 5–28
adding comments to the database, *Sys Mgr Man: Essentials*, 5–30
adding messages to the database, *Sys Mgr Man: Essentials*, 5–32
changing Digital-supplied data, *Sys Mgr Man: Essentials*, 5–31
compressing the database after deletions, *Sys Mgr Man: Essentials*, 5–30
creating databases for different user groups, *Sys Mgr Man: Essentials*, 5–28
customizing the database, *Sys Mgr Man: Essentials*, 5–25
default database, *Sys Mgr Man: Essentials*, 5–28
deleting Digital-supplied messages from the database, *Sys Mgr Man: Essentials*, 5–29
message section files, *Sys Mgr Man: Essentials*, 5–26
order searched, *Sys Mgr Man: Essentials*, 5–26
search path of database files, *Sys Mgr Man: Essentials*, 5–28
/SECTION_FILE qualifier usage, *Sys Mgr Man: Essentials*, 5–26
Hierarchical storage controller devices
See HSC devices
High Sierra format
definition, *Sys Mgr Man: Essentials*, 8–4
on CD–ROM, *Sys Mgr Man: Essentials*, 8–4
Highwater marking
disabling for system performance, *Sys Mgr Man: Tuning*, 16–7
Holding a job, *Sys Mgr Man: Essentials*, 13–70
Holding job status, *Sys Mgr Man: Essentials*, 13–69, 13–78
Home blocks, *Sys Mgr Man: Essentials*, 8–58
in index file, *Sys Mgr Man: Tuning*, A–6
Host-based shadow set
mounting disks, *Sys Mgr Man: Essentials*, 10–41
HSC devices
configuring during system startup, *Sys Mgr Man: Essentials*, 7–7
disabling configuration during system startup, *Sys Mgr Man: Essentials*, 7–9
Hyphen (-), restriction on use of, *Sys Mgr Man: Essentials*, 3–18

I

I/O (input/output)
reducing on system disk, *Sys Mgr Man: Tuning*, 16–8
Identification record
ANALYZE/DISK_STRUCTURE, *Sys Mgr Man: Essentials*, 8–56
Identifier field
file, *Sys Mgr Man: Essentials*, 9–15
volume, *Sys Mgr Man: Essentials*, 8–39
Identifiers
Environmental, *Sys Mgr Man: Essentials*, 11–10
General, *Sys Mgr Man: Essentials*, 11–10
system-defined, *Sys Mgr Man: Essentials*, 11–10
types, *Sys Mgr Man: Essentials*, 11–10
UIC, *Sys Mgr Man: Essentials*, 11–10
Idle queue status, *Sys Mgr Man: Essentials*, 13–51
Image backup
command format for disks, *Sys Mgr Man: Essentials*, 10–32
command format for tapes, *Sys Mgr Man: Essentials*, 10–31
definition, *Sys Mgr Man: Essentials*, 10–3
restoring files from, *Sys Mgr Man: Essentials*, 10–42
to disk, *Sys Mgr Man: Essentials*, 10–32
to tape, *Sys Mgr Man: Essentials*, 10–31, 10–32
Image Registry facility, *Sys Mgr Man: Essentials*, 5–22
Images
See also Known images
concurrent access by multiple users, *Sys Mgr Man: Tuning*, 16–11
definition, *Sys Mgr Man: Tuning*, 16–9
determining frequency of use, *Sys Mgr Man: Tuning*, 16–15, 16–16
executable, *Sys Mgr Man: Tuning*, 16–9, 16–14, 16–15
execute-only, *Sys Mgr Man: Tuning*, 16–15
header resident, *Sys Mgr Man: Tuning*, 16–10, 16–12
installing

Images
installing (cont'd)
See also Known images
effect on RUN command, *Sys Mgr Man: Tuning*, 16–10
in system startup, *Sys Mgr Man: Essentials*, 5–4, 5–12, 5–13; *Sys Mgr Man: Tuning*, 16–10
reasons for, *Sys Mgr Man: Tuning*, 16–9
to improve system performance, *Sys Mgr Man: Tuning*, 16–7
known, *Sys Mgr Man: Tuning*, 16–10
See also Known images
linkable, *Sys Mgr Man: Tuning*, 16–9
permanently open, *Sys Mgr Man: Tuning*, 16–10, 16–12
privileged, *Sys Mgr Man: Tuning*, 16–10, 16–13, 16–14
security caution, *Sys Mgr Man: Tuning*, 16–13
privileged shareable, *Sys Mgr Man: Tuning*, 16–14
protected, *Sys Mgr Man: Tuning*, 16–11, 16–14
protecting installed, *Sys Mgr Man: Tuning*, 16–15
relinking to improve system performance, *Sys Mgr Man: Tuning*, 16–7
resident (Alpha), *Sys Mgr Man: Tuning*, 16–11
running in protected modes, *Sys Mgr Man: Tuning*, 16–11, 16–14
shareable, *Sys Mgr Man: Tuning*, 16–11, 16–14
assigning logical names for, *Sys Mgr Man: Tuning*, 16–17
slicing on Alpha systems, *Sys Mgr Man: Tuning*, 16–12
system version dependent
registering, *Sys Mgr Man: Essentials*, 5–22
user-level
calling of protected code, *Sys Mgr Man: Tuning*, 16–11, 16–14
version checking, *Sys Mgr Man: Essentials*, 5–23
writable, *Sys Mgr Man: Tuning*, 16–11
Incremental backup
command format for disks, *Sys Mgr Man: Essentials*, 10–35
command format for tapes, *Sys Mgr Man: Essentials*, 10–34
definition, *Sys Mgr Man: Essentials*, 10–3
restoring files from, *Sys Mgr Man: Essentials*, 10–44
to disk, *Sys Mgr Man: Essentials*, 10–34
to tape, *Sys Mgr Man: Essentials*, 10–33
INDEXF.SYS file
See Index files
Index files, *Sys Mgr Man: Tuning*, A–4 to A–8
alternate file header, *Sys Mgr Man: Tuning*, A–6
backup home block, *Sys Mgr Man: Tuning*, A–5
backup index file header, *Sys Mgr Man: Tuning*, A–6
bitmap, *Sys Mgr Man: Tuning*, A–6
boot block, *Sys Mgr Man: Tuning*, A–5, A–6
bootstrap image, *Sys Mgr Man: Tuning*, A–6
definition, *Sys Mgr Man: Tuning*, A–5
file headers, *Sys Mgr Man: Tuning*, A–6, A–8
file number, *Sys Mgr Man: Tuning*, A–4
home block, *Sys Mgr Man: Tuning*, A–5, A–6
INDEXF.SYS, *Sys Mgr Man: Tuning*, A–5
in volume sets, *Sys Mgr Man: Essentials*, 8–32
reserved file, *Sys Mgr Man: Tuning*, A–5
InfoServer
See also InfoServer Client for OpenVMS
automatic service, *Sys Mgr Man: Tuning*, 23–4
availability, *Sys Mgr Man: Tuning*, 23–4
backing up system disks, *Sys Mgr Man: Essentials*, 10–60
Client
and DECnet, *Sys Mgr Man: Tuning*, 23–10
commands, *Sys Mgr Man: Tuning*, 23–7
console terminal, *Sys Mgr Man: Tuning*, 23–6
determining default server name, *Sys Mgr Man: Tuning*, 23–6
downline loading with, *Sys Mgr Man: Tuning*, 23–2
fail over, *Sys Mgr Man: Tuning*, 23–4
functions, *Sys Mgr Man: Tuning*, 23–1
Help facility, *Sys Mgr Man: Tuning*, 23–8
load balancing, *Sys Mgr Man: Tuning*, 23–6
local connections, *Sys Mgr Man: Tuning*, 23–6
management session, ending, *Sys Mgr Man: Tuning*, 23–7
mounting devices
in system startup, *Sys Mgr Man: Tuning*, 23–14
multicast address feature, *Sys Mgr Man: Tuning*, 23–5
protocols, *Sys Mgr Man: Tuning*, 23–5
quick access to duplicate services for clients, *Sys Mgr Man: Tuning*, 23–4
relationship to client systems, *Sys Mgr Man: Tuning*, 23–2
remote connections, *Sys Mgr Man: Tuning*, 23–6
removing media, *Sys Mgr Man: Tuning*, 23–4
server name, *Sys Mgr Man: Tuning*, 23–6
service disconnection, *Sys Mgr Man: Tuning*, 23–4
setting up in system startup, *Sys Mgr Man: Tuning*, 23–10
software

InfoServer
software (cont'd)
starting client for OpenVMS, *Sys Mgr Man: Tuning*, 23–10
starting a session, *Sys Mgr Man: Tuning*, 23–6
starting Client, *Sys Mgr Man: Tuning*, 23–10
support for X terminals, *Sys Mgr Man: Tuning*, 23–4
system overview, *Sys Mgr Man: Tuning*, 23–1
virtual device server, *Sys Mgr Man: Tuning*, 23–1
virtual device units, *Sys Mgr Man: Tuning*, 23–13
X terminal client, *Sys Mgr Man: Tuning*, 23–4
InfoServer Client for OpenVMS
components, *Sys Mgr Man: Tuning*, 23–8
functions, *Sys Mgr Man: Tuning*, 23–8
mounting devices
in system startup, *Sys Mgr Man: Essentials*, 5–13
setting up in system startup, *Sys Mgr Man: Essentials*, 5–13
software, *Sys Mgr Man: Tuning*, 23–9
starting automatically, *Sys Mgr Man: Tuning*, 23–10
startup restrictions, *Sys Mgr Man: Tuning*, 23–12
InfoServer password, *Sys Mgr Man: Tuning*, 23–6
Initialization files
creating, *Sys Mgr Man: Tuning*, 20–15
establishing SHOW CLUSTER reports, *Sys Mgr Man: Tuning*, 20–5
SHOW CLUSTER
creating, *Sys Mgr Man: Tuning*, 20–14
SHOW_CLUSTER$INIT, *Sys Mgr Man: Tuning*, 20–14, 20–15
use with SYSMAN, *Sys Mgr Man: Essentials*, 2–19
Initialization of system
in a multiprocessing system, *Sys Mgr Man: Tuning*, 26–2
INITIALIZE command
See also Disk commands
See also INITIALIZE/QUEUE command
See also Initializing, volumes
creating volume identifiers for continuation volumes, *Sys Mgr Man: Essentials*, 8–40
disk volumes, *Sys Mgr Man: Essentials*, 8–11
for formatting page and swap file disks during system startup, *Sys Mgr Man: Essentials*, 5–6
mounting volume sets, *Sys Mgr Man: Essentials*, 8–40
qualifiers, *Sys Mgr Man: Essentials*, 8–12
setting device protection, *Sys Mgr Man: Essentials*, 8–16
tape volumes, *Sys Mgr Man: Essentials*, 9–12, 9–22
INITIALIZE command (cont'd)
tape volumes, *Sys Mgr Man: Essentials*, 9–21, 9–22
to format and write label to volume, *Sys Mgr Man: Essentials*, 8–11
INITIALIZE/QUEUE command, *Sys Mgr Man: Essentials*, 13–16, 13–48
activating autostart queues, *Sys Mgr Man: Essentials*, 13–15, 13–49
assigning a default form, *Sys Mgr Man: Essentials*, 13–63
assigning characteristics, *Sys Mgr Man: Essentials*, 13–59
canceling characteristics, *Sys Mgr Man: Essentials*, 13–59
controlling page overflow, *Sys Mgr Man: Essentials*, 13–44
creating a generic queue, *Sys Mgr Man: Essentials*, 13–17
mounting a form, *Sys Mgr Man: Essentials*, 13–64
setting block limits, *Sys Mgr Man: Essentials*, 13–32
setting scheduling policy, *Sys Mgr Man: Essentials*, 13–32
setting UIC-based protection on queues, *Sys Mgr Man: Essentials*, 13–23
specifying autostart information, *Sys Mgr Man: Essentials*, 13–15
specifying banner pages, *Sys Mgr Man: Essentials*, 13–60
specifying job processing options, *Sys Mgr Man: Essentials*, 13–31
specifying queue options, *Sys Mgr Man: Essentials*, 13–19
specifying reset modules, *Sys Mgr Man: Essentials*, 13–65
starting a nonautostart queue, *Sys Mgr Man: Essentials*, 13–16
Initializing
queues, *Sys Mgr Man: Essentials*, 13–15
See also INITIALIZE/QUEUE command
volumes
See also Disk commands
See also INITIALIZE command
assisting users, *Sys Mgr Man: Essentials*, 8–14
disk volumes, *Sys Mgr Man: Essentials*, 8–10, 8–13
results of, *Sys Mgr Man: Essentials*, 10–13
tape volumes, *Sys Mgr Man: Essentials*, 9–5
INITIAL phase of system startup, *Sys Mgr Man: Essentials*, 5–4, 5–18
Initial System Load
See ISL

Input an existing PCF, *Sys Mgr Man: Essentials*, 3–30
Input specifier
to the BACKUP command, *Sys Mgr Man: Essentials*, 10–5
Input symbiont for card reader
running interactively, *Sys Mgr Man: Essentials*, 7–19
Installation procedures
See also Installing software
See also POLYCENTER Software Installation utility
See also VMSINSTAL.COM command procedure
completing, *Sys Mgr Man: Essentials*, 3–13
definition, *Sys Mgr Man: Essentials*, 3–4
on Alpha systems, *Sys Mgr Man: Essentials*, 3–2
to run VAX system as a C2 system, *Sys Mgr Man: Essentials*, 3–5
INSTALL command
in SYSGEN, *Sys Mgr Man: Tuning*, 15–19
in system startup, *Sys Mgr Man: Essentials*, 5–7
Installed files
See also Known images
See Known images
Installing images, *Sys Mgr Man: Tuning*, 16–16
See also Install utility
See also Known images
effect on RUN command, *Sys Mgr Man: Tuning*, 16–10
in SYSTARTUP_VMS.COM, *Sys Mgr Man: Essentials*, 5–12, 5–13; *Sys Mgr Man: Tuning*, 16–10
reasons for, *Sys Mgr Man: Tuning*, 16–9
to improve system performance, *Sys Mgr Man: Tuning*, 16–7, 16–10
Installing page and swap files
in system startup, *Sys Mgr Man: Essentials*, 5–6; *Sys Mgr Man: Tuning*, 15–5, 15–20, 15–21
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–18, 15–22
with SYPAGSWPFILES.COM command procedure, *Sys Mgr Man: Tuning*, 15–20
with SYSGEN, *Sys Mgr Man: Tuning*, 15–19
Installing software
See also Installation procedure
See also POLYCENTER Software Installation utility
See also Software products
See also VMSINSTAL.COM command procedure
as a batch job, *Sys Mgr Man: Essentials*, 3–32
completing procedure, *Sys Mgr Man: Essentials*, 3–13
Installing software (cont'd)
creating a new PCF during, *Sys Mgr Man: Essentials*, 3–30
deinstalling software, *Sys Mgr Man: Essentials*, 3–34
identifying kit location, *Sys Mgr Man: Essentials*, 3–28
logging file activity, *Sys Mgr Man: Essentials*, 3–17
managing installed software, *Sys Mgr Man: Essentials*, 3–34
multiple products at once, *Sys Mgr Man: Essentials*, 3–29
noncompliant products, *Sys Mgr Man: Essentials*, 3–27
on alternate disk, *Sys Mgr Man: Essentials*, 3–12
preparing for installation, *Sys Mgr Man: Essentials*, 3–6
reconfiguring product options, *Sys Mgr Man: Essentials*, 3–32
removing installed software, *Sys Mgr Man: Essentials*, 3–34
responding to questions during installation, *Sys Mgr Man: Essentials*, 3–30
shutting down DECnet, *Sys Mgr Man: Essentials*, 3–7
specifying source and destination locations, *Sys Mgr Man: Essentials*, 3–28
supplying answers from a PCF, *Sys Mgr Man: Essentials*, 3–30
tracking dependencies, *Sys Mgr Man: Essentials*, 3–27
using a PCF for consistency, *Sys Mgr Man: Essentials*, 3–22
Install utility (INSTALL)
See also Installing images
See also Known images
and version checking, *Sys Mgr Man: Essentials*, 5–23
determining number frequency of image use, *Sys Mgr Man: Tuning*, 16–15, 16–16
improving system performance, *Sys Mgr Man: Tuning*, 16–7, 16–9, 16–10
listing number of concurrent file accesses, *Sys Mgr Man: Tuning*, 16–7
making images header resident, *Sys Mgr Man: Tuning*, 16–10, 16–12
making images privileged, *Sys Mgr Man: Tuning*, 16–10, 16–14
permanently open images, *Sys Mgr Man: Tuning*, 16–10, 16–12
reasons for using, *Sys Mgr Man: Tuning*, 16–9
/RESIDENT qualifier on Alpha, *Sys Mgr Man: Tuning*, 16–13
slicing feature on Alpha, *Sys Mgr Man: Tuning*, 16–13

Interactive identifiers, *Sys Mgr Man: Essentials*, 11–11
Interactive users
limiting for performance management, *Sys Mgr Man: Tuning*, 16–4
limiting in system startup, *Sys Mgr Man: Essentials*, 5–16
Interchange environment
protection, *Sys Mgr Man: Essentials*, 8–20
Interfaces to POLYCENTER Software Installation utility
DCL, *Sys Mgr Man: Essentials*, 3–18
DECwindows Motif, *Sys Mgr Man: Essentials*, 3–18
Interfaces to the Backup utility, *Sys Mgr Man: Essentials*, 10–4
Internet, *Sys Mgr Man: Tuning*, 21–13
Intrusion databases, *Sys Mgr Man: Essentials*, 11–7
Intrusions
detection and evasion, *Sys Mgr Man: Essentials*, 11–6
Intrusion services
managed by security server, *Sys Mgr Man: Essentials*, 5–5
Invoking AUTOGEN, *Sys Mgr Man: Tuning*, 14–9
Invoking SDA automatically, *Sys Mgr Man: Tuning*, 15–12
IO AUTOCONFIGURE command
in SYSMAN, *Sys Mgr Man: Essentials*, 7–6, 7–8
in system startup, *Sys Mgr Man: Essentials*, 5–4, 5–7, 7–6
IO CONNECT command
in SYSMAN, *Sys Mgr Man: Essentials*, 7–8
in system startup, *Sys Mgr Man: Essentials*, 5–7
IO LOAD command
in SYSMAN, *Sys Mgr Man: Essentials*, 7–8
IPC (Interrupt Priority C)
using to adjust quorum, *Sys Mgr Man: Essentials*, 8–63
using to cancel mount verification, *Sys Mgr Man: Essentials*, 8–63
using to enter the debugger, *Sys Mgr Man: Essentials*, 8–63
IRG (interrecord gap), *Sys Mgr Man: Essentials*, 8–6
ISL (Initial System Load), *Sys Mgr Man: Tuning*, 23–2
ISO 9660 standard
data interleaving, *Sys Mgr Man: Essentials*, 8–6
establishing default file attributes, *Sys Mgr Man: Essentials*, 8–24
format
definition, *Sys Mgr Man: Essentials*, 8–4
groups
ISO 9660 standard
groups (cont'd)
mounting, *Sys Mgr Man: Essentials*, 8–34
media
showing device information, *Sys Mgr Man: Essentials*, 7–5
media protection, *Sys Mgr Man: Essentials*, 8–17
mounting a volume for, *Sys Mgr Man: Essentials*, 8–23
partially mounted volume sets, *Sys Mgr Man: Essentials*, 8–35
partially recorded data blocks, *Sys Mgr Man: Essentials*, 8–5
restrictions, *Sys Mgr Man: Essentials*, 8–36
standard on OpenVMS, *Sys Mgr Man: Essentials*, 8–5
UNDEFINED record format errors, *Sys Mgr Man: Essentials*, 8–36
volume label and volume set label duplication, *Sys Mgr Man: Essentials*, 8–36
volume labels, *Sys Mgr Man: Essentials*, 8–36
volume set labels, *Sys Mgr Man: Essentials*, 8–36
volume sets
mounting, *Sys Mgr Man: Essentials*, 8–34
partially mounted, *Sys Mgr Man: Essentials*, 8–35
volume structure, *Sys Mgr Man: Essentials*, 8–3

J

Job banner pages, *Sys Mgr Man: Essentials*, 13–42, 13–60
See also File banner pages
$JOB card, *Sys Mgr Man: Essentials*, 7–18
Job controllers
See also JOBCTL process
See also Queue manager
and batch jobs, *Sys Mgr Man: Essentials*, 12–3, 13–2
communication with queue manager, *Sys Mgr Man: Essentials*, 12–3
relationship to queue manager, *Sys Mgr Man: Essentials*, 12–3, 12–7
starting queue manager, *Sys Mgr Man: Essentials*, 12–6, 12–7
tasks performed by, *Sys Mgr Man: Essentials*, 12–3
JOBCTL process
See also Job controllers
creation during system startup, *Sys Mgr Man: Essentials*, 5–5
Job retention
changing for a job, *Sys Mgr Man: Essentials*, 13–73

Job retention (cont'd)
specifying for a job, *Sys Mgr Man: Essentials*, 13–26
specifying for a queue, *Sys Mgr Man: Essentials*, 13–26
Jobs
See also Batch jobs
See also Output jobs
changing scheduling priority, *Sys Mgr Man: Essentials*, 13–72
controlling print position and alignment, *Sys Mgr Man: Essentials*, 13–75, 13–76
deleting, *Sys Mgr Man: Essentials*, 13–74
holding, *Sys Mgr Man: Essentials*, 13–70, 13–78
merging, *Sys Mgr Man: Essentials*, 13–56
modifying, *Sys Mgr Man: Essentials*, 13–70
moving from one queue to another, *Sys Mgr Man: Essentials*, 13–57
releasing, *Sys Mgr Man: Essentials*, 13–70
requeuing
executing, *Sys Mgr Man: Essentials*, 13–72
pending, *Sys Mgr Man: Essentials*, 13–73
retaining in a queue, *Sys Mgr Man: Essentials*, 13–73
suspending, *Sys Mgr Man: Essentials*, 13–75
Job scheduling, *Sys Mgr Man: Essentials*, 13–32, 13–72
for output jobs, *Sys Mgr Man: Essentials*, 13–32
priority
See Priority, job scheduling
Job status
definitions, *Sys Mgr Man: Essentials*, 13–69
error, *Sys Mgr Man: Essentials*, 13–26, 13–31, 13–54, 13–73
holding, *Sys Mgr Man: Essentials*, 13–71, 13–78
pending, *Sys Mgr Man: Essentials*, 13–72, 13–74, 13–78
retained, *Sys Mgr Man: Essentials*, 13–71, 13–73
showing, *Sys Mgr Man: Essentials*, 13–26, 13–68
use in job retention, *Sys Mgr Man: Essentials*, 13–26
Job table quota, *Sys Mgr Man: Essentials*, 6–43
Journal file of backup information, *Sys Mgr Man: Essentials*, 10–26
listing contents of, *Sys Mgr Man: Essentials*, 10–27
Journal file of queue database, *Sys Mgr Man: Essentials*, 12–4
See also Queue database
location, *Sys Mgr Man: Essentials*, 12–6, 12–7
changing, *Sys Mgr Man: Essentials*, 12–6
JTQUOTA process quota, *Sys Mgr Man: Essentials*, 6–43

K

Kernel mode
calling images running in, *Sys Mgr Man: Tuning*, 16–11, 16–14
logical names, *Sys Mgr Man: Tuning*, 16–15
Keypad definitions, *Sys Mgr Man: Tuning*, 20–5, 20–7
Kits
See Software products
Known file lists
definition, *Sys Mgr Man: Tuning*, 16–10
in system startup, *Sys Mgr Man: Essentials*, 5–12
Known images
definition, *Sys Mgr Man: Tuning*, 16–10
deleting, *Sys Mgr Man: Tuning*, 16–18
dismounting volume, *Sys Mgr Man: Tuning*, 16–18
displaying, *Sys Mgr Man: Tuning*, 16–16
evaluating merits of installing, *Sys Mgr Man: Tuning*, 16–15, 16–16
file specification for, *Sys Mgr Man: Tuning*, 16–16
installing, *Sys Mgr Man: Tuning*, 16–16
in system startup, *Sys Mgr Man: Essentials*, 5–12; *Sys Mgr Man: Tuning*, 16–10
privilege enhancement, *Sys Mgr Man: Tuning*, 16–13
removing, *Sys Mgr Man: Tuning*, 16–18
resident, *Sys Mgr Man: Tuning*, 16–12

L

Labels
backup tape, *Sys Mgr Man: Essentials*, 10–21
changing volume, *Sys Mgr Man: Essentials*, 3–33
header, *Sys Mgr Man: Essentials*, 8–24
initializing volume to write, *Sys Mgr Man: Essentials*, 8–11
trailer, *Sys Mgr Man: Essentials*, 8–7
LAD Control Program (LADCP) utility, *Sys Mgr Man: Tuning*, 23–13 to 23–14
BIND command, *Sys Mgr Man: Tuning*, 23–14
exiting, *Sys Mgr Man: Tuning*, 23–13
Help facility, *Sys Mgr Man: Tuning*, 23–14
invoking, *Sys Mgr Man: Tuning*, 23–13
making remote InfoServer devices available locally, *Sys Mgr Man: Tuning*, 23–14
summary, *Sys Mgr Man: Tuning*, 23–13
LADCP (LAD Control Program)
See LAD Control Program utility

LANACP utility
See LAN Auxiliary Control Program utility
LAN Auxiliary Control Program (LANACP) utility
OPCOM messages displayed, *Sys Mgr Man: Tuning*, 22–22
running, *Sys Mgr Man: Tuning*, 22–6
stopping, *Sys Mgr Man: Tuning*, 22–6
LAN Control Program (LANACP) utility
servers, *Sys Mgr Man: Tuning*, 22–5
LAN Control Program (LANCP) utility, *Sys Mgr Man: Tuning*, 22–6
clearing counters, *Sys Mgr Man: Tuning*, 22–25
deleting device information, *Sys Mgr Man: Tuning*, 22–17
deleting node information, *Sys Mgr Man: Tuning*, 22–20
device database management, *Sys Mgr Man: Tuning*, 22–14
device management, *Sys Mgr Man: Tuning*, 22–9
disabling MOP downline load service, *Sys Mgr Man: Tuning*, 22–23
displaying device information, *Sys Mgr Man: Tuning*, 22–15
displaying node information, *Sys Mgr Man: Tuning*, 22–18
displaying OPCOM messages, *Sys Mgr Man: Tuning*, 22–25
displaying status and counters, *Sys Mgr Man: Tuning*, 22–23, 22–24
enabling MOP downline load service, *Sys Mgr Man: Tuning*, 22–23
load trace facility, *Sys Mgr Man: Tuning*, 22–25
MOP console carrier, *Sys Mgr Man: Tuning*, 22–26
MOP downline load service management, *Sys Mgr Man: Tuning*, 22–23
MOP trigger boot, *Sys Mgr Man: Tuning*, 22–27
node database management, *Sys Mgr Man: Tuning*, 22–17
running, *Sys Mgr Man: Tuning*, 22–7
setting device information, *Sys Mgr Man: Tuning*, 22–15
setting node information, *Sys Mgr Man: Tuning*, 22–18
setting up LAN MOP, *Sys Mgr Man: Tuning*, 22–22
SPAWN function, *Sys Mgr Man: Tuning*, 22–9
using command files, *Sys Mgr Man: Tuning*, 22–9
LANCP (Local Area Network Control Program)
See LAN Control Program utility
LAN drivers
address
multicast, *Sys Mgr Man: Tuning*, 22–3
node, *Sys Mgr Man: Tuning*, 22–3
physical, *Sys Mgr Man: Tuning*, 22–3
addresses, *Sys Mgr Man: Tuning*, 22–3
Ethernet, *Sys Mgr Man: Tuning*, 22–2
FDDI, *Sys Mgr Man: Tuning*, 22–2
port, *Sys Mgr Man: Tuning*, 22–2
protocol type, *Sys Mgr Man: Tuning*, 22–2
Token Ring, *Sys Mgr Man: Tuning*, 22–2
Language formats, *Sys Mgr Man: Essentials*, 5–43
Languages
specifying, *Sys Mgr Man: Essentials*, 5–44, 5–45, 5–50
LANs
using multiple LAN adapters for LAT node, *Sys Mgr Man: Tuning*, 24–8
LANs (local area networks)
clearing counters, *Sys Mgr Man: Tuning*, 22–25
connections, *Sys Mgr Man: Tuning*, 21–8
deleting device information, *Sys Mgr Man: Tuning*, 22–17
deleting node information, *Sys Mgr Man: Tuning*, 22–20
device management, *Sys Mgr Man: Tuning*, 22–9
disabling MOP downline load service, *Sys Mgr Man: Tuning*, 22–23
displaying device information, *Sys Mgr Man: Tuning*, 22–15
displaying LAN device configurations, *Sys Mgr Man: Tuning*, 22–9
displaying LAN device parameters, *Sys Mgr Man: Tuning*, 22–10
displaying node information, *Sys Mgr Man: Tuning*, 22–18
displaying OPCOM messages, *Sys Mgr Man: Tuning*, 22–25
displaying status and counters, *Sys Mgr Man: Tuning*, 22–23, 22–24
enabling MOP downline load service, *Sys Mgr Man: Tuning*, 22–23
LANACP device database management, *Sys Mgr Man: Tuning*, 22–14
LANACP related OPCOM messages, *Sys Mgr Man: Tuning*, 22–22
LAN Auxiliary Control Program (LANACP) utility, *Sys Mgr Man: Tuning*, 22–5
LAN Control Program (LANCP) utility, *Sys Mgr Man: Tuning*, 22–6
LANCP command files, *Sys Mgr Man: Tuning*, 22–9
LANCP SPAWN function, *Sys Mgr Man: Tuning*, 22–9

LANs (local area networks) (cont'd)
LAN firmware updates, *Sys Mgr Man: Tuning*, 22–14
LAN MOP and DECnet MOP, *Sys Mgr Man: Tuning*, 22–20
load trace facility, *Sys Mgr Man: Tuning*, 22–25
migrating DECnet MOP to LAN MOP, *Sys Mgr Man: Tuning*, 22–20
MOP console carrier, *Sys Mgr Man: Tuning*, 22–26
MOP downline load service, *Sys Mgr Man: Tuning*, 22–20
MOP downline load service management, *Sys Mgr Man: Tuning*, 22–23
MOP trigger boot, *Sys Mgr Man: Tuning*, 22–27
node database management, *Sys Mgr Man: Tuning*, 22–17
running the LANACP utility, *Sys Mgr Man: Tuning*, 22–6
sample LAN MOP set up, *Sys Mgr Man: Tuning*, 22–22
setting device information, *Sys Mgr Man: Tuning*, 22–15
setting LAN device parameters, *Sys Mgr Man: Tuning*, 22–12
setting node information, *Sys Mgr Man: Tuning*, 22–18
setting up LAN MOP with CLUSTER_CONFIG, *Sys Mgr Man: Tuning*, 22–21
stopping the LANACP utility, *Sys Mgr Man: Tuning*, 22–6
system management enhancements, *Sys Mgr Man: Tuning*, 22–1
LASTCP (LASTport Control Program)
See LASTport Control Program utility
LASTport Control Program (LASTCP), *Sys Mgr Man: Tuning*, 23–5 to 23–13
account requirements, *Sys Mgr Man: Tuning*, 23–12
command summary, *Sys Mgr Man: Tuning*, 23–10
exiting, *Sys Mgr Man: Tuning*, 23–9
functions, *Sys Mgr Man: Tuning*, 23–12
Help facility, *Sys Mgr Man: Tuning*, 23–10
invoking, *Sys Mgr Man: Tuning*, 23–9
MAXBUF system parameter requirement, *Sys Mgr Man: Tuning*, 23–12
privileges required, *Sys Mgr Man: Tuning*, 23–9
LASTport Control Program utility
functions, *Sys Mgr Man: Tuning*, 23–9
LASTport/Disk protocol, *Sys Mgr Man: Tuning*, 23–5
LASTport/Disk service, *Sys Mgr Man: Tuning*, 23–12
ESS$DADDRIVER, *Sys Mgr Man: Tuning*, 23–13
LASTport protocol, *Sys Mgr Man: Tuning*, 23–5
LASTport/Tape protocol, *Sys Mgr Man: Tuning*, 23–5
LASTport/Tape service, *Sys Mgr Man: Tuning*, 23–12
ESS$MADDRIVER, *Sys Mgr Man: Tuning*, 23–13
LASTport transport, *Sys Mgr Man: Tuning*, 23–13
LAT$CONFIG.COM command procedure, *Sys Mgr Man: Tuning*, 24–16
invoking during system startup, *Sys Mgr Man: Essentials*, 5–15
LAT$STARTUP.COM command procedure, *Sys Mgr Man: Tuning*, 24–16, 24–17
invoking during system startup, *Sys Mgr Man: Essentials*, 5–15
LAT$SYSTARTUP.COM command procedure, *Sys Mgr Man: Tuning*, 24–16, 24–17, 24–18, 24–22
example, *Sys Mgr Man: Tuning*, 24–23
invoking during system startup, *Sys Mgr Man: Essentials*, 5–15
LATACP
See LAT Ancillary Control Process
LAT Ancillary Control Process (LATACP), *Sys Mgr Man: Tuning*, 24–24
LAT Control Program (LATCP) utility, *Sys Mgr Man: Tuning*, 24–3, 24–14
See also LAT software
DELETE QUEUE_ENTRY command, *Sys Mgr Man: Tuning*, 24–21
exiting, *Sys Mgr Man: Tuning*, 24–15
features, *Sys Mgr Man: Tuning*, 24–14
invoking, *Sys Mgr Man: Tuning*, 24–15
/[NO]ANNOUNCEMENTS qualifier, *Sys Mgr Man: Tuning*, 24–4
/[NO]LARGE_BUFFER qualifier, *Sys Mgr Man: Tuning*, 24–13
queuing incoming requests, *Sys Mgr Man: Tuning*, 24–21
SET NODE command, *Sys Mgr Man: Tuning*, 24–21
SET SERVICE command, *Sys Mgr Man: Tuning*, 24–21
setting up limited services, *Sys Mgr Man: Tuning*, 24–21
SHOW QUEUE_ENTRY command, *Sys Mgr Man: Tuning*, 24–21
SHOW SERVICE command, *Sys Mgr Man: Tuning*, 24–21
summary of commands, *Sys Mgr Man: Tuning*, 24–15

LATCP (Local Area Transport Control Program)
See LAT Control Program (LATCP) utility
LAT server, *Sys Mgr Man: Tuning*, 23–6
LAT software
See also LAT Control Program utility
advantages and uses, *Sys Mgr Man: Tuning*, 24–2
application programs, *Sys Mgr Man: Tuning*, 24–2
creating a service, *Sys Mgr Man: Tuning*, 24–19
customizing, *Sys Mgr Man: Tuning*, 24–18
disabling service announcements, *Sys Mgr Man: Tuning*, 24–4
enabling outgoing connections, *Sys Mgr Man: Tuning*, 24–22
load balancing, *Sys Mgr Man: Tuning*, 24–3
managing the database size, *Sys Mgr Man: Tuning*, 24–24
modems, *Sys Mgr Man: Tuning*, 24–2
outgoing connections, *Sys Mgr Man: Tuning*, 24–2, 24–3, 24–14
printers, *Sys Mgr Man: Tuning*, 24–2
autostart queues on, *Sys Mgr Man: Essentials*, 13–4, 13–10
increasing availability of, *Sys Mgr Man: Essentials*, 13–4, 13–10
LATSYM symbiont, *Sys Mgr Man: Essentials*, 13–3, 13–78
PRTSMB symbiont, *Sys Mgr Man: Essentials*, 13–78
sample configuration, *Sys Mgr Man: Essentials*, 13–10
setting up, *Sys Mgr Man: Essentials*, 13–14
spooling, *Sys Mgr Man: Essentials*, 7–14
troubleshooting, *Sys Mgr Man: Essentials*, 13–78
queuing incoming requests, *Sys Mgr Man: Tuning*, 24–21
service
announcements, *Sys Mgr Man: Tuning*, 24–4, 24–14
database, *Sys Mgr Man: Tuning*, 24–14
dedicated applications, *Sys Mgr Man: Tuning*, 24–2
defined, *Sys Mgr Man: Tuning*, 24–2
node, *Sys Mgr Man: Tuning*, 24–3, 24–4, 24–14
remote printing, *Sys Mgr Man: Tuning*, 24–2
setting up limited services, *Sys Mgr Man: Tuning*, 24–21
setting up ports, *Sys Mgr Man: Tuning*, 24–20
starting network in command procedure, *Sys Mgr Man: Essentials*, 5–15; *Sys Mgr Man: Tuning*, 24–17
LAT software (cont'd)
starting with LAT$STARTUP.COM, *Sys Mgr Man: Essentials*, 5–15; *Sys Mgr Man: Tuning*, 24–16, 24–17
terminals, *Sys Mgr Man: Essentials*, 7–13; *Sys Mgr Man: Tuning*, 24–2
determining characteristics of, *Sys Mgr Man: Essentials*, 7–13
disconnecting, *Sys Mgr Man: Essentials*, 7–12
using large buffers, *Sys Mgr Man: Tuning*, 24–13
using multiple LAN adapters, *Sys Mgr Man: Tuning*, 24–8
LATSYM symbiont, *Sys Mgr Man: Essentials*, 13–3, 13–78
Layered products
startup database, *Sys Mgr Man: Essentials*, 5–18, 5–19
startup phases, *Sys Mgr Man: Essentials*, 5–18
Layered software installation, *Sys Mgr Man: Essentials*, 3–18
Level 1 routers
See Routers, Level 1
Level 2 routers
See Routers, Level 2
Lexical functions
F$GETJPI, *Sys Mgr Man: Tuning*, 26–9
F$GETQUI, *Sys Mgr Man: Essentials*, 13–51
F$GETSYI, *Sys Mgr Man: Tuning*, 26–9
getting information about queues, *Sys Mgr Man: Essentials*, 13–51
getting information about vector processing, *Sys Mgr Man: Tuning*, 26–9
LIBDECOMP.COM command procedure, *Sys Mgr Man: Tuning*, 16–6
Libraries
See also Device control libraries
See Device control libraries
License database, *Sys Mgr Man: Essentials*, 3–8
logical name defining location, *Sys Mgr Man: Essentials*, 5–8
License Management Facility (LMF), *Sys Mgr Man: Essentials*, 3–8, 5–5
LICENSE MODIFY command, *Sys Mgr Man: Essentials*, 3–8
Licenses
loading, *Sys Mgr Man: Essentials*, 3–8
in system startup, *Sys Mgr Man: Essentials*, 5–5
Limits
See Process limits
See UAFs (user authorization files), resource limits
Line printer
UETP test image, *Sys Mgr Man: Tuning*, 17–34

Line printers
preparing for UETP, *Sys Mgr Man: Tuning*, 17–3, 17–5, 17–8
testing with UETP, *Sys Mgr Man: Tuning*, 17–31
UETP output, *Sys Mgr Man: Tuning*, 17–33
UETP test image, *Sys Mgr Man: Tuning*, 17–34
Lines
communications
definition, *Sys Mgr Man: Tuning*, 21–4
network, *Sys Mgr Man: Tuning*, 21–5
definition, *Sys Mgr Man: Tuning*, 21–4
overflow
controlling, *Sys Mgr Man: Essentials*, 13–43
Linkable images, *Sys Mgr Man: Tuning*, 16–9
LINK command
See also Linker utility
/SELECTIVE_SEARCH qualifier, *Sys Mgr Man: Essentials*, 5–23
Linker utility (LINK)
linking against SYS.STB, *Sys Mgr Man: Essentials*, 5–23
Links
logical
See Logical links
List operations
with BACKUP, *Sys Mgr Man: Essentials*, 10–19
LM$JOURNAL file type, *Sys Mgr Man: Tuning*, 25–3
LMF$LICENSE logical name
defining during system startup, *Sys Mgr Man: Essentials*, 5–8
Load balancing
using LAT software, *Sys Mgr Man: Tuning*, 24–3
LOAD command
in SYSGEN (VAX), *Sys Mgr Man: Essentials*, 7–7
Loading device drivers
automatically, *Sys Mgr Man: Essentials*, 7–6, 7–8
manually, *Sys Mgr Man: Essentials*, 7–8
on VAX, *Sys Mgr Man: Essentials*, 7–7
Load leveling
dynamic, *Sys Mgr Man: Tuning*, 26–2
LOADS logical name, *Sys Mgr Man: Tuning*, 17–34
Load tests
defining user load for UETP, *Sys Mgr Man: Tuning*, 17–14
description, *Sys Mgr Man: Tuning*, 17–34
error during UETP, *Sys Mgr Man: Tuning*, 17–27
running individually, *Sys Mgr Man: Tuning*, 17–13
Load trace facility, *Sys Mgr Man: Tuning*, 22–25
Local Area Network Auxiliary Control Program
See LAN Auxiliary Control Program utility
Local area networks
See LANs
Local identifiers, *Sys Mgr Man: Essentials*, 11–11
Local nodes
definition, *Sys Mgr Man: Essentials*, 2–13
Local page and swap files
installing with SATELLITE_PAGE.COM
procedure, *Sys Mgr Man: Essentials*, 5–6; *Sys Mgr Man: Tuning*, 15–20
Location
of page file, *Sys Mgr Man: Tuning*, 15–3, 15–5
specifying alternate, *Sys Mgr Man: Tuning*, 15–24
of queue database, *Sys Mgr Man: Essentials*, 12–5, 12–7
master file, *Sys Mgr Man: Essentials*, 12–6
queue and journal files, *Sys Mgr Man: Essentials*, 12–6
changing, *Sys Mgr Man: Essentials*, 12–6
of swap file, *Sys Mgr Man: Tuning*, 15–5
specifying alternate, *Sys Mgr Man: Tuning*, 15–24
of system dump file, *Sys Mgr Man: Tuning*, 15–2
of system files
redefining with logical names, *Sys Mgr Man: Tuning*, 16–8
Location of software products, *Sys Mgr Man: Essentials*, 3–28
Log file generated by UETP
OLDUETP.LOG, *Sys Mgr Man: Tuning*, 17–21
Log files
generated by UETP
OLDUETP.LOG, *Sys Mgr Man: Tuning*, 17–20
moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
operator
creating new, *Sys Mgr Man: Tuning*, 18–22
enabling and disabling classes, *Sys Mgr Man: Tuning*, 18–23
maintaining, *Sys Mgr Man: Tuning*, 18–24
printing, *Sys Mgr Man: Tuning*, 18–24
restarting, *Sys Mgr Man: Tuning*, 18–24
security alarm messages, *Sys Mgr Man: Tuning*, 18–21
setting up, *Sys Mgr Man: Tuning*, 18–22
specifying location, *Sys Mgr Man: Tuning*, 18–22
troubleshooting the queue manager, *Sys Mgr Man: Essentials*, 12–14
security audit, *Sys Mgr Man: Tuning*, 18–2

Log files
security audit (cont'd)
creating new version, *Sys Mgr Man: Tuning*, 18–30
reviewing, *Sys Mgr Man: Tuning*, 18–26
Log files generated by UETP
during the load test, *Sys Mgr Man: Tuning*, 17–27
NETSERVER.LOG, *Sys Mgr Man: Tuning*, 17–26
Logging in, *Sys Mgr Man: Essentials*, 2–9
See also Logins
to a remote host, *Sys Mgr Man: Tuning*, 21–13
when errors in login procedures prevent, *Sys Mgr Man: Essentials*, 4–8
when errors in startup procedures prevent, *Sys Mgr Man: Essentials*, 4–8
when forgotten passwords prevent, *Sys Mgr Man: Essentials*, 4–10
Logging out
using command procedure, *Sys Mgr Man: Essentials*, 6–22
while using VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–7
Logging startup
with SYSMAN, *Sys Mgr Man: Essentials*, 4–15
Logical links
network, *Sys Mgr Man: Tuning*, 21–5
definition, *Sys Mgr Man: Tuning*, 21–4
Logical names
See also Symbols
ACCOUNTNG, *Sys Mgr Man: Tuning*, 19–4
AGEN$FEEDBACK_REQ_TIME, *Sys Mgr Man: Tuning*, 14–21
assigning for shareable images, *Sys Mgr Man: Tuning*, 16–17
assigning systemwide in system startup, *Sys Mgr Man: Essentials*, 5–8
assigning to customize the SHUTDOWN.COM procedure, *Sys Mgr Man: Essentials*, 4–32
assigning to devices, *Sys Mgr Man: Essentials*, 5–11
for software product installations, *Sys Mgr Man: Essentials*, 3–28
for source and destination locations, *Sys Mgr Man: Essentials*, 3–28
for system components
recommended privilege mode, *Sys Mgr Man: Essentials*, 5–9
LMF$LICENSE, *Sys Mgr Man: Essentials*, 5–8
NETNODE_REMOTE, *Sys Mgr Man: Essentials*, 5–8
NETPROXY, *Sys Mgr Man: Essentials*, 5–8
OPC$LOGFILE_CLASSES, *Sys Mgr Man: Tuning*, 18–23
OPC$LOGFILE_ENABLE, *Sys Mgr Man: Tuning*, 18–23
Logical names (cont'd)
OPC$LOGFILE_NAME, *Sys Mgr Man: Tuning*, 18–22, 18–24
OPC$OPA0_ENABLE, *Sys Mgr Man: Tuning*, 18–24
overriding source and destination locations, *Sys Mgr Man: Essentials*, 3–28
PCSI$DESTINATION, *Sys Mgr Man: Essentials*, 3–28
PCSI$SOURCE, *Sys Mgr Man: Essentials*, 3–28
privilege modes, *Sys Mgr Man: Essentials*, 5–9; *Sys Mgr Man: Tuning*, 16–15
QMAN$MASTER, *Sys Mgr Man: Essentials*, 12–6
redefining location of system files, *Sys Mgr Man: Tuning*, 16–8
RIGHTSLIST, *Sys Mgr Man: Essentials*, 5–8
SHOW_CLUSTER$INIT, *Sys Mgr Man: Tuning*, 20–14
SHUTDOWN$DECNET_MINUTES, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$DISABLE_AUTOSTART, *Sys Mgr Man: Essentials*, 4–33, 13–56
SHUTDOWN$INFORM_NODES, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$MINIMUM_MINUTES, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$QUEUE_MINUTES, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$TIME, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$VERBOSE, *Sys Mgr Man: Essentials*, 4–33
specifying as mail addresses, *Sys Mgr Man: Essentials*, 5–33
STARTUP$STARTUP_LAYERED, *Sys Mgr Man: Essentials*, 5–18
STARTUP$STARTUP_VMS, *Sys Mgr Man: Essentials*, 5–18
SYS$ANNOUNCE, *Sys Mgr Man: Essentials*, 5–14
SYS$AUDIT_SERVER_INHIBIT, *Sys Mgr Man: Essentials*, 5–9; *Sys Mgr Man: Tuning*, 18–28
SYS$DECDTM_INHIBIT, *Sys Mgr Man: Tuning*, 25–21
SYS$ERRORLOG, *Sys Mgr Man: Essentials*, 5–8
SYS$JOURNAL, *Sys Mgr Man: Tuning*, 25–3
SYS$MONITOR, *Sys Mgr Man: Essentials*, 5–8
SYS$STARTUP, *Sys Mgr Man: Essentials*, 5–2
SYS$SYLOGIN, *Sys Mgr Man: Essentials*, 5–17
SYS$WELCOME, *Sys Mgr Man: Essentials*, 5–15
SYSUAF, *Sys Mgr Man: Essentials*, 5–8

Logical names (cont'd)
translation of, *Sys Mgr Man: Essentials*, 5–33
trusted, *Sys Mgr Man: Tuning*, 16–15
UAFALTERNATE, *Sys Mgr Man: Essentials*, 4–10
using in SYSMAN, *Sys Mgr Man: Essentials*, 2–14
VMSMAIL_PROFILE, *Sys Mgr Man: Essentials*, 5–8
Logical names used by UETP
CTRLNAME, *Sys Mgr Man: Tuning*, 17–32
LOADS, *Sys Mgr Man: Tuning*, 17–34
SYS$INPUT, *Sys Mgr Man: Tuning*, 17–31
SYS$OUTPUT, *Sys Mgr Man: Tuning*, 17–33
Logical name tables
definition, *Sys Mgr Man: Essentials*, 5–8
Logical queues
assigning, *Sys Mgr Man: Essentials*, 13–56
description, *Sys Mgr Man: Essentials*, 13–4
recommended use, *Sys Mgr Man: Essentials*, 13–4, 13–56
Login command procedures
booting without, *Sys Mgr Man: Essentials*, 4–8
defining announcements in, *Sys Mgr Man: Essentials*, 5–14
defining location of during system startup, *Sys Mgr Man: Essentials*, 5–17
definition, *Sys Mgr Man: Essentials*, 5–16, 5–17
for captive account, *Sys Mgr Man: Essentials*, 6–29
sample, *Sys Mgr Man: Essentials*, 6–30
for SYSTEM account, *Sys Mgr Man: Essentials*, 5–16
individual, *Sys Mgr Man: Essentials*, 6–19, 6–20
sample, *Sys Mgr Man: Essentials*, 6–21
in SYSMAN, *Sys Mgr Man: Essentials*, 2–16
LOGIN.COM, *Sys Mgr Man: Essentials*, 5–16
SYLOGIN.COM, *Sys Mgr Man: Essentials*, 5–16
systemwide, *Sys Mgr Man: Essentials*, 5–16, 6–20
sample, *Sys Mgr Man: Essentials*, 6–20
to a remote host, *Sys Mgr Man: Tuning*, 21–13
user-specified, *Sys Mgr Man: Essentials*, 6–20
when errors prevent you from logging in, *Sys Mgr Man: Essentials*, 4–8
Logins
See also Logging in
controlling number of dialup attempts, *Sys Mgr Man: Essentials*, 11–6
restricting by function, *Sys Mgr Man: Essentials*, 6–28
restricting by time, *Sys Mgr Man: Essentials*, 6–27, 6–28
sequence of events, *Sys Mgr Man: Essentials*, 6–5
Log in to a remote host, *Sys Mgr Man: Tuning*, 21–13
LOGOUT command, *Sys Mgr Man: Essentials*, 6–23; *Sys Mgr Man: Tuning*, 26–9
Logout command procedures
SYLOGOUT.COM, *Sys Mgr Man: Essentials*, 6–22
Long report format
See Console report during UETP
Loopback tests
network
circuit-level, *Sys Mgr Man: Tuning*, 21–17
definition, *Sys Mgr Man: Tuning*, 21–17
node-level, *Sys Mgr Man: Tuning*, 21–17
Lost files
recovering, *Sys Mgr Man: Essentials*, 8–56 to 8–58
LPBEGIN phase of system startup, *Sys Mgr Man: Essentials*, 5–19
LPBETA phase of system startup, *Sys Mgr Man: Essentials*, 5–19
LPMAIN phase of system startup, *Sys Mgr Man: Essentials*, 5–19
LTA*n* devices, *Sys Mgr Man: Essentials*, 7–12
LTDRIVER (LAT port driver)
turning on and off, *Sys Mgr Man: Tuning*, 24–14

M

Machine check errors
if returned when booting, *Sys Mgr Man: Essentials*, 4–16
MAD virtual tape unit, *Sys Mgr Man: Tuning*, 23–12
Magnetic tape ancillary control process
See MTACP
MAIL
managing, *Sys Mgr Man: Essentials*, 6–38
Mail utility (MAIL)
logical name for, *Sys Mgr Man: Essentials*, 5–8
MAIL$SYSTEM_FLAGS logical name, *Sys Mgr Man: Essentials*, 5–32
managing accounts, *Sys Mgr Man: Essentials*, 6–38
READ/NEW command, *Sys Mgr Man: Essentials*, 5–33
sending AUTOGEN report with, *Sys Mgr Man: Tuning*, 14–22
user profile entry
modifying, *Sys Mgr Man: Essentials*, 6–38
Maintenance operation protocol
See MOP
Management environment
clusterwide, *Sys Mgr Man: Essentials*, 2–15
individual nodes, *Sys Mgr Man: Essentials*, 2–14

Management environment (cont'd)
local and nonlocal environments, *Sys Mgr Man: Essentials*, 2–13
Manager, queue
See Queue managers
Managing a multiprocessing environment, *Sys Mgr Man: Tuning*, 26–3
tasks for, *Sys Mgr Man: Tuning*, 26–3
Managing a vector processing environment
tasks for, *Sys Mgr Man: Tuning*, 26–5
Managing devices
magnetic tape
tasks for, *Sys Mgr Man: Essentials*, 7–16
printers
setting characteristics, *Sys Mgr Man: Essentials*, 7–13
tasks for, *Sys Mgr Man: Essentials*, 7–13
tasks for, *Sys Mgr Man: Essentials*, 7–1
terminals
setting characteristics, *Sys Mgr Man: Essentials*, 7–10
tasks for, *Sys Mgr Man: Essentials*, 7–10
Managing installed software, *Sys Mgr Man: Essentials*, 3–34
Managing page, swap, and dump files
tasks for, *Sys Mgr Man: Tuning*, 15–1
Managing performance, *Sys Mgr Man: Tuning*, 16–1
See also Tuning
See Performance, managing
choosing a workload management strategy, *Sys Mgr Man: Tuning*, 16–3
considering hardware capacity, *Sys Mgr Man: Tuning*, 16–6
distributing work load, *Sys Mgr Man: Tuning*, 16–3
evaluating tuning success, *Sys Mgr Man: Tuning*, 16–6
installing images, *Sys Mgr Man: Tuning*, 16–9
knowing your work load, *Sys Mgr Man: Tuning*, 16–2
options for, *Sys Mgr Man: Tuning*, 16–6
system tuning, *Sys Mgr Man: Tuning*, 16–4
tasks for, *Sys Mgr Man: Tuning*, 16–1
with vector processing, *Sys Mgr Man: Tuning*, 26–7
Managing system parameters
tasks for, *Sys Mgr Man: Tuning*, 14–1
Managing the LAT database size, *Sys Mgr Man: Tuning*, 24–24
Managing the queue manager and queue database, *Sys Mgr Man: Essentials*, 12–1
Mandatory updates
definition, *Sys Mgr Man: Essentials*, 3–5
Mapping pointers
resetting for windows, *Sys Mgr Man: Essentials*, 8–24
Marginal vector consumer
detection of, *Sys Mgr Man: Tuning*, 26–7
Margin size
specifying in forms, *Sys Mgr Man: Essentials*, 13–42
Mass storage control protocol
See MSCP
Master command procedure
See UETP.COM procedure
Master file directories
See MFDs
Master file of queue database, *Sys Mgr Man: Essentials*, 12–4
See also Queue database
location
specifying, *Sys Mgr Man: Essentials*, 12–6
mounting of disk holding, *Sys Mgr Man: Essentials*, 12–6
QMAN$MASTER logical name, *Sys Mgr Man: Essentials*, 12–6
saving, *Sys Mgr Man: Essentials*, 12–12
MAXACCTJOBS process limit, *Sys Mgr Man: Essentials*, 6–43
MAXDETACH process limit, *Sys Mgr Man: Essentials*, 6–43
Maximum account jobs process limit, *Sys Mgr Man: Essentials*, 6–43
Maximum detached process limit, *Sys Mgr Man: Essentials*, 6–43
MAXJOBS process limit, *Sys Mgr Man: Essentials*, 6–43
MAXSYSGROUP system parameter, *Sys Mgr Man: Essentials*, 11–8
MAX_DUMPFILE symbol, *Sys Mgr Man: Tuning*, 15–24
MAX_PAGEFILE*n*_SIZE symbol, *Sys Mgr Man: Tuning*, 15–24
MAX_PAGEFILE symbol, *Sys Mgr Man: Tuning*, 15–24
MAX_ prefix for AUTOGEN, *Sys Mgr Man: Tuning*, 14–20
MAX_SWAPFILE*n*_SIZE symbol, *Sys Mgr Man: Tuning*, 15–24
MAX_SWAPFILE symbol, *Sys Mgr Man: Tuning*, 15–24
Media errors
analyzing, *Sys Mgr Man: Essentials*, 8–64
Memory
allotted to vector consumer processes, *Sys Mgr Man: Tuning*, 26–7
conserving with shareable images, *Sys Mgr Man: Tuning*, 16–11
images in, *Sys Mgr Man: Tuning*, 16–11
information captured in crash dump, *Sys Mgr Man: Tuning*, 15–2
physical dump, *Sys Mgr Man: Tuning*, 15–3, 15–11

Memory
information captured in crash dump (cont'd)
selective dump, *Sys Mgr Man: Tuning*, 15–3, 15–11
making efficient use of by installing images, *Sys Mgr Man: Tuning*, 16–8
paging, *Sys Mgr Man: Tuning*, 15–4
sections in, *Sys Mgr Man: Tuning*, 16–12
swapping, *Sys Mgr Man: Tuning*, 15–4
when large sizes prevent storing a complete system dump, *Sys Mgr Man: Tuning*, 15–11
Messages
broadcast
removing, *Sys Mgr Man: Tuning*, 20–10
defining welcome, *Sys Mgr Man: Essentials*, 5–14
DIBOL
starting the DIBOL message manager, *Sys Mgr Man: Essentials*, 5–16
enabling and disabling, *Sys Mgr Man: Tuning*, 18–17
error
removing, *Sys Mgr Man: Tuning*, 20–10
error log, *Sys Mgr Man: Tuning*, 18–6
indicating execution of startup command procedures
site-independent, *Sys Mgr Man: Essentials*, 4–5
site-specific, *Sys Mgr Man: Essentials*, 4–5
indicating high page or swap file fragmentation, *Sys Mgr Man: Tuning*, 15–27
indicating insufficient page file size, *Sys Mgr Man: Tuning*, 15–9
indicating lack of installed page file, *Sys Mgr Man: Essentials*, 5–6
indicating successful boot, *Sys Mgr Man: Essentials*, 4–5
indicating that a vector processor is not available, *Sys Mgr Man: Tuning*, 26–6
indicating that login is possible, *Sys Mgr Man: Essentials*, 4–5
login welcome, *Sys Mgr Man: Essentials*, 2–10
OPCOM
See OPCOM messages
operator replies, *Sys Mgr Man: Essentials*, 2–25; *Sys Mgr Man: Tuning*, 18–19
operator requests, *Sys Mgr Man: Essentials*, 2–25
question mark (?), *Sys Mgr Man: Essentials*, 4–16
saving in a file, *Sys Mgr Man: Essentials*, 3–29
security alarm, *Sys Mgr Man: Tuning*, 18–21
sending to users with OPCOM, *Sys Mgr Man: Essentials*, 2–22
suppressing the display, *Sys Mgr Man: Tuning*, 14–12
user requests, *Sys Mgr Man: Tuning*, 18–19
Messages (cont'd)
using when managing a system, *Sys Mgr Man: Essentials*, 2–5
WRITEBOOT, *Sys Mgr Man: Essentials*, 4–18
MFD (master file directory)
BACKUP stores save-set file in, *Sys Mgr Man: Tuning*, A–8
contains directory structure for volume set, *Sys Mgr Man: Essentials*, 8–33
definition, *Sys Mgr Man: Tuning*, A–8
on root volume in volume set, *Sys Mgr Man: Essentials*, 8–31
reserved file, *Sys Mgr Man: Tuning*, A–8
reserved files listed in, *Sys Mgr Man: Essentials*, 8–4
Minimerge and volume shadowing, *Sys Mgr Man: Essentials*, 10–41
Minimum startup
booting with, *Sys Mgr Man: Essentials*, 4–14
MIN_DUMPFILE symbol, *Sys Mgr Man: Tuning*, 15–24
MIN_PAGEFILE*n*_SIZE symbol, *Sys Mgr Man: Tuning*, 15–24
MIN_PAGEFILE symbol, *Sys Mgr Man: Tuning*, 15–24
MIN_ prefix for AUTOGEN, *Sys Mgr Man: Tuning*, 14–20
MIN_SWAPFILE*n*_SIZE symbol, *Sys Mgr Man: Tuning*, 15–24
MIN_SWAPFILE symbol, *Sys Mgr Man: Tuning*, 15–24
MODE logical name, *Sys Mgr Man: Tuning*, 17–18, 17–38
Mode menu, *Sys Mgr Man: Essentials*, 3–37
Modes
See Executive mode
See Privilege mode
Modifiable startup command procedure
See Site-specific startup command procedure
Modifying size of page, swap, and dump files
See also Changing size of page, swap, and dump files
See Changing size of page, swap, and dump files
Modifying system parameters
See also Changing system parameters
See Changing system parameters
MODPARAMS.DAT file, *Sys Mgr Man: Tuning*, 14–17, 14–18
ADD_ prefix, *Sys Mgr Man: Tuning*, 14–19
controlling page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–18, 15–22
specifying location, *Sys Mgr Man: Tuning*, 15–24
specifying size of individual files, *Sys Mgr Man: Tuning*, 15–24

MODPARAMS.DAT file
controlling page, swap, and dump files (cont'd)
specifying size of total file space, *Sys Mgr Man: Tuning*, 15–23
controlling parameter values set by AUTOGEN, *Sys Mgr Man: Tuning*, 14–4, 14–18
creating page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–17, 15–24
including external parameter files in, *Sys Mgr Man: Tuning*, 14–22
increasing parameter values, *Sys Mgr Man: Tuning*, 14–19
MAX_ prefix, *Sys Mgr Man: Tuning*, 14–20
MIN_ prefix, *Sys Mgr Man: Tuning*, 14–20
sample, *Sys Mgr Man: Tuning*, 14–17
specifying an alternate default startup command, *Sys Mgr Man: Essentials*, 4–13
specifying parameter values
absolute, *Sys Mgr Man: Tuning*, 14–20
maximum, *Sys Mgr Man: Tuning*, 14–20
minimum, *Sys Mgr Man: Tuning*, 14–20
storing your system parameter changes in, *Sys Mgr Man: Tuning*, 14–5
Modules
device control
See Device control modules
MONITOR.COM command procedure, *Sys Mgr Man: Tuning*, 18–41
used with Monitor utility, *Sys Mgr Man: Tuning*, 18–40
MONITOR command
See also Monitor utility
controlling amount of time between displays of screen images, *Sys Mgr Man: Tuning*, 18–38
recording data on time spent, *Sys Mgr Man: Tuning*, 18–37
recording file system and process data, *Sys Mgr Man: Tuning*, 18–37
routing display output, *Sys Mgr Man: Tuning*, 18–36
specifying how log request is to run, *Sys Mgr Man: Tuning*, 18–35
specifying input file, *Sys Mgr Man: Tuning*, 18–37
specifying node, *Sys Mgr Man: Tuning*, 18–36
specifying time of display, *Sys Mgr Man: Tuning*, 18–37
storing display output, *Sys Mgr Man: Tuning*, 18–38
Monitoring a multiprocessing environment, *Sys Mgr Man: Tuning*, 26–3
Monitor utility (MONITOR)
See also MONITOR command
analyzing disk use with, *Sys Mgr Man: Essentials*, 8–8
class types, *Sys Mgr Man: Tuning*, 18–32
Monitor utility (MONITOR) (cont'd)
command procedures, *Sys Mgr Man: Tuning*, 18–40
for cluster summaries, *Sys Mgr Man: Tuning*, 18–41
to initiate continuous recording, *Sys Mgr Man: Tuning*, 18–42
to produce cluster summaries, *Sys Mgr Man: Tuning*, 18–43
description, *Sys Mgr Man: Tuning*, 18–32
directing display, *Sys Mgr Man: Tuning*, 18–34
displaying and recording concurrently, *Sys Mgr Man: Tuning*, 18–37
displaying live monitoring, *Sys Mgr Man: Tuning*, 18–35
displaying network information, *Sys Mgr Man: Tuning*, 21–17
entering commands, *Sys Mgr Man: Tuning*, 18–34
exiting from, *Sys Mgr Man: Tuning*, 18–34
invoking, *Sys Mgr Man: Tuning*, 18–34
logical name for, *Sys Mgr Man: Essentials*, 5–8
MONITOR.COM command procedure
using to create summary file, *Sys Mgr Man: Tuning*, 18–40
MONSUM.COM command procedure
using to generate clusterwide multifile summary reports, *Sys Mgr Man: Tuning*, 18–40
moving log file to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
parameters, *Sys Mgr Man: Tuning*, 18–32
playing back monitoring, *Sys Mgr Man: Tuning*, 18–37
playing back remote monitoring, *Sys Mgr Man: Tuning*, 18–39
producing reports, *Sys Mgr Man: Tuning*, 18–40
qualifiers, *Sys Mgr Man: Tuning*, 18–34
recording live monitoring, *Sys Mgr Man: Tuning*, 18–36
reports, *Sys Mgr Man: Tuning*, 18–41
rerecording monitoring, *Sys Mgr Man: Tuning*, 18–40
running continuously, *Sys Mgr Man: Tuning*, 18–40
SUBMON.COM procedure
using to start MONITOR.COM as a detached process, *Sys Mgr Man: Tuning*, 18–40
version compatibility, *Sys Mgr Man: Tuning*, 18–44
MONSUM.COM command procedure, *Sys Mgr Man: Tuning*, 18–43
used with Monitor utility, *Sys Mgr Man: Tuning*, 18–40

MONSUM.COM command procedure (cont'd)
using to generate clusterwide multifile summary reports, *Sys Mgr Man: Tuning*, 18–40
MOP downline load service
clearing counters, *Sys Mgr Man: Tuning*, 22–25
console carrier, *Sys Mgr Man: Tuning*, 22–26
disabling, *Sys Mgr Man: Tuning*, 22–23
displaying status and counters, *Sys Mgr Man: Tuning*, 22–23, 22–24
enabling, *Sys Mgr Man: Tuning*, 22–23
LANACP server, *Sys Mgr Man: Tuning*, 22–23
LAN MOP, *Sys Mgr Man: Tuning*, 22–20
coexisting with DECnet MOP, *Sys Mgr Man: Tuning*, 22–20
migrating DECnet MOP to LAN MOP, *Sys Mgr Man: Tuning*, 22–20
sample setup, *Sys Mgr Man: Tuning*, 22–22
setup with CLUSTER_CONFIG.COM, *Sys Mgr Man: Tuning*, 22–21
trigger boot, *Sys Mgr Man: Tuning*, 22–27
MOP protocol
downline loading, *Sys Mgr Man: Tuning*, 23–2
Motif
See DECwindows Motif interface
MOUNT command
See also Mounting volumes
adding to volume sets, *Sys Mgr Man: Essentials*, 8–22
assigning a volume set name, *Sys Mgr Man: Essentials*, 8–31
avoiding use of /CLUSTER with SYSMAN DO command, *Sys Mgr Man: Tuning*, 20–21
controlling whether header labels are written to a volume, *Sys Mgr Man: Essentials*, 8–24
creating a public volume, *Sys Mgr Man: Essentials*, 8–24
creating disk volume sets, *Sys Mgr Man: Essentials*, 8–30
creating volume sets, *Sys Mgr Man: Essentials*, 8–22
disabling automatic notification of mount failures, *Sys Mgr Man: Essentials*, 8–22
disabling automatic volume switching, *Sys Mgr Man: Essentials*, 8–40
disabling mount verification feature for disks, *Sys Mgr Man: Essentials*, 8–23
disabling mount verification feature for tapes, *Sys Mgr Man: Essentials*, 8–24
disk and tape volumes, *Sys Mgr Man: Essentials*, 8–20
enabling automatic notification of mount failures, *Sys Mgr Man: Essentials*, 8–22
enabling mount verification feature for disks, *Sys Mgr Man: Essentials*, 8–23
enabling mount verification feature for tapes, *Sys Mgr Man: Essentials*, 8–24
MOUNT command (cont'd)
enabling the processing of subsystem ACEs, *Sys Mgr Man: Essentials*, 8–23
enabling the write cache for a tape device, *Sys Mgr Man: Essentials*, 8–24
ensuring that tape volume set has been initialized, *Sys Mgr Man: Essentials*, 8–40
establishing default file attributes for records on ISO 9660 media, *Sys Mgr Man: Essentials*, 8–24
foreign volumes, *Sys Mgr Man: Essentials*, 8–22
for foreign volumes, *Sys Mgr Man: Essentials*, 8–24, 9–13
including a quoted text string as part of mount request, *Sys Mgr Man: Essentials*, 8–22
inhibiting access checks, *Sys Mgr Man: Essentials*, 8–25
in system startup
for remote InfoServer devices, *Sys Mgr Man: Tuning*, 23–14
mounting page and swap file disks, *Sys Mgr Man: Essentials*, 5–6
special consideration about operator assistance, *Sys Mgr Man: Essentials*, 5–11
in VMScluster environment, *Sys Mgr Man: Essentials*, 8–21, 8–22
ISO 9660 media, *Sys Mgr Man: Essentials*, 8–23
overriding expiration date, *Sys Mgr Man: Essentials*, 9–14
overriding protection, *Sys Mgr Man: Essentials*, 8–25
overriding protection checks, *Sys Mgr Man: Essentials*, 8–23
overriding the volume identification field, *Sys Mgr Man: Essentials*, 8–25
overriding UIC in second volume label, *Sys Mgr Man: Essentials*, 8–25
parameters, *Sys Mgr Man: Essentials*, 8–21
protection codes, *Sys Mgr Man: Essentials*, 9–12
public volumes, *Sys Mgr Man: Essentials*, 8–21
qualifiers, *Sys Mgr Man: Essentials*, 8–22, 8–24
requesting operator assistance, *Sys Mgr Man: Essentials*, 8–22
resetting the number mapping pointers, *Sys Mgr Man: Essentials*, 8–24
specifying block size for tape, *Sys Mgr Man: Essentials*, 8–24
specifying number of bytes in each record, *Sys Mgr Man: Essentials*, 8–26
specifying record size, *Sys Mgr Man: Essentials*, 8–26
specifying that other users can access current volume, *Sys Mgr Man: Essentials*, 8–23

MOUNT command (cont'd)
specifying the number of directories that the system keeps in memory, *Sys Mgr Man: Essentials*, 8–22
specifying the number of disk blocks allocated, *Sys Mgr Man: Essentials*, 8–22
specifying UIC, *Sys Mgr Man: Essentials*, 8–25
suspending quota operation on a volume, *Sys Mgr Man: Essentials*, 8–52
tape, *Sys Mgr Man: Essentials*, 9–21
to mount disk holding page and swap files, *Sys Mgr Man: Tuning*, 15–20
Mounted forms
matching stock, *Sys Mgr Man: Essentials*, 13–42
Mounting forms, *Sys Mgr Man: Essentials*, 13–64
Mounting of queue database disk, *Sys Mgr Man: Essentials*, 12–6
Mounting volumes, *Sys Mgr Man: Essentials*, 10–15
See also MOUNT command
disks, *Sys Mgr Man: Essentials*, 8–20
for queue database files, *Sys Mgr Man: Essentials*, 12–6
in system startup, *Sys Mgr Man: Essentials*, 5–11
early, *Sys Mgr Man: Essentials*, 5–12
for page and swap files, *Sys Mgr Man: Essentials*, 5–6
InfoServer, *Sys Mgr Man: Tuning*, 23–14
special consideration about operator assistance, *Sys Mgr Man: Essentials*, 5–11
virtual device unit, *Sys Mgr Man: Tuning*, 23–14
if device is unavailable, *Sys Mgr Man: Essentials*, 8–27
in a VMScluster environment, *Sys Mgr Man: Essentials*, 8–21
operator assistance, *Sys Mgr Man: Essentials*, 8–18
public, *Sys Mgr Man: Essentials*, 8–21
substituting, *Sys Mgr Man: Essentials*, 8–27
tape, *Sys Mgr Man: Essentials*, 8–20
tape volume sets, *Sys Mgr Man: Essentials*, 8–37
Mounting volume sets
See also MOUNT command
disk, *Sys Mgr Man: Essentials*, 8–31, 8–32, 8–33
tape
with automatic volume switching disabled, *Sys Mgr Man: Essentials*, 8–40
Mount messages
disabling with SUBSYSTEM qualifier, *Sys Mgr Man: Essentials*, 8–23
Mount utility (MOUNT)
using to mount ISO 9660 volume sets, *Sys Mgr Man: Essentials*, 8–34
Mount verification, *Sys Mgr Man: Essentials*, 8–58
aborted
OPCOM message, *Sys Mgr Man: Essentials*, 8–63
aborting by dismounting, *Sys Mgr Man: Essentials*, 8–62
canceling, *Sys Mgr Man: Essentials*, 8–62, 8–63
definition, *Sys Mgr Man: Essentials*, 8–58
device off line, *Sys Mgr Man: Essentials*, 8–59, 8–61
device write-locked, *Sys Mgr Man: Essentials*, 8–61
enabling, *Sys Mgr Man: Essentials*, 8–60
enabling and disabling, *Sys Mgr Man: Essentials*, 8–23
for tapes, *Sys Mgr Man: Essentials*, 8–24
messages, *Sys Mgr Man: Essentials*, 8–60
operation of, *Sys Mgr Man: Essentials*, 8–59
timeout, *Sys Mgr Man: Essentials*, 8–60
OPCOM message, *Sys Mgr Man: Essentials*, 8–60
MOVE keypad function, *Sys Mgr Man: Tuning*, 20–7
Moving jobs from one queue to another, *Sys Mgr Man: Essentials*, 13–57
MSCP (mass storage control protocol), *Sys Mgr Man: Tuning*, 20–2
MSGHLP
See Help Message utility
.MSGHLP$DATA files
adding to the Help Message database, *Sys Mgr Man: Essentials*, 5–28
MSGHLP$LIBRARY.MSGHLP$DATA file, *Sys Mgr Man: Essentials*, 5–28
MSGHLP$LIBRARY logical name, *Sys Mgr Man: Essentials*, 5–28
MTACP (magnetic tape ancillary control process)
definition, *Sys Mgr Man: Essentials*, 8–6
Multiple queue managers, *Sys Mgr Man: Essentials*, 12–3, 12–10
commands affected by, *Sys Mgr Man: Essentials*, 12–4
managing queues with, *Sys Mgr Man: Essentials*, 12–10
moving queues, *Sys Mgr Man: Essentials*, 12–10
restriction, *Sys Mgr Man: Essentials*, 12–3
specifying names of, *Sys Mgr Man: Essentials*, 12–4
specifying queue manager name, *Sys Mgr Man: Essentials*, 12–4, 12–10
use of queue database, *Sys Mgr Man: Essentials*, 12–4

Multiple-site network, *Sys Mgr Man: Tuning*, 21–9
Multiprocessing, *Sys Mgr Man: Tuning*, 26–2
  definition, *Sys Mgr Man: Tuning*, 26–2
  displaying information, *Sys Mgr Man: Tuning*, 26–3
  monitoring, *Sys Mgr Man: Tuning*, 26–3
  system parameters, *Sys Mgr Man: Tuning*, 26–3
MULTIPROCESSING system parameter, *Sys Mgr Man: Tuning*, 26–3
MVTIMEOUT system parameter, *Sys Mgr Man: Essentials*, 8–60, 8–62

N

Namespaces
  in DECnet/OSI full name, *Sys Mgr Man: Tuning*, 21–3
Naming conventions
  for queue and journal files for additional queue managers, *Sys Mgr Man: Essentials*, 12–5
  of devices, *Sys Mgr Man: Essentials*, 7–1
    in a VMScluster environment, *Sys Mgr Man: Essentials*, 7–1
    virtual terminals, *Sys Mgr Man: Essentials*, 7–12
NCP (Network Control Program)
  commands, *Sys Mgr Man: Tuning*, 21–11 to 21–15
    CLEAR, *Sys Mgr Man: Tuning*, 21–11
    DEFINE, *Sys Mgr Man: Tuning*, 21–11
    LIST, *Sys Mgr Man: Tuning*, 21–15
    PURGE, *Sys Mgr Man: Tuning*, 21–11
    SET, *Sys Mgr Man: Tuning*, 21–11
    SHOW, *Sys Mgr Man: Tuning*, 21–15
  controlling proxy access, *Sys Mgr Man: Essentials*, 6–38
  display commands, *Sys Mgr Man: Tuning*, 21–15
  Ethernet configurator, *Sys Mgr Man: Tuning*, 21–17
  testing network, *Sys Mgr Man: Tuning*, 21–17
  using to build or modify configuration database, *Sys Mgr Man: Tuning*, 21–11
NET$PROXY.DAT file, *Sys Mgr Man: Essentials*, 11–7
NET$PROXY.DAT files, *Sys Mgr Man: Essentials*, 6–4, 6–34
  moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
NET$PROXY logical name
  defining to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
NETCONFIG.COM command procedure
  configuring a node automatically, *Sys Mgr Man: Tuning*, 21–11, 21–12
NETDRIVER (network driver)
  connecting, *Sys Mgr Man: Essentials*, 7–8, 7–9
NETNODE_REMOTE logical name
  defining during system startup, *Sys Mgr Man: Essentials*, 5–8
NETPROXY.DAT files, *Sys Mgr Man: Essentials*, 6–4, 6–34
  moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
NETPROXY logical name
  defining during system startup, *Sys Mgr Man: Essentials*, 5–8
  defining to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
Network communications device
  connecting, *Sys Mgr Man: Essentials*, 7–8, 7–9
Network Control Program
  See NCP
Network identifiers, *Sys Mgr Man: Essentials*, 11–11
Network interface
  See DECnet
Network proxy accounts, *Sys Mgr Man: Essentials*, 6–34
Networks
  See also DECnet
  See also Nodes
  area routing, *Sys Mgr Man: Tuning*, 21–6
  becoming node in, *Sys Mgr Man: Tuning*, 21–11
  bridge, *Sys Mgr Man: Tuning*, 21–7
  channel, *Sys Mgr Man: Tuning*, 21–5
  circuit
    definition, *Sys Mgr Man: Tuning*, 21–4
  communications device, *Sys Mgr Man: Essentials*, 7–9
  communications line, *Sys Mgr Man: Tuning*, 21–12
  configuration command procedure
    See NETCONFIG.COM
  configurations, *Sys Mgr Man: Tuning*, 21–6
    See also Configurations, network
  connecting to existing, *Sys Mgr Man: Tuning*, 21–12
  connections, *Sys Mgr Man: Tuning*, 21–8
    local area, *Sys Mgr Man: Tuning*, 21–8
    multipoint, *Sys Mgr Man: Tuning*, 21–9
    point-to-point, *Sys Mgr Man: Tuning*, 21–8
    wide area, *Sys Mgr Man: Tuning*, 21–8
    worldwide, *Sys Mgr Man: Tuning*, 21–9
  definition, *Sys Mgr Man: Tuning*, 21–4
  end node, *Sys Mgr Man: Tuning*, 21–6
  Ethernet
    definition, *Sys Mgr Man: Tuning*, 21–5
  getting information about nodes, *Sys Mgr Man: Tuning*, 21–10
  interface

Networks
interface (cont'd)
OpenVMS, *Sys Mgr Man: Tuning*, 21–7
Internet, *Sys Mgr Man: Tuning*, 21–13
LAN enhancements, *Sys Mgr Man: Tuning*, 22–1
line
definition, *Sys Mgr Man: Tuning*, 21–4
logical link
definition, *Sys Mgr Man: Tuning*, 21–4
loopback tests, *Sys Mgr Man: Tuning*, 21–17
managing a node, *Sys Mgr Man: Tuning*, 21–14
monitoring tools, *Sys Mgr Man: Tuning*, 21–15
DTS/DTR, *Sys Mgr Man: Tuning*, 21–17
Monitor utility, *Sys Mgr Man: Tuning*, 21–17
NCP display commands, *Sys Mgr Man: Tuning*, 21–15
NCP Ethernet configurator, *Sys Mgr Man: Tuning*, 21–17
multinode, *Sys Mgr Man: Tuning*, 21–5
multiple-area, *Sys Mgr Man: Tuning*, 21–6
multiple-site, *Sys Mgr Man: Tuning*, 21–9
nodes
configuring, *Sys Mgr Man: Tuning*, 21–12
connecting, *Sys Mgr Man: Tuning*, 21–8
definition, *Sys Mgr Man: Tuning*, 21–4
object
definition, *Sys Mgr Man: Tuning*, 21–4
planning configuration, *Sys Mgr Man: Tuning*, 21–12
proxy database
creating, *Sys Mgr Man: Essentials*, 6–35
logical name defining location, *Sys Mgr Man: Essentials*, 5–8
remote node database
logical name defining location, *Sys Mgr Man: Essentials*, 5–8
router, *Sys Mgr Man: Tuning*, 21–5
routing
adaptive, *Sys Mgr Man: Tuning*, 21–5
definition, *Sys Mgr Man: Tuning*, 21–5
save sets, *Sys Mgr Man: Essentials*, 10–7
security
providing for, *Sys Mgr Man: Tuning*, 21–12
starting in system startup, *Sys Mgr Man: Essentials*, 5–15
testing
using NCP (Network Control Program), *Sys Mgr Man: Tuning*, 21–17
verifying connection to, *Sys Mgr Man: Tuning*, 21–12
Node names
assigning, *Sys Mgr Man: Tuning*, 21–3
DECnet/OSI full names, *Sys Mgr Man: Tuning*, 21–2
Node names (cont'd)
DECnet for OpenVMS, *Sys Mgr Man: Tuning*, 21–2
required for OpenVMS InfoServer Client startup, *Sys Mgr Man: Tuning*, 23–10
setting up a system for assigning, *Sys Mgr Man: Tuning*, 21–3
specifying in MAIL, *Sys Mgr Man: Essentials*, 5–33
Nodes
See also Networks
adjacent, *Sys Mgr Man: Tuning*, 21–4
configuring in a network, *Sys Mgr Man: Tuning*, 21–12
connecting to network, *Sys Mgr Man: Tuning*, 21–8
definition, *Sys Mgr Man: Tuning*, 21–4
deleting node information, *Sys Mgr Man: Tuning*, 22–20
end, *Sys Mgr Man: Tuning*, 21–6
establishing in network, *Sys Mgr Man: Tuning*, 21–11
getting information about other nodes in network, *Sys Mgr Man: Tuning*, 21–10
in LAT database, *Sys Mgr Man: Tuning*, 24–14
monitoring, *Sys Mgr Man: Tuning*, 21–14
multiple network, *Sys Mgr Man: Tuning*, 21–5
nonrouting, *Sys Mgr Man: Tuning*, 21–6
preparing system to become network node, *Sys Mgr Man: Tuning*, 21–12
providing security for, *Sys Mgr Man: Tuning*, 21–12
router definition, *Sys Mgr Man: Tuning*, 21–5
routing, *Sys Mgr Man: Tuning*, 21–6
setting node information, *Sys Mgr Man: Tuning*, 22–18
tools to monitor network, *Sys Mgr Man: Tuning*, 21–15
transferring files between, *Sys Mgr Man: Essentials*, 9–24
Nondeductible resource, *Sys Mgr Man: Essentials*, 6–3
Nonrouting nodes
See End nodes
Nonstop boot
definition, *Sys Mgr Man: Essentials*, 4–3
performing, *Sys Mgr Man: Essentials*, 4–3
NPAGEDYN system parameter, *Sys Mgr Man: Tuning*, 26–7
NUM_ETHERADAPT symbol, *Sys Mgr Man: Tuning*, 14–21
NUM_NODES symbol, *Sys Mgr Man: Tuning*, 14–21

O

Objects
  network
    definition, *Sys Mgr Man: Tuning*, 21–4
  protecting volume, *Sys Mgr Man: Essentials*, 8–14
OLDUETP.LOG file, *Sys Mgr Man: Tuning*, 17–20
OPA0: device, *Sys Mgr Man: Essentials*, 2–23
OPC$LOGFILE_CLASSES logical name, *Sys Mgr Man: Tuning*, 18–23
OPC$LOGFILE_ENABLE logical name, *Sys Mgr Man: Tuning*, 18–23
OPC$LOGFILE_NAME logical name, *Sys Mgr Man: Tuning*, 18–24
  for operator log file, *Sys Mgr Man: Tuning*, 18–22
OPC$OPA0_ENABL logical name, *Sys Mgr Man: Tuning*, 18–24
OPCCRASH.EXE program, *Sys Mgr Man: Essentials*, 4–27
OPCOM
  messages
    displayed during LANACP start, *Sys Mgr Man: Tuning*, 22–22
    generated by LANACP, *Sys Mgr Man: Tuning*, 22–25
OPCOM (Operator Communication Manager), *Sys Mgr Man: Essentials*, 10–17
  See also Operator log file
  automatic restart, *Sys Mgr Man: Essentials*, 2–22
  classes
    enabling, *Sys Mgr Man: Essentials*, 2–23
      enabling and disabling for log file, *Sys Mgr Man: Tuning*, 18–23
  communicating with operators, *Sys Mgr Man: Essentials*, 8–18
  communicating with users, *Sys Mgr Man: Essentials*, 8–18
  components of, *Sys Mgr Man: Essentials*, 2–19
  default behavior, *Sys Mgr Man: Essentials*, 2–21
  disabling operator terminals, *Sys Mgr Man: Essentials*, 2–24
  enabling operator classes, *Sys Mgr Man: Essentials*, 2–23
  enabling operator terminals, *Sys Mgr Man: Essentials*, 2–23
  failure, *Sys Mgr Man: Essentials*, 2–22
  illustration, *Sys Mgr Man: Essentials*, 2–19
  log file, *Sys Mgr Man: Essentials*, 2–21; *Sys Mgr Man: Tuning*, 18–17
  messages
    See OPCOM messages
OPCOM (Operator Communication Manager) (cont'd)
  mount verification, *Sys Mgr Man: Essentials*, 8–60
  operator log file
    See Operator log file
  operator terminals, *Sys Mgr Man: Essentials*, 2–21
  printing operator log file, *Sys Mgr Man: Tuning*, 18–24
  process, *Sys Mgr Man: Essentials*, 2–21
    creation during system startup, *Sys Mgr Man: Essentials*, 5–5
  process dump file, *Sys Mgr Man: Essentials*, 2–22
  replying to operator requests, *Sys Mgr Man: Essentials*, 2–25
  requirements, *Sys Mgr Man: Essentials*, 2–21
  restarting operator log file, *Sys Mgr Man: Tuning*, 18–24
  sending requests to an operator, *Sys Mgr Man: Essentials*, 2–25
  setting up operator log file, *Sys Mgr Man: Tuning*, 18–22
  specifying default state of operator log file, *Sys Mgr Man: Tuning*, 18–23
  startup of, *Sys Mgr Man: Essentials*, 2–22
  uses of, *Sys Mgr Man: Essentials*, 2–19
  using to request assistance, *Sys Mgr Man: Essentials*, 10–17
OPCOM.DMP process dump file, *Sys Mgr Man: Essentials*, 2–22
OPCOM messages, *Sys Mgr Man: Essentials*, 2–21
  continuation volume request, *Sys Mgr Man: Essentials*, 8–40, 9–24
  controlling, *Sys Mgr Man: Essentials*, 2–22
  mount request message, *Sys Mgr Man: Essentials*, 8–26
  mount verification, *Sys Mgr Man: Essentials*, 8–60
    aborted message, *Sys Mgr Man: Essentials*, 8–63
    timeout message, *Sys Mgr Man: Essentials*, 8–60
  operator reply, *Sys Mgr Man: Tuning*, 18–19
  security alarm, *Sys Mgr Man: Tuning*, 18–21
  sending, *Sys Mgr Man: Essentials*, 2–22
  SYSGEN, *Sys Mgr Man: Tuning*, 18–20
  user request, *Sys Mgr Man: Tuning*, 18–19
Open file limit, *Sys Mgr Man: Essentials*, 6–43
Open images, *Sys Mgr Man: Tuning*, 16–10, 16–12
OpenVMS Management Station
  description, *Sys Mgr Man: Essentials*, 2–2
  documentation, *Sys Mgr Man: Essentials*, 2–4
  features, *Sys Mgr Man: Essentials*, 2–4

Operating systems
building on another disk, *Sys Mgr Man: Essentials*, 2–27
copying to another system disk, *Sys Mgr Man: Essentials*, 2–29
installing, *Sys Mgr Man: Essentials*, 3–4
updating, *Sys Mgr Man: Essentials*, 3–5
upgrading, *Sys Mgr Man: Essentials*, 3–5
Operation button on POLYCENTER Software Installation utility, *Sys Mgr Man: Essentials*, 3–37, 3–39
OPERATOR.LOG file
See Operator log file
Operator assistance
operator classes, *Sys Mgr Man: Essentials*, 2–23
replying to operator requests, *Sys Mgr Man: Essentials*, 2–25
with MOUNT command, *Sys Mgr Man: Essentials*, 8–18
Operator classes
See OPCOM, classes
Operator communication manager
See OPCOM
Operator consoles
enabling in system startup, *Sys Mgr Man: Essentials*, 5–5
Operator log file
See also OPCOM
See also OPCOM messages
closing current and opening new, *Sys Mgr Man: Tuning*, 18–17
creating new, *Sys Mgr Man: Tuning*, 18–22
definition, *Sys Mgr Man: Essentials*, 2–21
device status message, *Sys Mgr Man: Tuning*, 18–17, 18–22
enabling and disabling classes, *Sys Mgr Man: Tuning*, 18–23
enabling in system startup, *Sys Mgr Man: Essentials*, 5–5
initialization message, *Sys Mgr Man: Tuning*, 18–17
logging user requests in, *Sys Mgr Man: Tuning*, 18–19
maintaining, *Sys Mgr Man: Tuning*, 18–24
printing, *Sys Mgr Man: Tuning*, 18–22, 18–24
purging, *Sys Mgr Man: Tuning*, 18–24
during system startup, *Sys Mgr Man: Essentials*, 5–14
recording changes to system parameters, *Sys Mgr Man: Tuning*, 18–20
request identification number, *Sys Mgr Man: Tuning*, 18–19
response recorded in, *Sys Mgr Man: Tuning*, 18–19
restarting, *Sys Mgr Man: Tuning*, 18–24
Operator log file (cont'd)
security alarm messages in, *Sys Mgr Man: Tuning*, 18–21
setting up, *Sys Mgr Man: Tuning*, 18–16, 18–22
specifying default state, *Sys Mgr Man: Tuning*, 18–23
terminal enable and disable message, *Sys Mgr Man: Tuning*, 18–22
troubleshooting the queue manager, *Sys Mgr Man: Essentials*, 12–14
user request and operator reply messages, *Sys Mgr Man: Tuning*, 18–22
volume mount and dismount messages, *Sys Mgr Man: Tuning*, 18–22
Operators (computer)
assistance with MOUNT command in system startup, *Sys Mgr Man: Essentials*, 5–11
classes of, *Sys Mgr Man: Essentials*, 2–23
replying to requests, *Sys Mgr Man: Essentials*, 2–25
requesting assistance from, *Sys Mgr Man: Essentials*, 8–18
sending requests to, *Sys Mgr Man: Essentials*, 2–25
Operator terminals, *Sys Mgr Man: Essentials*, 2–21
designating, *Sys Mgr Man: Essentials*, 2–23
in batch or SYSTARTUP, *Sys Mgr Man: Essentials*, 2–21
enabling and disabling, *Sys Mgr Man: Essentials*, 2–23, 2–24; *Sys Mgr Man: Tuning*, 18–17
security alarms, *Sys Mgr Man: Tuning*, 18–29
security terminal, *Sys Mgr Man: Tuning*, 18–21
setting up, *Sys Mgr Man: Essentials*, 8–18
user request, *Sys Mgr Man: Essentials*, 8–18
Optional files
adding and removing, *Sys Mgr Man: Essentials*, 5–1
Option list
parameter for VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–12
Options
for software product installations, *Sys Mgr Man: Essentials*, 3–30
Outgoing LAT connections, *Sys Mgr Man: Tuning*, 24–3, 24–14
enabling with LAT software, *Sys Mgr Man: Tuning*, 24–22
Output a newly created PCF, *Sys Mgr Man: Essentials*, 3–30
Output devices
See Printers
See Terminals

Output during UETP
terminal and line printer, *Sys Mgr Man: Tuning*, 17–33
Output execution queues
See also Execution queues
See also Output queues
definition, *Sys Mgr Man: Essentials*, 13–3
Output jobs
See also Output queues
aligning forms, *Sys Mgr Man: Essentials*, 13–76
allowing to complete before stopping a queue, *Sys Mgr Man: Essentials*, 13–55
changing scheduling priority, *Sys Mgr Man: Essentials*, 13–72
controlling, *Sys Mgr Man: Essentials*, 13–68
controlling print position and alignment, *Sys Mgr Man: Essentials*, 13–75, 13–76
deleting, *Sys Mgr Man: Essentials*, 13–74
holding and releasing, *Sys Mgr Man: Essentials*, 13–70
modifying, *Sys Mgr Man: Essentials*, 13–70
monitoring, *Sys Mgr Man: Essentials*, 13–68
requeuing
executing, *Sys Mgr Man: Essentials*, 13–72
pending, *Sys Mgr Man: Essentials*, 13–73
resuming printing, *Sys Mgr Man: Essentials*, 13–75, 13–76
retaining in a queue, *Sys Mgr Man: Essentials*, 13–73
scheduling, *Sys Mgr Man: Essentials*, 13–32, 13–72
status
See Job status
suspending, *Sys Mgr Man: Essentials*, 13–75
Output queues
See also Output jobs
See also Output queuing environment
allowing jobs to complete before stopping, *Sys Mgr Man: Essentials*, 13–55
assigning a default form, *Sys Mgr Man: Essentials*, 13–63
canceling characteristics, *Sys Mgr Man: Essentials*, 13–59
changing DEFAULT form, *Sys Mgr Man: Essentials*, 13–63
commands for managing, *Sys Mgr Man: Essentials*, 13–46
controlling line overflow in forms, *Sys Mgr Man: Essentials*, 13–42
creating, *Sys Mgr Man: Essentials*, 13–15
defining a form, *Sys Mgr Man: Essentials*, 13–61
deleting, *Sys Mgr Man: Essentials*, 13–57
execution
description, *Sys Mgr Man: Essentials*, 13–3
Output queues
execution (cont'd)
printer, *Sys Mgr Man: Essentials*, 13–3
server, *Sys Mgr Man: Essentials*, 13–3
terminal, *Sys Mgr Man: Essentials*, 13–3
mounting a form on, *Sys Mgr Man: Essentials*, 13–64
on standalone workstations, *Sys Mgr Man: Essentials*, 13–8
options, *Sys Mgr Man: Essentials*, 13–18
banner pages, *Sys Mgr Man: Essentials*, 13–33
characteristics, *Sys Mgr Man: Essentials*, 13–28
controlling page and line overflow, *Sys Mgr Man: Essentials*, 13–43
device control libraries, *Sys Mgr Man: Essentials*, 13–44
forms, *Sys Mgr Man: Essentials*, 13–42
qualifiers for specifying, *Sys Mgr Man: Essentials*, 13–20 to 13–21
restricting access, *Sys Mgr Man: Essentials*, 13–22
retaining jobs, *Sys Mgr Man: Essentials*, 13–26
suppressing form feed, *Sys Mgr Man: Essentials*, 13–44
order of device control module output, *Sys Mgr Man: Essentials*, 13–45
pausing, *Sys Mgr Man: Essentials*, 13–53, 13–75
to align position of print for preprinted forms, *Sys Mgr Man: Essentials*, 13–75, 13–76
to change position of print, *Sys Mgr Man: Essentials*, 13–75
rerouting jobs in, *Sys Mgr Man: Essentials*, 13–56
specifying page and margin size in forms, *Sys Mgr Man: Essentials*, 13–42
starting, *Sys Mgr Man: Essentials*, 13–15
status, *Sys Mgr Man: Essentials*, 13–50
stopping, *Sys Mgr Man: Essentials*, 13–54, 13–55
before shutting down a node, *Sys Mgr Man: Essentials*, 13–55
Output queuing environment
for LAT printers, *Sys Mgr Man: Essentials*, 13–10
for mixed printers, *Sys Mgr Man: Essentials*, 13–9
for multiple printers of the same kind, *Sys Mgr Man: Essentials*, 13–11
in VMScluster environments, *Sys Mgr Man: Essentials*, 13–12
on a standalone workstation, *Sys Mgr Man: Essentials*, 13–8

Output queuing environment (cont'd)
sample configurations, *Sys Mgr Man: Essentials*, 13–8 to 13–14
single printer, *Sys Mgr Man: Essentials*, 13–8
spooled printers, *Sys Mgr Man: Essentials*, 13–13
steps for setting up, *Sys Mgr Man: Essentials*, 13–14
Output specifier
to the BACKUP command, *Sys Mgr Man: Essentials*, 10–5
Overdraft limit
user exceeding quota, *Sys Mgr Man: Essentials*, 8–49
Overflow
See Lines, overflow
See Page overflow
Overriding DECnet parameters, *Sys Mgr Man: Tuning*, 14–21
Owner
category of user access, *Sys Mgr Man: Essentials*, 11–8
Ownership
file
displaying, *Sys Mgr Man: Essentials*, 9–7

P

PAGEFILE.SYS file, *Sys Mgr Man: Tuning*, 15–2, 15–4
See also Page files
as system dump file, *Sys Mgr Man: Tuning*, 15–2, 15–7
required location, *Sys Mgr Man: Tuning*, 15–3
requirement
location, *Sys Mgr Man: Tuning*, 16–8
PAGEFILE*n*_NAME symbol, *Sys Mgr Man: Tuning*, 15–17, 15–24
PAGEFILE*n*_SIZE symbol, *Sys Mgr Man: Tuning*, 15–18, 15–24
Page files
as system dump file, *Sys Mgr Man: Tuning*, 15–2, 15–7, 15–17
releasing dump from, *Sys Mgr Man: Tuning*, 15–16
size required for, *Sys Mgr Man: Tuning*, 15–4
changing sizes
with SWAPFILES.COM, *Sys Mgr Man: Tuning*, 15–25
creating
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–17, 15–24
with SYSGEN, *Sys Mgr Man: Tuning*, 15–18
definition, *Sys Mgr Man: Tuning*, 15–4
deleting after creating a new version, *Sys Mgr Man: Tuning*, 15–28
Page files (cont'd)
displaying, *Sys Mgr Man: Tuning*, 15–6
size calculated by AUTOGEN, *Sys Mgr Man: Tuning*, 15–18, 15–22
fragmentation of, *Sys Mgr Man: Tuning*, 15–27
freeing dump information from, *Sys Mgr Man: Tuning*, 15–4, 15–16
installing
in system startup, *Sys Mgr Man: Essentials*, 5–6; *Sys Mgr Man: Tuning*, 15–5, 15–20, 15–21
when resized with AUTOGEN, *Sys Mgr Man: Tuning*, 15–18, 15–22
with SYPAGSWPFILES.COM procedure, *Sys Mgr Man: Tuning*, 15–20
with SYSGEN, *Sys Mgr Man: Tuning*, 15–19
limiting usage of, *Sys Mgr Man: Tuning*, 15–10
location
specifying for individual files, *Sys Mgr Man: Tuning*, 15–24
message
indicating high fragmentation, *Sys Mgr Man: Tuning*, 15–27
indicating insufficient size, *Sys Mgr Man: Tuning*, 15–9
indicating lack of installed, *Sys Mgr Man: Essentials*, 5–6
monitoring usage of, *Sys Mgr Man: Tuning*, 15–6
mounting disk during system startup, *Sys Mgr Man: Essentials*, 5–6; *Sys Mgr Man: Tuning*, 15–20
moving to improve performance, *Sys Mgr Man: Tuning*, 16–8
on a satellite, *Sys Mgr Man: Tuning*, 15–20
primary, *Sys Mgr Man: Tuning*, 15–5
purging, *Sys Mgr Man: Tuning*, 15–28
releasing dump from, *Sys Mgr Man: Tuning*, 15–16
requirements
location, *Sys Mgr Man: Tuning*, 15–3, 15–5, 16–8
size for saving dumps, *Sys Mgr Man: Tuning*, 15–4
saving dump contents on reboot, *Sys Mgr Man: Tuning*, 15–2
secondary, *Sys Mgr Man: Tuning*, 15–5, 15–20
sizes
See Page file sizes
tasks for managing, *Sys Mgr Man: Tuning*, 15–1
VMScluster satellite node, *Sys Mgr Man: Essentials*, 5–6
writing crash dumps to, *Sys Mgr Man: Tuning*, 15–2

Page file size
calculation changes, *Sys Mgr Man: Tuning*, 15–9
Page file sizes
calculating
for paging, *Sys Mgr Man: Tuning*, 15–8
for saving dumps, *Sys Mgr Man: Tuning*, 15–7
manually, *Sys Mgr Man: Tuning*, 15–7, 15–8
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–5
changing
recommended method, *Sys Mgr Man: Tuning*, 15–22
when to increase size, *Sys Mgr Man: Tuning*, 15–9
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–22
with SYSGEN, *Sys Mgr Man: Tuning*, 15–26
determining current, *Sys Mgr Man: Tuning*, 15–23
displaying AUTOGEN's calculations, *Sys Mgr Man: Tuning*, 15–18, 15–22
message indicating insufficient, *Sys Mgr Man: Tuning*, 15–9
required
for paging, *Sys Mgr Man: Tuning*, 15–8
for saving dumps, *Sys Mgr Man: Tuning*, 15–4, 15–7
specifying
for individual files, *Sys Mgr Man: Tuning*, 15–22, 15–24
total for multiple files, *Sys Mgr Man: Tuning*, 15–22, 15–23
when to increase, *Sys Mgr Man: Tuning*, 15–9
Pagelets
size, *Sys Mgr Man: Essentials*, 6–39
size of, *Sys Mgr Man: Essentials*, 6–39
Page overflow
controlling, *Sys Mgr Man: Essentials*, 13–44
Pages
size, *Sys Mgr Man: Essentials*, 6–39
Page setup modules, *Sys Mgr Man: Essentials*, 13–45
See also Device control modules
specifying forms, *Sys Mgr Man: Essentials*, 13–42
Page width and length
specifying in forms, *Sys Mgr Man: Essentials*, 13–42
Paging, *Sys Mgr Man: Tuning*, 15–4
increased with vector processing, *Sys Mgr Man: Tuning*, 26–7
Paging file process limit, *Sys Mgr Man: Essentials*, 6–44
PAKs (Product Authorization Keys)
installing before using VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–8
loading in system startup, *Sys Mgr Man: Essentials*, 5–5
preventing nodes from sharing, *Sys Mgr Man: Essentials*, 3–8
registering for DECnet, *Sys Mgr Man: Tuning*, 21–12
PAN command, *Sys Mgr Man: Tuning*, 20–6
PAN keypad function, *Sys Mgr Man: Tuning*, 20–7
Paper
See Printer paper
See Stock
Paper jam
pausing printer to fix, *Sys Mgr Man: Essentials*, 13–75
Parameter files
See also System parameters
ALPHAVMSSYS.PAR (Alpha), *Sys Mgr Man: Tuning*, 14–3
initializing parameters at boot time, *Sys Mgr Man: Tuning*, 14–36
role in the boot process, *Sys Mgr Man: Essentials*, 4–2
booting with alternate, *Sys Mgr Man: Essentials*, 4–7
changing
effects of, *Sys Mgr Man: Tuning*, 14–34
with SYSGEN, *Sys Mgr Man: Tuning*, 14–33
with SYSMAN, *Sys Mgr Man: Tuning*, 14–28
common in a VMScluster environment, *Sys Mgr Man: Tuning*, 14–22
creating new
with SYSGEN, *Sys Mgr Man: Tuning*, 14–35
default, *Sys Mgr Man: Essentials*, 4–6; *Sys Mgr Man: Tuning*, 14–3
including in MODPARAMS.DAT file, *Sys Mgr Man: Tuning*, 14–22
limitation on error checking in, *Sys Mgr Man: Tuning*, 14–19
MODPARAMS.DAT, *Sys Mgr Man: Tuning*, 14–17
sample, *Sys Mgr Man: Tuning*, 14–17
multiple, with AUTOGEN, *Sys Mgr Man: Tuning*, 14–22
VAXVMSSYS.PAR (VAX), *Sys Mgr Man: Tuning*, 14–3
initializing active parameters at boot time, *Sys Mgr Man: Tuning*, 14–36

Parameter files
VAXVMSSYS.PAR (VAX) (cont'd)
role in the boot process, *Sys Mgr Man: Essentials*, 4–2
Parameters
passing to a startup command procedure, *Sys Mgr Man: Essentials*, 5–19
system
See also System parameters
See System parameters
PARAMETERS command, *Sys Mgr Man: Essentials*, 2–11
See also System parameters
in SYSMAN, *Sys Mgr Man: Tuning*, 14–25
summary, *Sys Mgr Man: Tuning*, 14–30
Partition, *Sys Mgr Man: Tuning*, 23–2
$PASSWORD card, *Sys Mgr Man: Essentials*, 7–18
Password generators
using to obtain initial password, *Sys Mgr Man: Essentials*, 11–2
Password management
forced change, *Sys Mgr Man: Essentials*, 11–4
guidelines for protecting, *Sys Mgr Man: Essentials*, 11–5
how to preexpire, *Sys Mgr Man: Essentials*, 11–3
length of passwords, *Sys Mgr Man: Essentials*, 11–4
reasons to assign system passwords, *Sys Mgr Man: Essentials*, 11–3
secondary passwords, *Sys Mgr Man: Essentials*, 11–4
setting the expiration for, *Sys Mgr Man: Essentials*, 11–4
setting up initial passwords, *Sys Mgr Man: Essentials*, 11–2
when to use dual passwords, *Sys Mgr Man: Essentials*, 11–2
Passwords
See also Password management
conditions required for SYSMAN, *Sys Mgr Man: Essentials*, 2–13
entering when logging in, *Sys Mgr Man: Essentials*, 2–10
forgotten by user, *Sys Mgr Man: Essentials*, 6–23
for VMScluster access
modifying, *Sys Mgr Man: Tuning*, 20–17
history list, *Sys Mgr Man: Essentials*, 11–2
InfoServer system, *Sys Mgr Man: Tuning*, 23–6
modifying system, *Sys Mgr Man: Essentials*, 6–9
modifying user, *Sys Mgr Man: Essentials*, 6–13
when forgotten prevents you from logging in, *Sys Mgr Man: Essentials*, 4–10
Passwords (cont'd)
with SYSMAN, *Sys Mgr Man: Essentials*, 2–16
PATHWORKS
startup restrictions, *Sys Mgr Man: Tuning*, 23–12
Paused queue status, *Sys Mgr Man: Essentials*, 13–51
Pausing an output queue, *Sys Mgr Man: Essentials*, 13–75
to align position of print for preprinted form, *Sys Mgr Man: Essentials*, 13–75, 13–76
to change position of print, *Sys Mgr Man: Essentials*, 13–75
Pausing queue status, *Sys Mgr Man: Essentials*, 13–51
PCF (product configuration file), *Sys Mgr Man: Essentials*, 3–20
command to create, *Sys Mgr Man: Essentials*, 3–23
command to input, *Sys Mgr Man: Essentials*, 3–30
command to output, *Sys Mgr Man: Essentials*, 3–30
creating before installation, *Sys Mgr Man: Essentials*, 3–22
creating during installation, *Sys Mgr Man: Essentials*, 3–30
naming, *Sys Mgr Man: Essentials*, 3–23
use existing, *Sys Mgr Man: Essentials*, 3–30
PCSI
See POLYCENTER Software Installation utility
PCSI$CONFIGURATION, *Sys Mgr Man: Essentials*, 3–23
PCSI$DESTINATION, *Sys Mgr Man: Essentials*, 3–28
PCSI$SOURCE, *Sys Mgr Man: Essentials*, 3–28
PDF (product description file), *Sys Mgr Man: Essentials*, 3–20
Pending bad block log file
BADLOG.SYS, *Sys Mgr Man: Tuning*, A–9
reserved file, *Sys Mgr Man: Tuning*, A–9
Pending jobs
requeuing, *Sys Mgr Man: Essentials*, 13–73
troubleshooting, *Sys Mgr Man: Essentials*, 13–78
Pending job status
definition, *Sys Mgr Man: Essentials*, 13–69
deleting a job in, *Sys Mgr Man: Essentials*, 13–74
determining whether a job is in, *Sys Mgr Man: Essentials*, 13–78
inducing with STOP/QUEUE/REQUEUE command, *Sys Mgr Man: Essentials*, 13–72
requeuing a job in, *Sys Mgr Man: Essentials*, 13–73

Percent sign (%)
as wildcard character
using with tape volumes, *Sys Mgr Man: Essentials*, 9–16
Performance
See also Managing performance
See also Tuning
disk, *Sys Mgr Man: Essentials*, 8–28
effect of file extension on, *Sys Mgr Man: Tuning*, 16–7
importance of correct page file size, *Sys Mgr Man: Tuning*, 15–8
importance of correct swap file size, *Sys Mgr Man: Tuning*, 15–10
importance of sufficient hardware capacity, *Sys Mgr Man: Tuning*, 16–6
improving
decompressing system libraries, *Sys Mgr Man: Tuning*, 16–6
designing efficient applications, *Sys Mgr Man: Tuning*, 16–4
disabling high-water marking, *Sys Mgr Man: Tuning*, 16–7
encouraging batch processing, *Sys Mgr Man: Tuning*, 16–3
for vector processing with batch queues, *Sys Mgr Man: Tuning*, 26–7
installing frequently used images, *Sys Mgr Man: Tuning*, 16–7, 16–9
moving page and swap files off system disk, *Sys Mgr Man: Tuning*, 15–5
options for, *Sys Mgr Man: Tuning*, 16–6
reducing system disk I/O, *Sys Mgr Man: Tuning*, 16–8
relinking images, *Sys Mgr Man: Tuning*, 16–7
restricting the number of interactive users, *Sys Mgr Man: Tuning*, 16–4
restricting user login hours, *Sys Mgr Man: Tuning*, 16–4
setting RMS file extend parameters, *Sys Mgr Man: Tuning*, 16–7
slicing shareable images, *Sys Mgr Man: Tuning*, 16–12
tuning the system, *Sys Mgr Man: Tuning*, 16–4
with AUTOGEN feedback, *Sys Mgr Man: Tuning*, 14–10
load balancing on public volumes, *Sys Mgr Man: Essentials*, 8–9
monitoring, *Sys Mgr Man: Tuning*, 16–2
tools used for, *Sys Mgr Man: Tuning*, 16–2
of vector processing, *Sys Mgr Man: Tuning*, 26–4
testing for disks, *Sys Mgr Man: Essentials*, 8–8
Performance management
using MONITOR, *Sys Mgr Man: Tuning*, 18–40
Permanent databases
network, *Sys Mgr Man: Tuning*, 21–11
PGFLQUOTA process limit, *Sys Mgr Man: Essentials*, 6–44
limiting page file usage with, *Sys Mgr Man: Tuning*, 15–10
minimum recommended value, *Sys Mgr Man: Tuning*, 15–10
value for efficient backups, *Sys Mgr Man: Essentials*, 10–10
Phase controller for UETP
See UETPHAS00.EXE file
Physical dump, *Sys Mgr Man: Tuning*, 15–3
compared to selective dump, *Sys Mgr Man: Tuning*, 15–3, 15–11
POLYCENTER Software Installation utility, *Sys Mgr Man: Essentials*, 3–18
See also PRODUCT command
commands and operations, *Sys Mgr Man: Essentials*, 3–19
creating a PCF, *Sys Mgr Man: Essentials*, 3–22
databases, *Sys Mgr Man: Essentials*, 3–20
DCL interface, *Sys Mgr Man: Essentials*, 3–18
DECwindows Motif interface, *Sys Mgr Man: Essentials*, 3–18
exiting from DECwindows Motif, *Sys Mgr Man: Essentials*, 3–35
getting help, *Sys Mgr Man: Essentials*, 3–35
installing layered products, *Sys Mgr Man: Essentials*, 3–6
installing software, *Sys Mgr Man: Essentials*, 3–27
installing the OpenVMS Alpha operating system, *Sys Mgr Man: Essentials*, 3–2
interfaces to use, *Sys Mgr Man: Essentials*, 3–18
managing installed software, *Sys Mgr Man: Essentials*, 3–34
preliminary steps, *Sys Mgr Man: Essentials*, 3–28
PRODUCT command to run, *Sys Mgr Man: Essentials*, 3–18
product configuration files, *Sys Mgr Man: Essentials*, 3–20
registering products, *Sys Mgr Man: Essentials*, 3–27
registering volume labels, *Sys Mgr Man: Essentials*, 3–33
removing installed products, *Sys Mgr Man: Essentials*, 3–34
starting the utility, *Sys Mgr Man: Essentials*, 3–18

POLYCENTER Software Installation utility (cont'd)
upgrading the OpenVMS Alpha operating system, *Sys Mgr Man: Essentials*, 3–2
using a product database, *Sys Mgr Man: Essentials*, 3–26
using to add or delete optional system files, *Sys Mgr Man: Essentials*, 5–2
Pooled resource, *Sys Mgr Man: Essentials*, 6–2
Ports
setting up LAT, *Sys Mgr Man: Essentials*, 13–14
Position of print
aligning for preprinted forms, *Sys Mgr Man: Essentials*, 13–75, 13–76
changing, *Sys Mgr Man: Essentials*, 13–75
PostScript printing, *Sys Mgr Man: Essentials*, 13–9
PRCLM process limit, *Sys Mgr Man: Essentials*, 6–44
Preventing autostart queues from starting, *Sys Mgr Man: Essentials*, 13–54
Primary bootstrap image
definition, *Sys Mgr Man: Essentials*, 4–17
role in boot process, *Sys Mgr Man: Essentials*, 4–2
PRIMARY day
defining for accounts, *Sys Mgr Man: Essentials*, 6–27
Primary page file, *Sys Mgr Man: Tuning*, 15–5
See also PAGEFILE.SYS file
See also Page files
location requirement, *Sys Mgr Man: Tuning*, 16–8
Primary processors, *Sys Mgr Man: Tuning*, 26–2
Primary swap file, *Sys Mgr Man: Tuning*, 15–5
See also SWAPFILE.SYS file
See also Swap files
PRINT command
bypassing symbiont formatting, *Sys Mgr Man: Essentials*, 13–44
overriding default form-feed options with, *Sys Mgr Man: Essentials*, 13–44
preventing users from executing, *Sys Mgr Man: Essentials*, 13–53
processing of, *Sys Mgr Man: Essentials*, 13–2
specifying a form, *Sys Mgr Man: Essentials*, 13–42
specifying banner pages, *Sys Mgr Man: Essentials*, 13–60
specifying characteristics, *Sys Mgr Man: Essentials*, 13–28
specifying job retention, *Sys Mgr Man: Essentials*, 13–73
specifying scheduling priority, *Sys Mgr Man: Essentials*, 13–72
specifying setup and page setup modules, *Sys Mgr Man: Essentials*, 13–67
Printer paper
controlling with forms, *Sys Mgr Man: Essentials*, 13–42, 13–64
pausing to align, *Sys Mgr Man: Essentials*, 13–75
sheet feed, *Sys Mgr Man: Essentials*, 13–62
specifying stock, *Sys Mgr Man: Essentials*, 13–62
specifying width, *Sys Mgr Man: Essentials*, 13–61
Printers
controlling functions of, *Sys Mgr Man: Essentials*, 13–44
LAT
See LAT software, printers
managing
tasks for, *Sys Mgr Man: Essentials*, 7–13
setting characteristics, *Sys Mgr Man: Essentials*, 7–13, 13–14
in system startup, *Sys Mgr Man: Essentials*, 5–12, 7–14
setting up before creating queues, *Sys Mgr Man: Essentials*, 13–14
spooled, *Sys Mgr Man: Essentials*, 13–14
definition, *Sys Mgr Man: Essentials*, 7–14
despooling, *Sys Mgr Man: Essentials*, 7–16
recommended use, *Sys Mgr Man: Essentials*, 7–14
requirement, *Sys Mgr Man: Essentials*, 7–14
sample configuration, *Sys Mgr Man: Essentials*, 13–13
spooling, *Sys Mgr Man: Essentials*, 7–15
testing spooling of, *Sys Mgr Man: Essentials*, 7–16
troubleshooting, *Sys Mgr Man: Essentials*, 13–77
Print forms
See Forms
Printing
distributed, *Sys Mgr Man: Essentials*, 13–13
from applications using spooled printers, *Sys Mgr Man: Essentials*, 7–14
job status, *Sys Mgr Man: Essentials*, 13–69
PostScript, *Sys Mgr Man: Essentials*, 13–9
remotely, *Sys Mgr Man: Essentials*, 13–13
resuming at a specified position, *Sys Mgr Man: Essentials*, 13–75, 13–76
Print jobs
See also Batch jobs
See also Output jobs
See also Print queues
See Output jobs
Print queues
See also Batch queues
See also Output queues

Print queues (cont'd)
See Output queues
Print symbionts
See Symbionts
Priority, *Sys Mgr Man: Essentials*, 6–2, 6–31
base, *Sys Mgr Man: Essentials*, 6–2, 6–31
choosing for a batch queue, *Sys Mgr Man: Essentials*, 13–30
effect of changing, *Sys Mgr Man: Essentials*, 13–6
specifying for a batch queue, *Sys Mgr Man: Essentials*, 13–29
specifying for a queue, *Sys Mgr Man: Essentials*, 13–19
job scheduling, *Sys Mgr Man: Essentials*, 13–72
See also Job scheduling
changing for a job, *Sys Mgr Man: Essentials*, 13–72, 13–73
display on banner pages, *Sys Mgr Man: Essentials*, 13–35, 13–37, 13–40, 13–41
specifying for a job, *Sys Mgr Man: Essentials*, 13–70, 13–72
Private volumes, *Sys Mgr Man: Essentials*, 8–9
Privileged images, *Sys Mgr Man: Tuning*, 16–10, 16–14
Privilege mode
recommended for logical names of system components, *Sys Mgr Man: Essentials*, 5–9
Privileges
all, *Sys Mgr Man: Essentials*, 6–4
allowing nonprivileged users to run programs that require, *Sys Mgr Man: Tuning*, 16–14
changing in SYSMAN, *Sys Mgr Man: Essentials*, 2–17
enhancement for installed files, *Sys Mgr Man: Tuning*, 16–13
files, *Sys Mgr Man: Essentials*, 6–4
for SYSTEM account, *Sys Mgr Man: Essentials*, 2–9
GRPPRV, *Sys Mgr Man: Essentials*, 11–8
process, *Sys Mgr Man: Essentials*, 6–3
required for UETP, *Sys Mgr Man: Tuning*, 17–23
required to mount volumes, *Sys Mgr Man: Essentials*, 8–15
SECURITY
to mount volume with protected subsystems, *Sys Mgr Man: Essentials*, 8–15
SYSNAM, *Sys Mgr Man: Essentials*, 8–16
SYSPRV, *Sys Mgr Man: Essentials*, 8–16
giving rights of system user, *Sys Mgr Man: Essentials*, 11–8
VOLPRO, *Sys Mgr Man: Essentials*, 8–14
to mount volume as foreign, *Sys Mgr Man: Essentials*, 8–15
Problems
See also Troubleshooting
booting
fixing by booting with default parameter values, *Sys Mgr Man: Essentials*, 4–8
fixing by booting with minimum startup, *Sys Mgr Man: Essentials*, 4–14
hardware, *Sys Mgr Man: Essentials*, 4–16
invalid boot block, *Sys Mgr Man: Essentials*, 4–17
devices not recognized by the system, *Sys Mgr Man: Essentials*, 7–7
forgotten password
fixing by booting without the UAF, *Sys Mgr Man: Essentials*, 4–10
logging in, *Sys Mgr Man: Essentials*, 4–8, 4–10
OPCOM failure, *Sys Mgr Man: Essentials*, 2–22
Processes
maintaining when disconnecting a terminal, *Sys Mgr Man: Essentials*, 7–11
priority, *Sys Mgr Man: Essentials*, 6–31
See also Priority, base
See also Priority, job scheduling
Processing environments
multiprocessing
See Multiprocessing
vector processing
See Vector processing
Processing job status, *Sys Mgr Man: Essentials*, 13–69
Process limits, *Sys Mgr Man: Essentials*, 6–2
account jobs, *Sys Mgr Man: Essentials*, 6–43
adjusting for vector processing, *Sys Mgr Man: Tuning*, 26–7
AST queue, *Sys Mgr Man: Essentials*, 6–42
CPU default
specifying a value for batch queues, *Sys Mgr Man: Essentials*, 13–19, 13–29, 13–31
CPU maximum
specifying a value for batch queues, *Sys Mgr Man: Essentials*, 13–19, 13–29, 13–31
CPU time, *Sys Mgr Man: Essentials*, 6–42
detached process, *Sys Mgr Man: Essentials*, 6–43
direct I/O count, *Sys Mgr Man: Essentials*, 6–42
enqueue quota, *Sys Mgr Man: Essentials*, 6–43
expiration, *Sys Mgr Man: Essentials*, 6–43
for Snapshot facility, *Sys Mgr Man: Essentials*, 4–20
jobwide logical name table, *Sys Mgr Man: Essentials*, 6–43
open file, *Sys Mgr Man: Essentials*, 6–43
paging file, *Sys Mgr Man: Essentials*, 6–44

Process limits (cont'd)
process jobs, *Sys Mgr Man: Essentials*, 6–43
setting, *Sys Mgr Man: Essentials*, 6–40
subprocess creation, *Sys Mgr Man: Essentials*, 6–44
system resources, *Sys Mgr Man: Essentials*, 6–40
timer queue entry, *Sys Mgr Man: Essentials*, 6–44
working set
quota, *Sys Mgr Man: Essentials*, 6–44
Processors
adding to a multiprocessing active set, *Sys Mgr Man: Tuning*, 26–3
removing from a multiprocessing active set, *Sys Mgr Man: Tuning*, 26–3
Process quotas
see Process limits
recommended values for backups, *Sys Mgr Man: Essentials*, 10–10
setting before a backup, *Sys Mgr Man: Essentials*, 10–9
working set
extent, *Sys Mgr Man: Essentials*, 6–44
Product Authorization Keys
See PAKs
PRODUCT command
See POLYCENTER Software Installation Utility
DCL interface syntax, *Sys Mgr Man: Essentials*, 3–18
DECwindows Motif interface syntax, *Sys Mgr Man: Essentials*, 3–18
Product configuration file
See PCF
Product database, *Sys Mgr Man: Essentials*, 3–35
definition of, *Sys Mgr Man: Essentials*, 3–20
how to use, *Sys Mgr Man: Essentials*, 3–26
noncompliant products, *Sys Mgr Man: Essentials*, 3–27
removing products from, *Sys Mgr Man: Essentials*, 3–34
retrieving product information from, *Sys Mgr Man: Essentials*, 3–33
tracking software dependencies, *Sys Mgr Man: Essentials*, 3–27
volume label change, *Sys Mgr Man: Essentials*, 3–33
Product dependencies, *Sys Mgr Man: Essentials*, 3–27
Product description file
See PDF
PRODUCT INSTALL command, *Sys Mgr Man: Essentials*, 3–27
Product list
obtaining, *Sys Mgr Man: Essentials*, 3–10
VMSINSTAL.COM parameter, *Sys Mgr Man: Essentials*, 3–10
Product listing, *Sys Mgr Man: Essentials*, 3–35
Profiles
in MAIL, *Sys Mgr Man: Essentials*, 6–38
in SYSMAN, *Sys Mgr Man: Essentials*, 2–16
changing default directory, *Sys Mgr Man: Essentials*, 2–17
changing privileges, *Sys Mgr Man: Essentials*, 2–17
Protected images, *Sys Mgr Man: Tuning*, 16–11, 16–14, 16–15
Protected subsystems, *Sys Mgr Man: Essentials*, 6–16
enabling, *Sys Mgr Man: Essentials*, 8–28
mounting volumes with, *Sys Mgr Man: Essentials*, 8–15
Protection
See also Protection codes
See also Security
ACL-based, *Sys Mgr Man: Essentials*, 6–16, 6–33, 11–9
applying to public disk volumes, *Sys Mgr Man: Essentials*, 8–15
applying to queues, *Sys Mgr Man: Essentials*, 13–22 to 13–26
applying to system dump file, *Sys Mgr Man: Tuning*, 15–4
assigning code when mounting a volume, *Sys Mgr Man: Essentials*, 8–23
changing, *Sys Mgr Man: Essentials*, 8–16
changing with PROTECTION qualifier, *Sys Mgr Man: Essentials*, 8–18
default, *Sys Mgr Man: Essentials*, 9–7
changing, *Sys Mgr Man: Essentials*, 9–8
directory
changing with SET SECURITY /PROTECTION command, *Sys Mgr Man: Essentials*, 9–12
specifying with CREATE/DIRECTORY command, *Sys Mgr Man: Essentials*, 9–12
specifying with /PROTECTION qualifier, *Sys Mgr Man: Essentials*, 9–12
disk file, *Sys Mgr Man: Essentials*, 9–6
disk volumes, *Sys Mgr Man: Essentials*, 8–16
displaying, *Sys Mgr Man: Essentials*, 9–7
file, *Sys Mgr Man: Essentials*, 9–4
default, *Sys Mgr Man: Essentials*, 9–6, 9–7
directory, *Sys Mgr Man: Essentials*, 9–4, 9–10
disk, *Sys Mgr Man: Essentials*, 9–4
for public disk volumes, *Sys Mgr Man: Essentials*, 8–15
for system dump file, *Sys Mgr Man: Tuning*, 15–4
ISO 9660-formatted media, *Sys Mgr Man: Essentials*, 8–17
magnetic tape, *Sys Mgr Man: Essentials*, 9–12

Protection (cont'd)
for interchange environments, *Sys Mgr Man: Essentials*, 8–20
format for object, *Sys Mgr Man: Essentials*, 11–8
mask, *Sys Mgr Man: Essentials*, 8–17
mounting a volume with protected subsystems, *Sys Mgr Man: Essentials*, 8–28
of devices, *Sys Mgr Man: Essentials*, 7–5
overriding, *Sys Mgr Man: Essentials*, 8–25
qualifier for SET VOLUME command, *Sys Mgr Man: Essentials*, 8–18
UIC-based codes, *Sys Mgr Man: Essentials*, 6–16
volumes, *Sys Mgr Man: Essentials*, 9–4
disk, *Sys Mgr Man: Essentials*, 8–14, 8–15
ISO 9660-formatted media, *Sys Mgr Man: Essentials*, 8–17
standard-labeled, *Sys Mgr Man: Essentials*, 8–19
tape, *Sys Mgr Man: Essentials*, 8–15, 8–19, 9–22
VOLPRO privilege, *Sys Mgr Man: Essentials*, 8–14
volume set
ISO 9660-formatted media, *Sys Mgr Man: Essentials*, 8–17
Protection checks
MOUNT command
overriding, *Sys Mgr Man: Essentials*, 8–23
Protection codes
access types, *Sys Mgr Man: Essentials*, 11–8
changing, *Sys Mgr Man: Essentials*, 9–8
format, *Sys Mgr Man: Essentials*, 11–8
null access specification, *Sys Mgr Man: Essentials*, 11–8
specifying, *Sys Mgr Man: Essentials*, 9–6
Protocols
for using TCP/IP, *Sys Mgr Man: Tuning*, 21–13
FTP, *Sys Mgr Man: Tuning*, 21–14
LASTport, *Sys Mgr Man: Tuning*, 23–5
LASTport/Disk, *Sys Mgr Man: Tuning*, 23–5
LASTport/Tape, *Sys Mgr Man: Tuning*, 23–6
RCP, *Sys Mgr Man: Tuning*, 21–14
RLOGIN, *Sys Mgr Man: Tuning*, 21–13
TELNET, *Sys Mgr Man: Tuning*, 21–13
virtual terminal, *Sys Mgr Man: Tuning*, 21–13
Proxy accounts
adding, *Sys Mgr Man: Essentials*, 6–36
Proxy authorization files
NET$PROXY.DAT, *Sys Mgr Man: Essentials*, 6–4, 6–35
NETPROXY.DAT, *Sys Mgr Man: Essentials*, 6–4, 6–35
Proxy database
logical name defining location, *Sys Mgr Man: Essentials*, 5–8
Proxy database (cont'd)
managed by security server, *Sys Mgr Man: Essentials*, 5–5
Proxy logins
controlling system use, *Sys Mgr Man: Essentials*, 6–38
PRTSMB symbiont, *Sys Mgr Man: Essentials*, 13–3
on LAT printers, *Sys Mgr Man: Essentials*, 13–78
PTF (product text file), *Sys Mgr Man: Essentials*, 3–20
Public volumes
access to, *Sys Mgr Man: Essentials*, 8–8
conditions for using, *Sys Mgr Man: Essentials*, 8–8
creating with SYSTEM qualifier, *Sys Mgr Man: Essentials*, 8–24
definition, *Sys Mgr Man: Essentials*, 8–8
initializing, *Sys Mgr Man: Essentials*, 8–12
guidelines, *Sys Mgr Man: Essentials*, 8–13
load balancing, *Sys Mgr Man: Essentials*, 8–9
mounting, *Sys Mgr Man: Essentials*, 8–20, 8–21
in system startup, *Sys Mgr Man: Essentials*, 5–11
mounting volume sets, *Sys Mgr Man: Essentials*, 8–30
on large configurations, *Sys Mgr Man: Essentials*, 8–8
on small configurations, *Sys Mgr Man: Essentials*, 8–8
planning, *Sys Mgr Man: Essentials*, 8–8
protecting, *Sys Mgr Man: Essentials*, 8–15
setting protection, *Sys Mgr Man: Essentials*, 8–18
testing disk performance, *Sys Mgr Man: Essentials*, 8–8
PURGE command
NCP, *Sys Mgr Man: Tuning*, 21–11
saving disk space, *Sys Mgr Man: Essentials*, 8–53

Q

QMAN$MASTER.DAT file
See Master file of queue database
QMAN$MASTER logical name, *Sys Mgr Man: Essentials*, 12–6
defining during system startup, *Sys Mgr Man: Essentials*, 5–8
defining to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
QUANTUM system parameter, *Sys Mgr Man: Tuning*, 26–8
Queue characteristics
canceling, *Sys Mgr Man: Essentials*, 13–59

Queue characteristics (cont'd)
commands for managing, *Sys Mgr Man: Essentials*, 13–57
defining, *Sys Mgr Man: Essentials*, 13–58
definition, *Sys Mgr Man: Essentials*, 13–28
deleting, *Sys Mgr Man: Essentials*, 13–59
obtaining information about, *Sys Mgr Man: Essentials*, 13–51, 13–59
problems
deleting, *Sys Mgr Man: Essentials*, 13–81
mismatch, *Sys Mgr Man: Essentials*, 13–80
sample use of, *Sys Mgr Man: Essentials*, 13–28
specifying
on a job, *Sys Mgr Man: Essentials*, 13–28
on a queue, *Sys Mgr Man: Essentials*, 13–19, 13–28, 13–59
storage of in queue database, *Sys Mgr Man: Essentials*, 13–17
Queue commands
affected by multiple queue managers, *Sys Mgr Man: Essentials*, 12–4
creating a queue database, *Sys Mgr Man: Essentials*, 12–7
creating queues, *Sys Mgr Man: Essentials*, 13–48
deleting queues, *Sys Mgr Man: Essentials*, 13–57
displaying information about the queue manager, *Sys Mgr Man: Essentials*, 12–11
displaying jobs, *Sys Mgr Man: Essentials*, 13–68
displaying queues, *Sys Mgr Man: Essentials*, 13–49
enabling autostart, *Sys Mgr Man: Essentials*, 13–16, 13–18, 13–48
managing banner pages, *Sys Mgr Man: Essentials*, 13–60
managing characteristics, *Sys Mgr Man: Essentials*, 13–57
managing device control libraries, *Sys Mgr Man: Essentials*, 13–65
managing forms and stock, *Sys Mgr Man: Essentials*, 13–61
managing queues, *Sys Mgr Man: Essentials*, 13–46
modifying jobs, *Sys Mgr Man: Essentials*, 13–70
modifying queues, *Sys Mgr Man: Essentials*, 13–53
pausing queues, *Sys Mgr Man: Essentials*, 13–53
relationship between starting and enabling autostart, *Sys Mgr Man: Essentials*, 13–4
setting UIC-based protection, *Sys Mgr Man: Essentials*, 13–23
showing UIC-based protection, *Sys Mgr Man: Essentials*, 13–23
Queue commands (cont'd)
specifying options, *Sys Mgr Man: Essentials*, 13–19, 13–31
starting queues
autostart, *Sys Mgr Man: Essentials*, 13–16, 13–18
nonautostart, *Sys Mgr Man: Essentials*, 13–17, 13–48
starting the queue manager, *Sys Mgr Man: Essentials*, 12–7
caution, *Sys Mgr Man: Essentials*, 12–8
creating an additional queue manager, *Sys Mgr Man: Essentials*, 12–10
restarting, *Sys Mgr Man: Essentials*, 12–9
specifying failover list, *Sys Mgr Man: Essentials*, 12–9
specifying name of queue manager, *Sys Mgr Man: Essentials*, 12–10
specifying nodes to run the queue manager, *Sys Mgr Man: Essentials*, 12–6
stopping
all queues on a node, *Sys Mgr Man: Essentials*, 12–9
queues, *Sys Mgr Man: Essentials*, 13–49, 13–54
the queue manager, *Sys Mgr Man: Essentials*, 12–9
Queue configurations
sample batch queuing system, *Sys Mgr Man: Essentials*, 13–4 to 13–7
sample output queuing environment, *Sys Mgr Man: Essentials*, 13–8 to 13–14
Queue database
See also Journal file of queue database
See also Master file of queue database
See also Queue file of queue database
closing, *Sys Mgr Man: Essentials*, 12–9
creating, *Sys Mgr Man: Essentials*, 12–7
default location, *Sys Mgr Man: Essentials*, 12–5
definition, *Sys Mgr Man: Essentials*, 12–4
detection of queue corruption, *Sys Mgr Man: Essentials*, 13–84
determining location, *Sys Mgr Man: Essentials*, 12–7
determining location of master file, *Sys Mgr Man: Essentials*, 12–5
determining location of queue and journal file, *Sys Mgr Man: Essentials*, 12–6
files comprising, *Sys Mgr Man: Essentials*, 12–4
for multiple queue managers, *Sys Mgr Man: Essentials*, 12–4, 12–5
naming convention, *Sys Mgr Man: Essentials*, 12–5
function, *Sys Mgr Man: Essentials*, 12–4

Queue database (cont'd)
logical name defining location, *Sys Mgr Man: Essentials*, 5–8
managing, *Sys Mgr Man: Essentials*, 12–1
mounting the disk holding, *Sys Mgr Man: Essentials*, 12–6
moving, *Sys Mgr Man: Essentials*, 12–5
moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
requirement in a VMScluster environment, *Sys Mgr Man: Essentials*, 12–5, 12–6
restoring a damaged, *Sys Mgr Man: Essentials*, 12–12
saving, *Sys Mgr Man: Essentials*, 12–11
specifying location, *Sys Mgr Man: Essentials*, 12–5
Queue failover, *Sys Mgr Man: Essentials*, 13–4
Queue file of queue database, *Sys Mgr Man: Essentials*, 12–4
See also Queue database
location, *Sys Mgr Man: Essentials*, 12–6, 12–7
changing, *Sys Mgr Man: Essentials*, 12–6
saving, *Sys Mgr Man: Essentials*, 12–12
Queue files
detection of corruption, *Sys Mgr Man: Essentials*, 13–84
Queue forms
See Forms
Queue managers
automatic restart, *Sys Mgr Man: Essentials*, 12–3, 12–7, 12–9, 12–17
availability of, *Sys Mgr Man: Essentials*, 12–9, 12–17
communication with job controller, *Sys Mgr Man: Essentials*, 12–3
creating an additional, *Sys Mgr Man: Essentials*, 12–10
default name, *Sys Mgr Man: Essentials*, 12–4
definition, *Sys Mgr Man: Essentials*, 12–1
displaying information about, *Sys Mgr Man: Essentials*, 12–11
failover, *Sys Mgr Man: Essentials*, 12–3, 12–17
forcing, *Sys Mgr Man: Essentials*, 12–9
list of nodes for, *Sys Mgr Man: Essentials*, 12–9, 12–17
function, *Sys Mgr Man: Essentials*, 12–1
improving performance, *Sys Mgr Man: Essentials*, 12–13
limiting nodes that can run, *Sys Mgr Man: Essentials*, 12–9
managing, *Sys Mgr Man: Essentials*, 12–1
moving to another node in a VMScluster system, *Sys Mgr Man: Essentials*, 12–10
multiple, *Sys Mgr Man: Essentials*, 12–3, 12–10
commands affected by, *Sys Mgr Man: Essentials*, 12–4
Queue managers
multiple (cont'd)
managing queues with, *Sys Mgr Man: Essentials*, 12–10
moving a queue to another queue manager, *Sys Mgr Man: Essentials*, 12–10
restriction, *Sys Mgr Man: Essentials*, 12–3
specifying queue manager name, *Sys Mgr Man: Essentials*, 12–4, 12–10
naming, *Sys Mgr Man: Essentials*, 12–10
relationship to queues, *Sys Mgr Man: Essentials*, 12–1
restarting, *Sys Mgr Man: Essentials*, 12–9
role in queuing process, *Sys Mgr Man: Essentials*, 12–2, 13–2
print jobs, *Sys Mgr Man: Essentials*, 13–3
role in starting active autostart queues, *Sys Mgr Man: Essentials*, 13–4
specifying name of, *Sys Mgr Man: Essentials*, 12–10
specifying preferred order in which nodes run, *Sys Mgr Man: Essentials*, 12–9
starting initially, *Sys Mgr Man: Essentials*, 12–7
stopping, *Sys Mgr Man: Essentials*, 12–9
troubleshooting, *Sys Mgr Man: Essentials*, 12–14
Queue master file
logical name defining location, *Sys Mgr Man: Essentials*, 5–8
Queue names
default for batch queue, *Sys Mgr Man: Essentials*, 13–5
default for print queue, *Sys Mgr Man: Essentials*, 13–8
defining, *Sys Mgr Man: Essentials*, 13–15
Queue options, *Sys Mgr Man: Essentials*, 13–18
banner pages, *Sys Mgr Man: Essentials*, 13–33
characteristics, *Sys Mgr Man: Essentials*, 13–28
controlling job performance and resources, *Sys Mgr Man: Essentials*, 13–28
controlling page and line overflow, *Sys Mgr Man: Essentials*, 13–43
device control libraries, *Sys Mgr Man: Essentials*, 13–44
forms, *Sys Mgr Man: Essentials*, 13–42
qualifiers for specifying, *Sys Mgr Man: Essentials*, 13–19 to 13–22
restricting access, *Sys Mgr Man: Essentials*, 13–22
retaining jobs, *Sys Mgr Man: Essentials*, 13–26
suppressing form feed, *Sys Mgr Man: Essentials*, 13–44
Queues
See also Generic queues
activating autostart, *Sys Mgr Man: Essentials*, 13–15, 13–16, 13–49

Queues (cont'd)
allowing jobs to complete before shutdown, *Sys Mgr Man: Essentials*, 13–55
assigning a default form, *Sys Mgr Man: Essentials*, 13–63
assigning device control libraries, *Sys Mgr Man: Essentials*, 13–66
autostart, *Sys Mgr Man: Essentials*, 13–4
See also Autostart feature
See also Autostart queues
activating, *Sys Mgr Man: Essentials*, 13–4
on LAT queues, *Sys Mgr Man: Essentials*, 13–4, 13–10
starting, *Sys Mgr Man: Essentials*, 13–4
availability of, *Sys Mgr Man: Essentials*, 13–4, 13–15
batch
setting up for vector processing, *Sys Mgr Man: Tuning*, 26–7
batch execution
See Batch queues
canceling characteristics, *Sys Mgr Man: Essentials*, 13–59
changing DEFAULT form, *Sys Mgr Man: Essentials*, 13–63
changing options on, *Sys Mgr Man: Essentials*, 13–53
characteristics, *Sys Mgr Man: Essentials*, 13–28
closing, *Sys Mgr Man: Essentials*, 13–53
commands
See Queue commands
creating, *Sys Mgr Man: Essentials*, 13–15, 13–48
autostart execution, *Sys Mgr Man: Essentials*, 13–15, 13–16
generic, *Sys Mgr Man: Essentials*, 13–17
nonautostart execution, *Sys Mgr Man: Essentials*, 13–16
defining a characteristic, *Sys Mgr Man: Essentials*, 13–58
defining a form, *Sys Mgr Man: Essentials*, 13–61
deleting, *Sys Mgr Man: Essentials*, 13–57
deleting a job in, *Sys Mgr Man: Essentials*, 13–74
disabled, *Sys Mgr Man: Essentials*, 13–84
displaying information about, *Sys Mgr Man: Essentials*, 13–49
failover of, *Sys Mgr Man: Essentials*, 13–15
forms, *Sys Mgr Man: Essentials*, 13–42
gathering information with F$GETQUI, *Sys Mgr Man: Essentials*, 13–51
generic batch, *Sys Mgr Man: Essentials*, 13–7
See also Generic queues
generic output, *Sys Mgr Man: Essentials*, 13–11

Queues
generic output (cont'd)
See also Generic queues
holding and releasing jobs, *Sys Mgr Man: Essentials*, 13–70
in a VMScluster environment, *Sys Mgr Man: Tuning*, 20–2
initializing, *Sys Mgr Man: Essentials*, 13–48
managing with multiple queue managers, *Sys Mgr Man: Essentials*, 12–10
merging, *Sys Mgr Man: Essentials*, 13–56
modifying, *Sys Mgr Man: Essentials*, 13–53
monitoring, *Sys Mgr Man: Essentials*, 13–49
mounting a form on, *Sys Mgr Man: Essentials*, 13–42, 13–64
moving jobs from one queue to another, *Sys Mgr Man: Essentials*, 13–57
moving to another queue manager, *Sys Mgr Man: Essentials*, 12–10
on standalone workstations
batch, *Sys Mgr Man: Essentials*, 13–5
output, *Sys Mgr Man: Essentials*, 13–8
output execution
See Execution queues
See Output queues
pausing, *Sys Mgr Man: Essentials*, 13–53
print
See Output queues
problems deleting, *Sys Mgr Man: Essentials*, 13–81
protection, *Sys Mgr Man: Essentials*, 13–22 to 13–26
reinitializing existing, *Sys Mgr Man: Essentials*, 13–53
requeuing
executing job, *Sys Mgr Man: Essentials*, 13–72
pending job, *Sys Mgr Man: Essentials*, 13–73
simplifying startup, *Sys Mgr Man: Essentials*, 13–4
spooling printers, *Sys Mgr Man: Essentials*, 7–15, 13–14
stalled
status of, *Sys Mgr Man: Essentials*, 13–79, 13–81
troubleshooting, *Sys Mgr Man: Essentials*, 13–81
starting
autostart, *Sys Mgr Man: Essentials*, 13–48
in startup command procedure, *Sys Mgr Man: Essentials*, 13–18
in system startup, *Sys Mgr Man: Essentials*, 5–12
nonautostart, *Sys Mgr Man: Essentials*, 13–16, 13–48

Queues (cont'd)
startup command procedure, *Sys Mgr Man: Essentials*, 5–12
sample, *Sys Mgr Man: Essentials*, 13–18
stopped
status of, *Sys Mgr Man: Essentials*, 13–79
stopping, *Sys Mgr Man: Essentials*, 13–54
abruptly, *Sys Mgr Man: Essentials*, 13–54
all on a node, *Sys Mgr Man: Essentials*, 13–55
before shutdown, *Sys Mgr Man: Essentials*, 13–55
smoothly, *Sys Mgr Man: Essentials*, 13–54
types of, *Sys Mgr Man: Essentials*, 13–2
output execution, *Sys Mgr Man: Essentials*, 13–3
Queue status, *Sys Mgr Man: Essentials*, 13–49, 13–50
determining, *Sys Mgr Man: Essentials*, 13–77
Queuing system
See Batch and print queuing system
Quorum
using IPC to adjust, *Sys Mgr Man: Essentials*, 8–63
QUOTA.SYS file
See Quota file
Quota file
Analyze/Disk_Structure utility creates version, *Sys Mgr Man: Tuning*, A–9
contents, *Sys Mgr Man: Essentials*, 8–49
creating, *Sys Mgr Man: Essentials*, 8–51
deleting, *Sys Mgr Man: Essentials*, 8–53
disabling, *Sys Mgr Man: Essentials*, 8–53
QUOTA.SYS, *Sys Mgr Man: Tuning*, A–9
requirements for, *Sys Mgr Man: Essentials*, 8–51
reserved file, *Sys Mgr Man: Tuning*, A–9
UIC [0,0] in, *Sys Mgr Man: Essentials*, 8–49
updating, *Sys Mgr Man: Essentials*, 8–53
Quotas
See also Process limits
See also UAFs (user authorization files), resource limits
disk
See Disk quotas
required to run UETP, *Sys Mgr Man: Tuning*, 17–23

R

RAID (redundant arrays of independent disks)
volume shadowing support, *Sys Mgr Man: Essentials*, 10–40
RCP, *Sys Mgr Man: Tuning*, 21–14
Read access
See also Access types, read
Read access (cont'd)
for disk directory files, *Sys Mgr Man: Essentials*, 9–11
for disk files, *Sys Mgr Man: Essentials*, 9–6
granting through protection codes, *Sys Mgr Man: Essentials*, 11–8
Read error
if returned when booting, *Sys Mgr Man: Essentials*, 4–16
Read operation
See Access types, read
Read/write disk
partitions, *Sys Mgr Man: Tuning*, 23–2
Real-time priority, *Sys Mgr Man: Essentials*, 6–31
Rebuilding volumes, *Sys Mgr Man: Essentials*, 7–4
Reconfiguration of software installation options, *Sys Mgr Man: Essentials*, 3–32
Records
blocking on tapes, *Sys Mgr Man: Essentials*, 8–6
size, *Sys Mgr Man: Essentials*, 8–26
Recovering lost files, *Sys Mgr Man: Essentials*, 8–57
Reducing I/O on system disk, *Sys Mgr Man: Tuning*, 16–8
Redundant arrays of independent disks
See RAID
Registering images with system version dependencies, *Sys Mgr Man: Essentials*, 5–22
Reinitializing volumes, *Sys Mgr Man: Essentials*, 8–45
Release notes, extracting from software products, *Sys Mgr Man: Essentials*, 3–29
release notes on software products, *Sys Mgr Man: Essentials*, 3–29
Release Notes option
VMSINSTAL.COM, *Sys Mgr Man: Essentials*, 3–17
Releasing a job, *Sys Mgr Man: Essentials*, 13–70
Remote directory listings, *Sys Mgr Man: Tuning*, 21–14
Remote file access, *Sys Mgr Man: Tuning*, 21–14
Remote identifiers, *Sys Mgr Man: Essentials*, 11–11
Remote InfoServer device
BIND command, *Sys Mgr Man: Tuning*, 23–14
making available, *Sys Mgr Man: Tuning*, 23–14
mounting during system startup, *Sys Mgr Man: Tuning*, 23–14
Remote log (archive) file, *Sys Mgr Man: Tuning*, 18–27
Remote login, *Sys Mgr Man: Tuning*, 21–13

Remote monitoring
limitation, *Sys Mgr Man: Tuning*, 18–44
mixed-version VMScluster systems, *Sys Mgr Man: Tuning*, 18–44
Remote node database
logical name defining location, *Sys Mgr Man: Essentials*, 5–8
Remote nodes
definition, *Sys Mgr Man: Essentials*, 2–13
Remote printing, *Sys Mgr Man: Essentials*, 13–13
Remote queue status, *Sys Mgr Man: Essentials*, 13–51
Remote terminals, *Sys Mgr Man: Essentials*, 7–12
Remote terminal service, *Sys Mgr Man: Tuning*, 21–13
Removing installed software products, *Sys Mgr Man: Essentials*, 3–34
Repairing disk structure errors, *Sys Mgr Man: Essentials*, 8–56
REPLY command
canceling a user request, *Sys Mgr Man: Essentials*, 8–27; *Sys Mgr Man: Tuning*, 18–19
closing current operator log file, *Sys Mgr Man: Tuning*, 18–17
disabling operator terminals, *Sys Mgr Man: Essentials*, 2–24; *Sys Mgr Man: Tuning*, 18–18
enabling operator terminals, *Sys Mgr Man: Essentials*, 2–23; *Sys Mgr Man: Tuning*, 18–17
specifying terminal, *Sys Mgr Man: Tuning*, 18–18
ensuring that correct volume is mounted, *Sys Mgr Man: Essentials*, 8–41
initializing tape, *Sys Mgr Man: Essentials*, 8–42
linking continuation volume to volume set, *Sys Mgr Man: Essentials*, 8–41
opening a new operator log file, *Sys Mgr Man: Tuning*, 18–17, 18–22
putting request in wait state, *Sys Mgr Man: Essentials*, 8–27
replying to requests, *Sys Mgr Man: Essentials*, 2–25, 8–27
response not recorded in operator log file, *Sys Mgr Man: Tuning*, 18–19
sending messages to users, *Sys Mgr Man: Essentials*, 2–22
REPLY/ENABLE=SECURITY command
enabling security operator terminals, *Sys Mgr Man: Tuning*, 18–29
Reporting errors
disk structure, *Sys Mgr Man: Essentials*, 8–56
Reporting on product dependencies, *Sys Mgr Man: Essentials*, 3–27
Reports
AUTOGEN, *Sys Mgr Man: Tuning*, 14–22
SHOW CLUSTER
adding classes of data, *Sys Mgr Man: Tuning*, 20–4
adding data, *Sys Mgr Man: Tuning*, 20–8
adding fields of data, *Sys Mgr Man: Tuning*, 20–4
changing default at startup, *Sys Mgr Man: Tuning*, 20–14
command to modify, *Sys Mgr Man: Tuning*, 20–10
compressing reports, *Sys Mgr Man: Tuning*, 20–11
controlling displays, *Sys Mgr Man: Tuning*, 20–5, 20–10
controlling with command procedures, *Sys Mgr Man: Tuning*, 20–5, 20–16
moving display, *Sys Mgr Man: Tuning*, 20–12
organization of, *Sys Mgr Man: Tuning*, 20–4
panning, *Sys Mgr Man: Tuning*, 20–6
scrolling, *Sys Mgr Man: Tuning*, 20–14
REQUEST command
recording request in operator log file, *Sys Mgr Man: Tuning*, 18–19
sending requests to an operator, *Sys Mgr Man: Essentials*, 2–25; *Sys Mgr Man: Tuning*, 18–19
Request identification number
in operator log file, *Sys Mgr Man: Tuning*, 18–19
Requeuing
executing job, *Sys Mgr Man: Essentials*, 13–72
pending job, *Sys Mgr Man: Essentials*, 13–73
Reserved files, *Sys Mgr Man: Tuning*, A–4
backup log file (BACKUP.SYS), *Sys Mgr Man: Tuning*, A–9
bad block file (BADBLK.SYS), *Sys Mgr Man: Tuning*, A–8
continuation file (CONTIN.SYS), *Sys Mgr Man: Tuning*, A–9
index file (INDEXF.SYS), *Sys Mgr Man: Tuning*, A–5
list of, *Sys Mgr Man: Essentials*, 8–4
master file directory (MFD), *Sys Mgr Man: Tuning*, A–8
pending bad block log file (BADLOG.SYS), *Sys Mgr Man: Tuning*, A–9
quota file (QUOTA.SYS), *Sys Mgr Man: Tuning*, A–9
storage bit map file (BITMAP.SYS), *Sys Mgr Man: Tuning*, A–8
volume security profile (SECURITY.SYS), *Sys Mgr Man: Tuning*, A–9
volume set list file (VOLSET.SYS), *Sys Mgr Man: Tuning*, A–9

Reset button on POLYCENTER Software Installation utility, *Sys Mgr Man: Essentials*, 3–40
Reset modules, *Sys Mgr Man: Essentials*, 13–45
See also Device control modules
Resetting saved information, *Sys Mgr Man: Essentials*, 3–40
Resident images
installing (Alpha), *Sys Mgr Man: Tuning*, 16–11, 16–12, 16–13
in system startup, *Sys Mgr Man: Essentials*, 5–13
Resource accounting, *Sys Mgr Man: Tuning*, 19–1
Resource limits
See Process limits
Resource use
producing reports of, *Sys Mgr Man: Tuning*, 19–4
Restarting OPCOM (operator communication manager), *Sys Mgr Man: Essentials*, 2–22
Restarting the queue manager, *Sys Mgr Man: Essentials*, 12–9
Restore operations
with BACKUP, *Sys Mgr Man: Essentials*, 10–28
Restoring directories, *Sys Mgr Man: Essentials*, 10–29
Restoring files, *Sys Mgr Man: Essentials*, 10–28
from an image backup, *Sys Mgr Man: Essentials*, 10–42
from an incremental backup, *Sys Mgr Man: Essentials*, 10–44
levels of directory access, *Sys Mgr Man: Essentials*, 10–29
Restoring the queue database, *Sys Mgr Man: Essentials*, 12–12
Restricted accounts, *Sys Mgr Man: Essentials*, 6–29
RESTUSER.COM command procedure, *Sys Mgr Man: Essentials*, 10–36
Resuming printing of an output job, *Sys Mgr Man: Essentials*, 13–75, 13–76
Resuming queue status, *Sys Mgr Man: Essentials*, 13–51
Retained job status, *Sys Mgr Man: Essentials*, 13–73
definition, *Sys Mgr Man: Essentials*, 13–69
releasing jobs in, *Sys Mgr Man: Essentials*, 13–71
Retaining jobs in a queue
changing retention of a job, *Sys Mgr Man: Essentials*, 13–73
Rights databases, *Sys Mgr Man: Essentials*, 5–8
RIGHTSLIST.DAT files
default protection, *Sys Mgr Man: Essentials*, 6–4
RIGHTSLIST.DAT files (cont'd)
moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
RIGHTSLIST logical name
defining during system startup, *Sys Mgr Man: Essentials*, 5–8
defining to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
RLOGIN, *Sys Mgr Man: Tuning*, 21–13
RMS
access to files at record level, *Sys Mgr Man: Essentials*, 9–13
RMS_EXTEND_SIZE system parameter, *Sys Mgr Man: Tuning*, 16–7
Root directory
adding to an existing system disk, *Sys Mgr Man: Essentials*, 2–31
Root volume
definition, *Sys Mgr Man: Essentials*, 8–31
MFD on, *Sys Mgr Man: Essentials*, 8–31
Routers
Level 1
definition, *Sys Mgr Man: Tuning*, 21–5
Level 2
definition, *Sys Mgr Man: Tuning*, 21–6
network
definition, *Sys Mgr Man: Tuning*, 21–5
end node, *Sys Mgr Man: Tuning*, 21–6
Routing
levels of, *Sys Mgr Man: Tuning*, 21–5
network
adaptive, *Sys Mgr Man: Tuning*, 21–5
area, *Sys Mgr Man: Tuning*, 21–6
definition, *Sys Mgr Man: Tuning*, 21–5
Routing nodes
See Routers
RSM (Remote System Manager)
startup restrictions, *Sys Mgr Man: Tuning*, 23–12
RT–11 volume
block-addressable, *Sys Mgr Man: Essentials*, 9–24
RTA*n* devices, *Sys Mgr Man: Essentials*, 7–12
RUN command
effect of installed images on, *Sys Mgr Man: Tuning*, 16–10
RV60 optical disk drives
supported by UETP, *Sys Mgr Man: Tuning*, 17–7
RVN (relative volume number)
See File identification, relative volume number

S

Satellites
booting, *Sys Mgr Man: Tuning*, 22–20
cross-architecture booting, *Sys Mgr Man: Tuning*, 22–23
LAN MOP and DECnet MOP downline load service, *Sys Mgr Man: Tuning*, 22–20
migrating to LAN MOP downline load service, *Sys Mgr Man: Tuning*, 22–20
SATELLITE_PAGE.COM command procedure, *Sys Mgr Man: Essentials*, 5–6; *Sys Mgr Man: Tuning*, 15–20
execution of during system startup, *Sys Mgr Man: Essentials*, 5–5
SAVEDUMP system parameter, *Sys Mgr Man: Tuning*, 15–2, 15–3, 15–7
Save sets, *Sys Mgr Man: Essentials*, 10–6, 10–23
Files–11 disk, *Sys Mgr Man: Essentials*, 10–7
Get Save Set
VMSINSTAL.COM option, *Sys Mgr Man: Essentials*, 3–15
listing the contents of, *Sys Mgr Man: Essentials*, 10–18
magnetic tape, *Sys Mgr Man: Essentials*, 10–6, 10–7
multivolume, *Sys Mgr Man: Essentials*, 10–29
name, *Sys Mgr Man: Essentials*, 10–24
network, *Sys Mgr Man: Essentials*, 10–7
on multiple tapes, *Sys Mgr Man: Essentials*, 10–21
product
storing temporarily during installation, *Sys Mgr Man: Essentials*, 3–15
sequential disk, *Sys Mgr Man: Essentials*, 10–8
types, *Sys Mgr Man: Essentials*, 10–6
Saving the queue database, *Sys Mgr Man: Essentials*, 12–11
Scalars
definition, *Sys Mgr Man: Tuning*, 26–4
SCBs (storage control blocks)
in storage bit map file, *Sys Mgr Man: Tuning*, A–8
Scheduling, *Sys Mgr Man: Essentials*, 6–2, 6–31
See also Priority, job scheduling
of batch jobs, *Sys Mgr Man: Essentials*, 13–72
of print jobs, *Sys Mgr Man: Essentials*, 13–32, 13–72
SCROLL command, *Sys Mgr Man: Tuning*, 20–13
SCROLL keypad function, *Sys Mgr Man: Tuning*, 20–7
SCSI (Small Computer Systems Interface), *Sys Mgr Man: Essentials*, 10–40; *Sys Mgr Man: Tuning*, 21–9
SCSNODE system parameter, *Sys Mgr Man: Tuning*, 23–10
in DECdtm Services, *Sys Mgr Man: Tuning*, 25–22
unique value for DECdtm use, *Sys Mgr Man: Tuning*, 25–24
SDA (System Dump Analyzer utility)
See System Dump Analyzer utility
SDA CLUE commands
analyzing the system dump file, *Sys Mgr Man: Tuning*, 15–2
saving contents of system dump file, *Sys Mgr Man: Tuning*, 15–15
using to archive dump file information, *Sys Mgr Man: Tuning*, 15–12
using to collect dump file information, *Sys Mgr Man: Tuning*, 15–12
Search lists
priority of installed images, *Sys Mgr Man: Tuning*, 16–10
Search path of Help Message database files, *Sys Mgr Man: Essentials*, 5–28
Secondary Bootstrap image
role in boot process, *Sys Mgr Man: Essentials*, 4–2
SECONDARY day
defining for accounts, *Sys Mgr Man: Essentials*, 6–27
Secondary page and swap files, *Sys Mgr Man: Tuning*, 15–5
See also Page files
See also Swap files
creating
in system startup, *Sys Mgr Man: Tuning*, 15–20
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–17, 15–24
with SYPAGSWPFILES.COM procedure, *Sys Mgr Man: Tuning*, 15–20
improving system performance, *Sys Mgr Man: Tuning*, 16–8
installing
in system startup, *Sys Mgr Man: Tuning*, 15–20
requirement, *Sys Mgr Man: Tuning*, 15–5
with SYPAGSWPFILES.COM command procedure, *Sys Mgr Man: Tuning*, 15–20
installing during system startup, *Sys Mgr Man: Essentials*, 5–6
Secondary processors, *Sys Mgr Man: Tuning*, 26–2
Sections
global, *Sys Mgr Man: Tuning*, 16–12
pages, *Sys Mgr Man: Tuning*, 16–12

Sectors
Files–11
definition, *Sys Mgr Man: Tuning*, A–2
Security
See also Protection
alarm messages, *Sys Mgr Man: Tuning*, 18–21
alarms
enabling, *Sys Mgr Man: Tuning*, 18–29
auditing
description, *Sys Mgr Man: Tuning*, 18–25
enabling operator terminal, *Sys Mgr Man: Tuning*, 18–29
audit log files
See also Security audit log files
authenticating user's identity, *Sys Mgr Man: Tuning*, 21–14
enabling operator terminals, *Sys Mgr Man: Tuning*, 18–21
messages in operator log file, *Sys Mgr Man: Tuning*, 18–21
network
See also DECnet, security
providing for, *Sys Mgr Man: Tuning*, 21–12
OPCOM alarm messages, *Sys Mgr Man: Tuning*, 18–21
password management, *Sys Mgr Man: Essentials*, 11–2
protected subsystems, *Sys Mgr Man: Essentials*, 8–28
protecting public disk volumes, *Sys Mgr Man: Essentials*, 8–15
protecting queues, *Sys Mgr Man: Essentials*, 13–22
protecting the system dump file, *Sys Mgr Man: Tuning*, 15–4
risk of compromise by installing images with privileges, *Sys Mgr Man: Tuning*, 16–13
specifying alarm events, *Sys Mgr Man: Tuning*, 18–21
VMScluster, *Sys Mgr Man: Tuning*, 20–16
SECURITY.AUDIT$JOURNAL file
See Security audit log files
SECURITY.SYS file
See Volume security profile
Security auditing, *Sys Mgr Man: Essentials*, 11–13
archive file for each node in a cluster, *Sys Mgr Man: Tuning*, 18–27
defining log file in system startup, *Sys Mgr Man: Essentials*, 5–9
description, *Sys Mgr Man: Tuning*, 18–26
disabling events, *Sys Mgr Man: Tuning*, 18–29
displaying using SHOW AUDIT command, *Sys Mgr Man: Tuning*, 18–27
enabling operator terminal, *Sys Mgr Man: Tuning*, 18–29
Security audit log files
closing, *Sys Mgr Man: Tuning*, 18–30
creating new version, *Sys Mgr Man: Tuning*, 18–30
definition, *Sys Mgr Man: Tuning*, 18–2
reviewing, *Sys Mgr Man: Tuning*, 18–26
Security management, *Sys Mgr Man: Essentials*, 6–16
in SYSMAN on remote nodes, *Sys Mgr Man: Essentials*, 2–16
protecting queues, *Sys Mgr Man: Essentials*, 13–22 to 13–26
Security operator terminals, *Sys Mgr Man: Tuning*, 18–29
Security server
starting, *Sys Mgr Man: Essentials*, 5–5
Security Server process
intrusion database, *Sys Mgr Man: Essentials*, 11–7
network proxy database, *Sys Mgr Man: Essentials*, 11–7
SECURITY_AUDIT.AUDIT$JOURNAL file
See also Security audit log files
moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
SECURITY_SERVER
starting, *Sys Mgr Man: Essentials*, 5–5
SELECT command
in SHOW CLUSTER, *Sys Mgr Man: Tuning*, 20–13
Selective dump, *Sys Mgr Man: Tuning*, 15–3
compared to physical dump, *Sys Mgr Man: Tuning*, 15–3, 15–11
storing, *Sys Mgr Man: Tuning*, 15–11
SEQ (file sequence number)
See File identification, file sequence number
Sequential-disk save set, *Sys Mgr Man: Essentials*, 10–8
initializing, *Sys Mgr Man: Essentials*, 10–8
mounting, *Sys Mgr Man: Essentials*, 10–8
Server queues, *Sys Mgr Man: Essentials*, 13–3
Server queue status, *Sys Mgr Man: Essentials*, 13–51
Services
bindings, *Sys Mgr Man: Tuning*, 23–13
nodes, *Sys Mgr Man: Tuning*, 24–4
password protection, *Sys Mgr Man: Tuning*, 23–13
write protection, *Sys Mgr Man: Tuning*, 23–13
Sessions
maintaining on more than one terminal, *Sys Mgr Man: Essentials*, 7–11
maintaining when disconnecting a terminal, *Sys Mgr Man: Essentials*, 7–11
SET (Field) command
SHOW CLUSTER, *Sys Mgr Man: Tuning*, 20–11

SET ACCOUNTING command
controlling which resources are tracked, *Sys Mgr Man: Tuning*, 19–3
starting up a new accounting file, *Sys Mgr Man: Tuning*, 19–3
SET ACL command
for vector capability (VAX), *Sys Mgr Man: Tuning*, 26–8
modifying disk file characteristics, *Sys Mgr Man: Essentials*, 9–8
modifying file characteristics, *Sys Mgr Man: Essentials*, 9–9
SET AUDIT command
creating new version of security audit log file, *Sys Mgr Man: Tuning*, 18–30
on local node only, *Sys Mgr Man: Tuning*, 18–31
enabling security alarms, *Sys Mgr Man: Tuning*, 18–29
to create new version of security audit log file, *Sys Mgr Man: Tuning*, 18–30
to enable security auditing, *Sys Mgr Man: Tuning*, 18–26
SET command
in conversational boot, *Sys Mgr Man: Essentials*, 4–6
NCP, *Sys Mgr Man: Tuning*, 21–11
SET DEVICE command
spooling printers, *Sys Mgr Man: Essentials*, 7–15
SET DIRECTORY command
changing directory characteristics, *Sys Mgr Man: Essentials*, 9–12
limiting disk space consumed, *Sys Mgr Man: Essentials*, 8–8
modifying disk file characteristics, *Sys Mgr Man: Essentials*, 9–8
SET ENTRY command, *Sys Mgr Man: Essentials*, 13–70
changing forms, *Sys Mgr Man: Essentials*, 13–80
changing scheduling priority, *Sys Mgr Man: Essentials*, 13–72
holding jobs, *Sys Mgr Man: Essentials*, 13–70, 13–71
releasing jobs, *Sys Mgr Man: Essentials*, 13–70
requeuing pending jobs, *Sys Mgr Man: Essentials*, 13–73
specifying job retention, *Sys Mgr Man: Essentials*, 13–73
SET ENVIRONMENT command, *Sys Mgr Man: Essentials*, 2–14
SET FILE command
example, *Sys Mgr Man: Essentials*, 6–33
modifying disk file characteristics, *Sys Mgr Man: Essentials*, 9–8
using to assign an alias, *Sys Mgr Man: Essentials*, 9–10
SET FILE command (cont'd)
using to modify file characteristics, *Sys Mgr Man: Essentials*, 9–10
SET FUNCTION command, *Sys Mgr Man: Tuning*, 20–7
in SHOW CLUSTER, *Sys Mgr Man: Tuning*, 20–13
SET HOST command, *Sys Mgr Man: Tuning*, 21–13
SET HOST/LAT command, *Sys Mgr Man: Tuning*, 24–2
SET LOGINS command, *Sys Mgr Man: Tuning*, 16–4
SET MAGTAPE command, *Sys Mgr Man: Essentials*, 7–17, 8–42
SETPARAMS.DAT file, *Sys Mgr Man: Tuning*, 14–18
SET PRINTER command, *Sys Mgr Man: Essentials*, 7–13, 13–14
in system startup, *Sys Mgr Man: Essentials*, 5–12
SET PROFILE command
in SYSMAN, *Sys Mgr Man: Essentials*, 2–17
SET QUEUE command, *Sys Mgr Man: Essentials*, 13–53
assigning a default form, *Sys Mgr Man: Essentials*, 13–63
assigning characteristics, *Sys Mgr Man: Essentials*, 13–59, 13–80
canceling characteristics, *Sys Mgr Man: Essentials*, 13–59
controlling page overflow, *Sys Mgr Man: Essentials*, 13–44
mounting a form, *Sys Mgr Man: Essentials*, 13–64
setting block limits, *Sys Mgr Man: Essentials*, 13–32
setting scheduling policy, *Sys Mgr Man: Essentials*, 13–32
setting UIC-based protection on queues, *Sys Mgr Man: Essentials*, 13–23
specifying banner pages, *Sys Mgr Man: Essentials*, 13–60
specifying job processing options, *Sys Mgr Man: Essentials*, 13–31
specifying queue options with, *Sys Mgr Man: Essentials*, 13–19
specifying reset modules, *Sys Mgr Man: Essentials*, 13–65
SET QUORUM command
avoiding use of /CLUSTER with SYSMAN DO command, *Sys Mgr Man: Tuning*, 20–21
SET SECURITY command
for queues, *Sys Mgr Man: Essentials*, 13–23
modifying file characteristics, *Sys Mgr Man: Essentials*, 9–9

SET SECURITY/PROTECTION command, *Sys Mgr Man: Essentials*, 9–9
changing directory protection, *Sys Mgr Man: Essentials*, 9–12
setting default protection, *Sys Mgr Man: Essentials*, 9–8
SET/STARTUP command
in conversational boot, *Sys Mgr Man: Essentials*, 4–12
SET TERMINAL command, *Sys Mgr Man: Essentials*, 7–10, 13–14
despooling a printer, *Sys Mgr Man: Essentials*, 7–16
determining characteristics of a LAT line, *Sys Mgr Man: Essentials*, 7–13
enabling virtual terminals, *Sys Mgr Man: Essentials*, 7–12
in system startup, *Sys Mgr Man: Essentials*, 5–12, 7–11, 7–14
setting printer characteristics, *Sys Mgr Man: Essentials*, 7–13
SET TIMEOUT command, *Sys Mgr Man: Essentials*, 2–18
Setting system parameters
See Changing system parameters
Setting up printers, *Sys Mgr Man: Essentials*, 7–10, 7–13, 13–14
in system startup, *Sys Mgr Man: Essentials*, 5–12, 7–14
LAT, *Sys Mgr Man: Essentials*, 13–14
Setting up terminals, *Sys Mgr Man: Essentials*, 7–10, 7–13, 13–14
in system startup, *Sys Mgr Man: Essentials*, 5–12, 7–11
using system parameters, *Sys Mgr Man: Essentials*, 7–10
Setup module, *Sys Mgr Man: Essentials*, 13–45
See also Device control modules
specifying in forms, *Sys Mgr Man: Essentials*, 13–42
SET VOLUME command
changing protection code, *Sys Mgr Man: Essentials*, 8–18
disabling high-water marking, *Sys Mgr Man: Tuning*, 16–7
encoding labels on volumes, *Sys Mgr Man: Essentials*, 8–29
modifying disk volume characteristics, *Sys Mgr Man: Essentials*, 8–29
modifying file characteristics, *Sys Mgr Man: Essentials*, 9–8
performing data checks, *Sys Mgr Man: Essentials*, 8–29
specifying file retention periods, *Sys Mgr Man: Essentials*, 8–54
Shadow sets
backing up, *Sys Mgr Man: Essentials*, 10–40
mounting disks, *Sys Mgr Man: Essentials*, 10–41
restoring, *Sys Mgr Man: Essentials*, 10–47
Shareable images, *Sys Mgr Man: Tuning*, 16–11
assigning logical names for, *Sys Mgr Man: Tuning*, 16–17
Shareable linkage sections, *Sys Mgr Man: Tuning*, 16–13
Shared resource
definition, *Sys Mgr Man: Tuning*, 20–1
Sheet-feed paper
specifying in forms, *Sys Mgr Man: Essentials*, 13–42
Short report format
See Console report during UETP
SHOW ACCOUNTING command, *Sys Mgr Man: Tuning*, 19–2
SHOW ACL command, *Sys Mgr Man: Tuning*, 26–8
SHOW AUDIT command
to display event classes being audited, *Sys Mgr Man: Tuning*, 18–27
SHOW CLUSTER command
See Show Cluster utility
Show Cluster utility (SHOW CLUSTER)
controlling displays, *Sys Mgr Man: Tuning*, 20–5
defining keypad keys, *Sys Mgr Man: Tuning*, 20–7
refreshing the screen, *Sys Mgr Man: Tuning*, 20–10
reports, *Sys Mgr Man: Tuning*, 20–4
controlling displays, *Sys Mgr Man: Tuning*, 20–10
formatting, *Sys Mgr Man: Tuning*, 20–5, 20–11, 20–14
initialization file, *Sys Mgr Man: Tuning*, 20–10
using, *Sys Mgr Man: Tuning*, 20–4
SHOW command, *Sys Mgr Man: Essentials*, 9–1
in conversational boot, *Sys Mgr Man: Essentials*, 4–6
in SYSGEN, *Sys Mgr Man: Tuning*, 14–32
specifying languages, *Sys Mgr Man: Essentials*, 5–44, 5–45
time and date formats, *Sys Mgr Man: Essentials*, 5–43
SHOW CPU command, *Sys Mgr Man: Tuning*, 26–3, 26–9
SHOW DEVICES command, *Sys Mgr Man: Essentials*, 7–2, 7–17, 9–5
checking mounted volumes, *Sys Mgr Man: Essentials*, 8–38

SHOW DEVICES command (cont'd)
determining the status of files, *Sys Mgr Man: Essentials*, 8–44; *Sys Mgr Man: Tuning*, 16–18
devices not shown, *Sys Mgr Man: Essentials*, 7–7
examples, *Sys Mgr Man: Essentials*, 7–2
for ISO-9660 formatted devices, *Sys Mgr Man: Essentials*, 7–5
volume rebuild status, *Sys Mgr Man: Essentials*, 7–4
SHOW ENTRY command, *Sys Mgr Man: Essentials*, 13–68
SHOW ENVIRONMENT command, *Sys Mgr Man: Essentials*, 2–14
Showing system parameters
with conversational boot, *Sys Mgr Man: Essentials*, 4–6
with SYSGEN, *Sys Mgr Man: Tuning*, 14–32
with SYSMAN, *Sys Mgr Man: Tuning*, 14–27
SHOW INTRUSION command, *Sys Mgr Man: Essentials*, 11–7
SHOW MEMORY command
determining size of page and swap files, *Sys Mgr Man: Tuning*, 15–23
displaying page and swap files, *Sys Mgr Man: Tuning*, 15–6
monitoring page file usage, *Sys Mgr Man: Tuning*, 15–6
monitoring swap file usage, *Sys Mgr Man: Tuning*, 15–10
showing the size of nonpaged pool, *Sys Mgr Man: Tuning*, 26–7
SHOW PROCESS command, *Sys Mgr Man: Essentials*, 9–5; *Sys Mgr Man: Tuning*, 26–9
SHOW PROFILE command
in SYSMAN, *Sys Mgr Man: Essentials*, 2–17
SHOW PROTECTION command, *Sys Mgr Man: Essentials*, 9–5
SHOW QUEUE command
showing all jobs, *Sys Mgr Man: Essentials*, 13–50
showing batch jobs, *Sys Mgr Man: Essentials*, 13–50
showing brief information, *Sys Mgr Man: Essentials*, 13–50
showing complete information, *Sys Mgr Man: Essentials*, 13–50
showing completion status for jobs, *Sys Mgr Man: Essentials*, 13–26
showing files associated with a job, *Sys Mgr Man: Essentials*, 13–50
showing generic queues, *Sys Mgr Man: Essentials*, 13–50
showing jobs of a specified status, *Sys Mgr Man: Essentials*, 13–50
showing output execution queues, *Sys Mgr Man: Essentials*, 13–50
SHOW QUEUE command (cont'd)
showing queue status, *Sys Mgr Man: Essentials*, 13–49
showing total number of jobs, *Sys Mgr Man: Essentials*, 13–50
SHOW QUEUE/MANAGER command, *Sys Mgr Man: Essentials*, 12–11
SHOW QUOTA command, *Sys Mgr Man: Essentials*, 8–52
SHOW SECURITY command
for queues, *Sys Mgr Man: Essentials*, 13–23
Show software products, *Sys Mgr Man: Essentials*, 3–35
SHOW/STARTUP command
in conversational boot, *Sys Mgr Man: Essentials*, 4–12
SHOW_CLUSTER$INIT.COM command procedure, *Sys Mgr Man: Tuning*, 20–14
SHOW_CLUSTER$INIT logical name, *Sys Mgr Man: Tuning*, 20–14
Shutdown
See SHUTDOWN.COM command procedure
See System shutdown
SHUTDOWN$DECNET_MINUTES logical name, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$DISABLE_AUTOSTART logical name, *Sys Mgr Man: Essentials*, 4–33, 13–56
SHUTDOWN$INFORM_NODES logical name, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$MINIMUM_MINUTES logical name, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$QUEUE_MINUTES logical name, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$TIME logical name, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN$VERBOSE logical name, *Sys Mgr Man: Essentials*, 4–33
SHUTDOWN.COM command procedure, *Sys Mgr Man: Essentials*, 4–27
See also System shutdown
customizing, *Sys Mgr Man: Essentials*, 4–32, 4–33
defining time before shutdown, *Sys Mgr Man: Essentials*, 4–33
example, *Sys Mgr Man: Essentials*, 4–30
executing with SYSMAN, *Sys Mgr Man: Essentials*, 4–34
how to use, *Sys Mgr Man: Essentials*, 4–28
options
time of shutdown, *Sys Mgr Man: Essentials*, 4–33
order of events, *Sys Mgr Man: Essentials*, 4–31
reboot options, *Sys Mgr Man: Essentials*, 4–29
required privileges, *Sys Mgr Man: Essentials*, 4–27
when to use, *Sys Mgr Man: Essentials*, 4–27

Site-independent startup command procedure
See STARTUP.COM command procedure
Site-specific products
startup database, *Sys Mgr Man: Essentials*, 5–19
Site-specific shutdown command procedure
SYSHUTDWN.COM, *Sys Mgr Man: Essentials*, 4–27
Site-specific startup command procedure
order of execution, *Sys Mgr Man: Essentials*, 4–4
SATELLITE_PAGE.COM
See SATELLITE_PAGE.COM command procedure
SYCONFIG.COM
See SYCONFIG.COM command procedure
SYLOGICALS.COM
See SYLOGICALS.COM command procedure
SYPAGSWPFILES.COM
See SYPAGSWPFILES.COM command procedure
SYSECURITY.COM
See SYSECURITY.COM command procedure
SYSTARTUP_VMS.COM
See SYSTARTUP_VMS.COM command procedure
Slicing images, *Sys Mgr Man: Tuning*, 16–12
Small computer systems interface
See SCSI
SMISERVER process, *Sys Mgr Man: Essentials*, 2–12
attributes of, *Sys Mgr Man: Essentials*, 2–16
changing, *Sys Mgr Man: Essentials*, 2–16
starting, *Sys Mgr Man: Essentials*, 2–12
in system startup, *Sys Mgr Man: Essentials*, 5–5
SMP (symmetric multiprocessing)
See Multiprocessing
SMP_CPUS system parameter, *Sys Mgr Man: Tuning*, 26–3, 26–5
Snapshot facility
concepts, *Sys Mgr Man: Essentials*, 4–19
preparing system for, *Sys Mgr Man: Essentials*, 4–20
preparing system startup files for, *Sys Mgr Man: Essentials*, 4–20
required process limits, *Sys Mgr Man: Essentials*, 4–20
supported applications, *Sys Mgr Man: Essentials*, 4–20
when using with DECwindows, *Sys Mgr Man: Essentials*, 4–21
Software errors
OPCOM failure, *Sys Mgr Man: Essentials*, 2–22
queue manager, *Sys Mgr Man: Essentials*, 12–14
when booting, *Sys Mgr Man: Essentials*, 4–16
Software installation
See Installing software
identifying kit location, *Sys Mgr Man: Essentials*, 3–28
Software kits
See Software products
Software license
definition, *Sys Mgr Man: Essentials*, 3–8
Software Performance Reports
See SPRs
Software products
See also Installing software
consistent installation of, *Sys Mgr Man: Essentials*, 3–22
converting formats, *Sys Mgr Man: Essentials*, 3–33
copying to new locations, *Sys Mgr Man: Essentials*, 3–33
dependencies on other products, *Sys Mgr Man: Essentials*, 3–27
installing more than one, *Sys Mgr Man: Essentials*, 3–29
reconfiguring after installing, *Sys Mgr Man: Essentials*, 3–32
registering noncompliant software, *Sys Mgr Man: Essentials*, 3–27
release notes, *Sys Mgr Man: Essentials*, 3–29
removing after installing, *Sys Mgr Man: Essentials*, 3–34
tracking dependencies of, *Sys Mgr Man: Essentials*, 3–27
SOGW user category abbreviation, *Sys Mgr Man: Essentials*, 11–8
Sort/Merge utility (SORT/MERGE)
optimizing batch queues for, *Sys Mgr Man: Essentials*, 13–31
Source and destination locations of software kits, *Sys Mgr Man: Essentials*, 3–28
Source files for installing software
PCSI$SOURCE location, *Sys Mgr Man: Essentials*, 3–28
specifying location, *Sys Mgr Man: Essentials*, 3–28
Source parameter
VMSINSTAL.COM, *Sys Mgr Man: Essentials*, 3–11
SPAWN function
LAN Control Program (LANCP) utility, *Sys Mgr Man: Tuning*, 22–9

Spawn mode
as execution mode for a startup procedure, *Sys Mgr Man: Essentials*, 5–19
Spooled printers
See also Printers, spooled
See Printers, spooled
SPRs (Software Performance Reports)
including system dump file with, *Sys Mgr Man: Tuning*, 15–2, 15–11
submitting to report system failure, *Sys Mgr Man: Tuning*, 15–11
STABACKIT.COM command procedure, *Sys Mgr Man: Essentials*, 10–51, 10–53
Stalled job status, *Sys Mgr Man: Essentials*, 13–69
Stalled queues
status, *Sys Mgr Man: Essentials*, 13–51
Standalone BACKUP
booting, *Sys Mgr Man: Essentials*, 10–52, 10–54
building, *Sys Mgr Man: Essentials*, 10–51, 10–53
definition, *Sys Mgr Man: Essentials*, 10–51
qualifiers, *Sys Mgr Man: Essentials*, 10–50
relation to Backup utility, *Sys Mgr Man: Essentials*, 10–51
using to back up the system disk, *Sys Mgr Man: Essentials*, 10–51, 10–55, 10–58
using to restore the system disk, *Sys Mgr Man: Essentials*, 10–57
START/CPU command, *Sys Mgr Man: Tuning*, 26–3, 26–6
Starting InfoServer Client for OpenVMS, *Sys Mgr Man: Tuning*, 23–10
Starting queues, *Sys Mgr Man: Essentials*, 13–4
autostart, *Sys Mgr Man: Essentials*, 13–48
in startup command procedure, *Sys Mgr Man: Essentials*, 13–18
relationship to activating an autostart queue, *Sys Mgr Man: Essentials*, 13–4
nonautostart, *Sys Mgr Man: Essentials*, 13–16, 13–48
in startup command procedure, *Sys Mgr Man: Essentials*, 13–18
Starting queue status, *Sys Mgr Man: Essentials*, 13–51
Starting the LAT software
with LAT$STARTUP.COM, *Sys Mgr Man: Essentials*, 5–15; *Sys Mgr Man: Tuning*, 24–16, 24–17
Starting the queue manager, *Sys Mgr Man: Essentials*, 12–7
initially, *Sys Mgr Man: Essentials*, 12–7
restarting, *Sys Mgr Man: Essentials*, 12–9
STARTNET.COM command procedure, *Sys Mgr Man: Essentials*, 5–15, 7–9
START/QUEUE command, *Sys Mgr Man: Essentials*, 13–18
activating an autostart queue, *Sys Mgr Man: Essentials*, 13–49
assigning a default form, *Sys Mgr Man: Essentials*, 13–63
assigning characteristics, *Sys Mgr Man: Essentials*, 13–59
canceling characteristics, *Sys Mgr Man: Essentials*, 13–59
controlling page overflow, *Sys Mgr Man: Essentials*, 13–44
mounting a form, *Sys Mgr Man: Essentials*, 13–64
resuming printing of a suspended job, *Sys Mgr Man: Essentials*, 13–75
setting block limits, *Sys Mgr Man: Essentials*, 13–32
setting scheduling policy, *Sys Mgr Man: Essentials*, 13–32
setting UIC-based protection on queues, *Sys Mgr Man: Essentials*, 13–23
specifying autostart information, *Sys Mgr Man: Essentials*, 13–15
specifying banner pages, *Sys Mgr Man: Essentials*, 13–60
specifying job processing options, *Sys Mgr Man: Essentials*, 13–31
specifying queue options with, *Sys Mgr Man: Essentials*, 13–19
specifying reset modules, *Sys Mgr Man: Essentials*, 13–65
starting a generic queue, *Sys Mgr Man: Essentials*, 13–17
starting a nonautostart queue, *Sys Mgr Man: Essentials*, 13–48
START/QUEUE/MANAGER command, *Sys Mgr Man: Essentials*, 12–7, 12–9
caution, *Sys Mgr Man: Essentials*, 12–8
creating an additional queue manager, *Sys Mgr Man: Essentials*, 12–10
creating a queue database, *Sys Mgr Man: Essentials*, 12–7
specifying failover list, *Sys Mgr Man: Essentials*, 12–9
specifying name of queue manager, *Sys Mgr Man: Essentials*, 12–10
specifying nodes to run the queue manager, *Sys Mgr Man: Essentials*, 12–6
storage of, *Sys Mgr Man: Essentials*, 12–7
STARTUP$AUTOCONFIGURE_ALL symbol, *Sys Mgr Man: Essentials*, 7–9
STARTUP$INTERACTIVE_LOGINS symbol, *Sys Mgr Man: Essentials*, 5–16

STARTUP$STARTUP_LAYERED logical name, *Sys Mgr Man: Essentials*, 5–18
STARTUP$STARTUP_VMS logical name, *Sys Mgr Man: Essentials*, 5–18
STARTUP.COM command procedure, *Sys Mgr Man: Essentials*, 4–4, 5–3
configuring devices, *Sys Mgr Man: Essentials*, 5–7, 7–6
definition, *Sys Mgr Man: Essentials*, 5–2
description, *Sys Mgr Man: Essentials*, 4–12
if it does not execute, *Sys Mgr Man: Essentials*, 4–16
message indicating execution of, *Sys Mgr Man: Essentials*, 4–5
tasks performed by, *Sys Mgr Man: Essentials*, 4–4, 4–12; *Sys Mgr Man: Tuning*, 16–10
STARTUP command, *Sys Mgr Man: Essentials*, 2–11
See also Startup database
in SYSMAN, *Sys Mgr Man: Essentials*, 5–18
Startup command procedure
See also Site-specific startup command procedure
See also STARTUP.COM command procedure
booting without, *Sys Mgr Man: Essentials*, 4–8
changing execution mode, *Sys Mgr Man: Essentials*, 5–21
changing node restrictions, *Sys Mgr Man: Essentials*, 5–21
changing startup phase, *Sys Mgr Man: Essentials*, 5–21
creating your own, *Sys Mgr Man: Essentials*, 7–11, 7–14
enabling a temporarily disabled, *Sys Mgr Man: Essentials*, 5–22
known file lists, *Sys Mgr Man: Essentials*, 5–12
modifiable
See Site-specific startup command procedure
node restriction, *Sys Mgr Man: Essentials*, 5–19
passing parameters to, *Sys Mgr Man: Essentials*, 5–19
preventing from executing, *Sys Mgr Man: Essentials*, 5–21
temporarily, *Sys Mgr Man: Essentials*, 5–22
required
See STARTUP.COM command procedure
SATELLITE_PAGE.COM
See SATELLITE_PAGE.COM command procedure
setting up output devices, *Sys Mgr Man: Essentials*, 13–14
site-independent
Startup command procedure
site-independent (cont'd)
See also STARTUP.COM command procedure
specifying an alternate, *Sys Mgr Man: Essentials*, 4–12
as the default, *Sys Mgr Man: Essentials*, 4–13
site-specific, *Sys Mgr Man: Essentials*, 5–2, 5–3
See also Site-specific startup command procedure
announcements, *Sys Mgr Man: Essentials*, 5–14
.COM version, *Sys Mgr Man: Essentials*, 5–3
creating your own, *Sys Mgr Man: Essentials*, 5–2
definition, *Sys Mgr Man: Essentials*, 5–2
modifying, *Sys Mgr Man: Essentials*, 5–4
modifying to perform site-specific operations, *Sys Mgr Man: Essentials*, 5–2
order of execution, *Sys Mgr Man: Essentials*, 5–3
required location, *Sys Mgr Man: Essentials*, 5–4
.TEMPLATE version, *Sys Mgr Man: Essentials*, 5–3
use in VMSKITBLD, *Sys Mgr Man: Essentials*, 5–3
versions of, *Sys Mgr Man: Essentials*, 5–3
specifying execution mode, *Sys Mgr Man: Essentials*, 5–20
specifying node restrictions, *Sys Mgr Man: Essentials*, 5–20
specifying startup phase, *Sys Mgr Man: Essentials*, 5–20
starting queues, *Sys Mgr Man: Essentials*, 13–18
SYCONFIG.COM
See SYCONFIG.COM command procedure
SYLOGICALS.COM
See SYLOGICALS.COM command procedure
SYPAGSWPFILES.COM
See SYPAGSWPFILES.COM command procedure
SYSECURITY.COM
See SYSECURITY.COM command procedure
when errors prevent you from logging in, *Sys Mgr Man: Essentials*, 4–8
Startup database
adding files to, *Sys Mgr Man: Essentials*, 5–20
changing information in, *Sys Mgr Man: Essentials*, 5–21

Startup database (cont'd)
definition, *Sys Mgr Man: Essentials*, 5–18
deleting records in, *Sys Mgr Man: Essentials*, 5–21
disabling files in, *Sys Mgr Man: Essentials*, 5–22
reenabling disabled files in, *Sys Mgr Man: Essentials*, 5–22
restriction, *Sys Mgr Man: Essentials*, 5–21
showing contents of, *Sys Mgr Man: Essentials*, 5–20
showing name of the target, *Sys Mgr Man: Essentials*, 5–20
specifying the current, *Sys Mgr Man: Essentials*, 5–20
Startup phases
determining order of, *Sys Mgr Man: Essentials*, 5–18
layered product, *Sys Mgr Man: Essentials*, 5–18
END, *Sys Mgr Man: Essentials*, 5–19
LPBEGIN, *Sys Mgr Man: Essentials*, 5–19
LPBETA, *Sys Mgr Man: Essentials*, 5–19
LPMAIN, *Sys Mgr Man: Essentials*, 5–19
specifying, *Sys Mgr Man: Essentials*, 5–20
operating system
BASEENVIRON, *Sys Mgr Man: Essentials*, 5–4
CONFIGURE, *Sys Mgr Man: Essentials*, 5–4
DEVICE, *Sys Mgr Man: Essentials*, 5–4
INITIAL, *Sys Mgr Man: Essentials*, 5–4
Startup restrictions
InfoServer Client for OpenVMS software, *Sys Mgr Man: Tuning*, 23–12
PATHWORKS, *Sys Mgr Man: Tuning*, 23–12
RSM, *Sys Mgr Man: Tuning*, 23–12
SYSMAN, *Sys Mgr Man: Tuning*, 23–12
STARTUP SET OPTIONS command, *Sys Mgr Man: Essentials*, 4–15
STARTUP SHOW OPTIONS command, *Sys Mgr Man: Essentials*, 4–16
STARTUP_P1 system parameter, *Sys Mgr Man: Essentials*, 4–14
STARTUP_P2 system parameter, *Sys Mgr Man: Essentials*, 4–14
SYSMAN startup logging, *Sys Mgr Man: Essentials*, 4–15
Status of jobs
See Job status
Status of queues
See Queue status
Status of volume rebuilds, *Sys Mgr Man: Essentials*, 7–4
$STATUS values
accessing for uninstalled messages, *Sys Mgr Man: Essentials*, 5–26
Stock
See also Forms
commands used with, *Sys Mgr Man: Essentials*, 13–61
mismatch, *Sys Mgr Man: Essentials*, 13–42
troubleshooting, *Sys Mgr Man: Essentials*, 13–80
specifying, *Sys Mgr Man: Essentials*, 13–42
STOP/CPU command, *Sys Mgr Man: Tuning*, 26–3, 26–6
Stopped queue status, *Sys Mgr Man: Essentials*, 13–51
Stop pending queue status, *Sys Mgr Man: Essentials*, 13–51
Stopping queue
status, *Sys Mgr Man: Essentials*, 13–51
Stopping queues, *Sys Mgr Man: Essentials*, 13–54
abruptly, *Sys Mgr Man: Essentials*, 13–54
all queues on a node, *Sys Mgr Man: Essentials*, 12–9, 13–55
smoothly, *Sys Mgr Man: Essentials*, 13–54
Stopping the queue manager, *Sys Mgr Man: Essentials*, 12–9
STOP/QUEUE command, *Sys Mgr Man: Essentials*, 13–53, 13–75
STOP/QUEUE/MANAGER/CLUSTER command, *Sys Mgr Man: Essentials*, 12–9
STOP/QUEUE/NEXT command, *Sys Mgr Man: Essentials*, 13–49, 13–54
with autostart queues, *Sys Mgr Man: Essentials*, 13–54
STOP/QUEUE/RESET command, *Sys Mgr Man: Essentials*, 13–49, 13–54
with autostart queues, *Sys Mgr Man: Essentials*, 13–54
STOP/QUEUES/ON_NODE command, *Sys Mgr Man: Essentials*, 12–9
entering before shutting down a system, *Sys Mgr Man: Essentials*, 13–55
relationship to DISABLE AUTOSTART /QUEUES command, *Sys Mgr Man: Essentials*, 13–55
Storage bit map file
BITMAP.SYS, *Sys Mgr Man: Tuning*, A–8
definition, *Sys Mgr Man: Tuning*, A–8
reserved file, *Sys Mgr Man: Tuning*, A–8
storage control block (SCB), *Sys Mgr Man: Tuning*, A–8
Storage control blocks
See SCBs
StorageWorks RAID Array, *Sys Mgr Man: Essentials*, 10–41
SUBMIT command
preventing users from executing, *Sys Mgr Man: Essentials*, 13–53
processing of, *Sys Mgr Man: Essentials*, 13–2

SUBMIT command (cont'd)
specifying characteristics, *Sys Mgr Man: Essentials*, 13–28
specifying job retention, *Sys Mgr Man: Essentials*, 13–73
specifying scheduling priority, *Sys Mgr Man: Essentials*, 13–72
SUBMON.COM command procedure
sample, *Sys Mgr Man: Tuning*, 18–42
used with Monitor utility, *Sys Mgr Man: Tuning*, 18–40
Subprocesses
creating with the LANCP SPAWN function, *Sys Mgr Man: Tuning*, 22–9
subprocess creation limit, *Sys Mgr Man: Essentials*, 6–2, 6–44
Substituting volumes, *Sys Mgr Man: Essentials*, 8–27
Subsystem ACEs
example, *Sys Mgr Man: Essentials*, 11–10
Subsystems
protected, *Sys Mgr Man: Essentials*, 8–28
Supervisor mode
logical names, *Sys Mgr Man: Tuning*, 16–15
Suppressing message display, *Sys Mgr Man: Tuning*, 14–12
Suspending a job, *Sys Mgr Man: Essentials*, 13–75
SWAPFILE.SYS file, *Sys Mgr Man: Tuning*, 15–4
See also Swap files
SWAPFILE*n*_NAME symbol, *Sys Mgr Man: Tuning*, 15–17, 15–24
SWAPFILE*n*_SIZE symbol, *Sys Mgr Man: Tuning*, 15–18, 15–24
Swap files
changing sizes
with SWAPFILES.COM, *Sys Mgr Man: Tuning*, 15–25
creating
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–17, 15–24
with SYSGEN, *Sys Mgr Man: Tuning*, 15–18
definition, *Sys Mgr Man: Tuning*, 15–4
deleting after creating a new version, *Sys Mgr Man: Tuning*, 15–28
displaying, *Sys Mgr Man: Tuning*, 15–6
displaying the size calculated by AUTOGEN, *Sys Mgr Man: Tuning*, 15–18, 15–22
fragmentation of, *Sys Mgr Man: Tuning*, 15–27
installing
in system startup, *Sys Mgr Man: Essentials*, 5–5, 5–6; *Sys Mgr Man: Tuning*, 15–5, 15–20, 15–21
when resized with AUTOGEN, *Sys Mgr Man: Tuning*, 15–18, 15–22
with SYPAGSWPFILES.COM procedure, *Sys Mgr Man: Tuning*, 15–20
Swap files
installing (cont'd)
with SYSGEN, *Sys Mgr Man: Tuning*, 15–19
location
specifying for individual files, *Sys Mgr Man: Tuning*, 15–24
message indicating high fragmentation, *Sys Mgr Man: Tuning*, 15–27
monitoring usage of, *Sys Mgr Man: Tuning*, 15–6, 15–10
mounting disk during system startup, *Sys Mgr Man: Essentials*, 5–6; *Sys Mgr Man: Tuning*, 15–20
moving to improve performance, *Sys Mgr Man: Tuning*, 16–8
on a satellite, *Sys Mgr Man: Tuning*, 15–20
primary, *Sys Mgr Man: Tuning*, 15–5
purging, *Sys Mgr Man: Tuning*, 15–28
requirements
location, *Sys Mgr Man: Tuning*, 15–5
secondary, *Sys Mgr Man: Tuning*, 15–5, 15–20
size
See Swap file sizes
tasks for managing, *Sys Mgr Man: Tuning*, 15–1
VMScluster satellite node, *Sys Mgr Man: Essentials*, 5–6
SWAPFILES.COM command procedure
changing size of primary page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–25
Swap file sizes
calculating
manually, *Sys Mgr Man: Tuning*, 15–10
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–5
changing
recommended method, *Sys Mgr Man: Tuning*, 15–22
when to increase, *Sys Mgr Man: Tuning*, 15–10
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–22
with SYSGEN, *Sys Mgr Man: Tuning*, 15–26
determining current, *Sys Mgr Man: Tuning*, 15–23
displaying AUTOGEN's calculations, *Sys Mgr Man: Tuning*, 15–18, 15–22
specifying
for individual files, *Sys Mgr Man: Tuning*, 15–22, 15–24
total for multiple files, *Sys Mgr Man: Tuning*, 15–22, 15–23
when to increase, *Sys Mgr Man: Tuning*, 15–10
SWAPFILE symbol, *Sys Mgr Man: Tuning*, 15–24

Swapping
to move information between physical memory and files stored on disk, *Sys Mgr Man: Tuning*, 15–4
SYCONFIG.COM command procedure, *Sys Mgr Man: Essentials*, 5–3
AUTOGEN failure, *Sys Mgr Man: Essentials*, 7–10
configuring devices, *Sys Mgr Man: Essentials*, 7–6
in startup, *Sys Mgr Man: Essentials*, 5–4
modifying to connect special devices, *Sys Mgr Man: Essentials*, 5–7
modifying to mount disks early, *Sys Mgr Man: Essentials*, 5–12
STARTUP$AUTOCONFIGURE_ALL symbol, *Sys Mgr Man: Essentials*, 7–9
SYLOGICALS.COM command procedure, *Sys Mgr Man: Essentials*, 5–3, 5–8
in system startup, *Sys Mgr Man: Essentials*, 5–5
mounting the queue database disk, *Sys Mgr Man: Essentials*, 12–6
redefining location of master file of queue database, *Sys Mgr Man: Essentials*, 12–6
redefining location of system files, *Sys Mgr Man: Tuning*, 16–8
to specify default state of operator log files, *Sys Mgr Man: Tuning*, 18–23
SYLOGIN.COM command procedure, *Sys Mgr Man: Essentials*, 5–16, 6–20
ensuring execution, *Sys Mgr Man: Essentials*, 6–13
sample systemwide, *Sys Mgr Man: Essentials*, 6–20
SYLOGOUT.COM command procedure, *Sys Mgr Man: Essentials*, 6–22
Symbionts, *Sys Mgr Man: Essentials*, 13–53
bypass formatting, *Sys Mgr Man: Essentials*, 13–44
communicating with, *Sys Mgr Man: Essentials*, 13–75
default, *Sys Mgr Man: Essentials*, 13–3
determining, *Sys Mgr Man: Essentials*, 13–78
for LAT printers, *Sys Mgr Man: Essentials*, 13–3, 13–78
function of, *Sys Mgr Man: Essentials*, 12–3, 13–2
LATSYM, *Sys Mgr Man: Essentials*, 13–3, 13–78
PRTSMB on LAT printers, *Sys Mgr Man: Essentials*, 13–78
role in processing print jobs, *Sys Mgr Man: Essentials*, 13–3
user-written, *Sys Mgr Man: Essentials*, 13–3
Symbols
See also Logical names
Symbols (cont'd)
defining in MODPARAMS.DAT file, *Sys Mgr Man: Tuning*, 14–19, 15–17
for page, swap, and dump file sizes, *Sys Mgr Man: Tuning*, 15–23 to 15–25
for system parameters, *Sys Mgr Man: Tuning*, 14–19
NUM_ETHERADAPT, *Sys Mgr Man: Tuning*, 14–21
NUM_NODES, *Sys Mgr Man: Tuning*, 14–21
PAGEFILE*n*_NAME, *Sys Mgr Man: Tuning*, 15–17
PAGEFILE*n*_SIZE, *Sys Mgr Man: Tuning*, 15–18
STARTUP$AUTOCONFIGURE_ALL, *Sys Mgr Man: Essentials*, 7–9
STARTUP$INTERACTIVE_LOGINS, *Sys Mgr Man: Essentials*, 5–16
SWAPFILE*n*_NAME, *Sys Mgr Man: Tuning*, 15–17
SWAPFILE*n*_SIZE, *Sys Mgr Man: Tuning*, 15–18
Symmetric multiprocessing
See Multiprocessing
Symmetric vector processing configuration, *Sys Mgr Man: Tuning*, 26–4
SYPAGSWPFILES.COM command procedure, *Sys Mgr Man: Essentials*, 5–3; *Sys Mgr Man: Tuning*, 15–20
execution of during system startup, *Sys Mgr Man: Essentials*, 5–5, 5–6
modifying to install page and swap files, *Sys Mgr Man: Essentials*, 5–6
SYS$ANNOUNCE logical name, *Sys Mgr Man: Essentials*, 5–14
SYS$AUDIT_SERVER_INHIBIT logical name, *Sys Mgr Man: Essentials*, 5–9; *Sys Mgr Man: Tuning*, 18–28
SYS$BATCH default queue name, *Sys Mgr Man: Essentials*, 13–5
SYS$DECDTM_INHIBIT logical name, *Sys Mgr Man: Tuning*, 25–21
SYS$ERRORLOG logical name
defining during system startup, *Sys Mgr Man: Essentials*, 5–8
defining to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
SYS$INPUT logical name, *Sys Mgr Man: Tuning*, 17–31
SYS$JOURNAL logical name, *Sys Mgr Man: Tuning*, 25–3
SYS$LANGUAGE logical, *Sys Mgr Man: Essentials*, 5–50
SYS$MONITOR logical name
defining during system startup, *Sys Mgr Man: Essentials*, 5–8
defining to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8

SYS$OUTPUT logical name, *Sys Mgr Man: Tuning*, 17–33
SYS$PRINT default queue name, *Sys Mgr Man: Essentials*, 13–8
SYS$QUEUE_MANAGER.QMAN$JOURNAL file
See Journal file of queue database
SYS$QUEUE_MANAGER.QMAN$QUEUES file
See Queue file of queue database
SYS$QUEUE_MANAGER queue manager
as default queue manager, *Sys Mgr Man: Essentials*, 12–4
SYS$STARTUP logical name, *Sys Mgr Man: Essentials*, 4–4, 5–2, 5–18
SYS$SYLOGIN logical name, *Sys Mgr Man: Essentials*, 5–17
SYS$SYSTEM:NETNODE_REMOTE.DAT file
changing location of, *Sys Mgr Man: Tuning*, 21–10
contains configuration database, *Sys Mgr Man: Tuning*, 21–10
SYS$TEST logical name, *Sys Mgr Man: Tuning*, 17–4, 17–11, 17–20
SYS$UPDATE logical name
with VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–9
SYS$WELCOME logical name, *Sys Mgr Man: Essentials*, 5–15
SYSBOOT
See Conversational boot
SYSBOOT.EXE file, *Sys Mgr Man: Essentials*, 4–2
SYSDUMP.DMP file, *Sys Mgr Man: Tuning*, 15–2 to 15–4
See also System dump files
protection, *Sys Mgr Man: Tuning*, 15–4
required location, *Sys Mgr Man: Tuning*, 15–3
SYSECURITY.COM command procedure, *Sys Mgr Man: Essentials*, 5–3, 5–9
during system startup, *Sys Mgr Man: Essentials*, 5–5
SYSGEN (System Generation utility)
See System Generation utility
SYSGEN parameters
See System parameters
SYSHUTDWN.COM command procedure, *Sys Mgr Man: Essentials*, 4–27
SYSLOST.DIR file
lost files in, *Sys Mgr Man: Essentials*, 8–57
SYSMAN
See System Management utility
SYSMANINI logical name, *Sys Mgr Man: Essentials*, 2–19
SYSMAN parameters
See System parameters
SYSPRV privilege
giving rights of system user, *Sys Mgr Man: Essentials*, 11–8
SYSTARTUP_VMS.COM command procedure, *Sys Mgr Man: Essentials*, 5–3 to 5–17
creating systemwide announcements, *Sys Mgr Man: Essentials*, 5–14
defining announcements in, *Sys Mgr Man: Essentials*, 5–14
defining location of systemwide login procedure, *Sys Mgr Man: Essentials*, 5–17
defining welcome messages in, *Sys Mgr Man: Essentials*, 5–15
disabling error checking in, *Sys Mgr Man: Essentials*, 5–10
editing to start DECnet for OpenVMS VAX, *Sys Mgr Man: Essentials*, 5–15
enabling autostart, *Sys Mgr Man: Essentials*, 5–12
freeing page file of dump information, *Sys Mgr Man: Tuning*, 15–4, 15–17
installing known images, *Sys Mgr Man: Essentials*, 5–12; *Sys Mgr Man: Tuning*, 16–10
installing resident images (Alpha), *Sys Mgr Man: Essentials*, 5–13
invoking LAT command procedures, *Sys Mgr Man: Essentials*, 5–15
invoking the System Dump Analyzer utility, *Sys Mgr Man: Tuning*, 15–15
limiting the number of interactive users, *Sys Mgr Man: Essentials*, 5–16
making remote InfoServer disks available, *Sys Mgr Man: Essentials*, 5–13; *Sys Mgr Man: Tuning*, 23–14
message indicating execution of, *Sys Mgr Man: Essentials*, 4–5
modifying to perform site-specific operations during system startup, *Sys Mgr Man: Essentials*, 5–10
operations performed in, *Sys Mgr Man: Essentials*, 5–10
purging the operator log file, *Sys Mgr Man: Essentials*, 5–14
saving contents of system dump file in, *Sys Mgr Man: Tuning*, 15–15
setting printer device characteristics, *Sys Mgr Man: Essentials*, 5–12, 7–14
setting terminal device characteristics, *Sys Mgr Man: Essentials*, 5–12, 7–11
special consideration about operator assistance for MOUNT command, *Sys Mgr Man: Essentials*, 5–11
starting InfoServer Client for OpenVMS, *Sys Mgr Man: Essentials*, 5–13; *Sys Mgr Man: Tuning*, 23–10
starting queues, *Sys Mgr Man: Essentials*, 5–12

SYSTARTUP_VMS.COM command procedure (cont'd)
submitting batch jobs from, *Sys Mgr Man: Essentials*, 5–14
SYSTEM account
changing passwords for security of, *Sys Mgr Man: Essentials*, 2–9
exercising caution with privileges, *Sys Mgr Man: Essentials*, 2–9
initial modification, *Sys Mgr Man: Essentials*, 6–9
in UAF, *Sys Mgr Man: Essentials*, 6–6, 6–7
logging in to, *Sys Mgr Man: Essentials*, 2–10
setting process quotas for efficient backups, *Sys Mgr Man: Essentials*, 10–9
using AUTHORIZE to modify process limits, *Sys Mgr Man: Essentials*, 3–7
System console
? message, *Sys Mgr Man: Essentials*, 4–16
System crash
See CRASH.COM command procedure
See Crash dumps
See System failures
System directories
restoring original names
before upgrading, *Sys Mgr Man: Essentials*, 3–5
System disks
adding an alternate root directory, *Sys Mgr Man: Essentials*, 2–31
automatic mounting of, *Sys Mgr Man: Essentials*, 5–11
backing up, *Sys Mgr Man: Essentials*, 5–50, 10–47, 10–55, 10–58
backing up after installation, *Sys Mgr Man: Essentials*, 3–13
backing up for software installations, *Sys Mgr Man: Essentials*, 3–7
booting from an alternate, *Sys Mgr Man: Essentials*, 4–3
building with VMSKITBLD, *Sys Mgr Man: Essentials*, 2–27
completing a disk created with VMSKITBLD, *Sys Mgr Man: Essentials*, 2–29
configuring a system root added with VMSKITBLD, *Sys Mgr Man: Essentials*, 2–32
copying system files from, *Sys Mgr Man: Essentials*, 2–26
copying system files using VMSKITBLD, *Sys Mgr Man: Essentials*, 2–29
disk space needed to run UETP, *Sys Mgr Man: Tuning*, 17–5
fragmentation of, *Sys Mgr Man: Tuning*, 15–27
installing software on alternate, *Sys Mgr Man: Essentials*, 3–18
System disks (cont'd)
moving files off to improve system performance, *Sys Mgr Man: Tuning*, 16–8
moving page and swap files off, *Sys Mgr Man: Tuning*, 15–5
not in volume sets, *Sys Mgr Man: Essentials*, 8–31
quotas for, *Sys Mgr Man: Essentials*, 8–50
removing and adding optional system files, *Sys Mgr Man: Essentials*, 5–1
restoring, *Sys Mgr Man: Essentials*, 10–57
saving space by removing optional files, *Sys Mgr Man: Essentials*, 5–1
saving space on, *Sys Mgr Man: Tuning*, 15–5
test error during UETP, *Sys Mgr Man: Tuning*, 17–24, 17–25
testing with UETP, *Sys Mgr Man: Tuning*, 17–33
UETP test image, *Sys Mgr Man: Tuning*, 17–34
System Dump Analyzer utility (SDA)
analyzing the system dump file, *Sys Mgr Man: Tuning*, 15–2
in system startup, *Sys Mgr Man: Essentials*, 5–13; *Sys Mgr Man: Tuning*, 15–4, 15–11
COPY command, *Sys Mgr Man: Tuning*, 15–15
determining cause of system failure, *Sys Mgr Man: Tuning*, 15–11
freeing dump information from the page file, *Sys Mgr Man: Tuning*, 15–16
saving contents of system dump file, *Sys Mgr Man: Tuning*, 15–15
System dump files
See also Crash dumps
See also SYSDUMP.DMP file
See also System dump file sizes
See also System failures
analyzing, *Sys Mgr Man: Tuning*, 15–11
calculating size, *Sys Mgr Man: Tuning*, 15–6
comparison of contents of physical and selective dumps, *Sys Mgr Man: Tuning*, 15–3, 15–11
copying with BACKUP, *Sys Mgr Man: Tuning*, 15–4, 15–14
default location, *Sys Mgr Man: Tuning*, 15–2
definition, *Sys Mgr Man: Tuning*, 15–2
deleting after creating a new version, *Sys Mgr Man: Tuning*, 15–28
displaying the size calculated by AUTOGEN, *Sys Mgr Man: Tuning*, 15–18, 15–22
freeing page file, *Sys Mgr Man: Tuning*, 15–4
information captured in, *Sys Mgr Man: Tuning*, 15–2
installing
automatically, *Sys Mgr Man: Tuning*, 15–2

System dump files
installing (cont'd)
when resized with AUTOGEN, *Sys Mgr Man: Tuning*, 15–18, 15–22
insufficient disk space, *Sys Mgr Man: Tuning*, 15–11
investigating cause of system failure, *Sys Mgr Man: Tuning*, 15–11
lack of, *Sys Mgr Man: Tuning*, 15–2
overwriting of, *Sys Mgr Man: Tuning*, 15–2
protecting with UIC security, *Sys Mgr Man: Tuning*, 15–4
requirements, *Sys Mgr Man: Tuning*, 15–3
location, *Sys Mgr Man: Tuning*, 15–3
size, *Sys Mgr Man: Tuning*, 15–4
saving contents on reboot, *Sys Mgr Man: Essentials*, 5–13; *Sys Mgr Man: Tuning*, 15–15
saving minimal information in, *Sys Mgr Man: Tuning*, 15–11
size
See System dump file sizes
storing selective portions of memory, *Sys Mgr Man: Tuning*, 15–11
tasks for managing, *Sys Mgr Man: Tuning*, 15–1
use of page file for, *Sys Mgr Man: Tuning*, 15–2
System dump file sizes
calculating
manually, *Sys Mgr Man: Tuning*, 15–6
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–2
changing
recommended method, *Sys Mgr Man: Tuning*, 15–22
with AUTOGEN, *Sys Mgr Man: Tuning*, 15–11, 15–22
with SYSGEN, *Sys Mgr Man: Tuning*, 15–26
displaying AUTOGEN's calculations, *Sys Mgr Man: Tuning*, 15–18, 15–22
minimizing, *Sys Mgr Man: Tuning*, 15–11
required, *Sys Mgr Man: Tuning*, 15–4
for page file, *Sys Mgr Man: Tuning*, 15–4
System failures, *Sys Mgr Man: Tuning*, 15–2
See also Crash dumps
See also System dump files
determining cause, *Sys Mgr Man: Tuning*, 15–2, 15–11
reporting with a Software Performance Report, *Sys Mgr Man: Tuning*, 15–11
saving contents of system dump file after, *Sys Mgr Man: Essentials*, 5–13; *Sys Mgr Man: Tuning*, 15–15
writing of system dump file, *Sys Mgr Man: Tuning*, 15–2
System files
duplicating using VMSKITBLD, *Sys Mgr Man: Essentials*, 2–26
moving off system disk to improve performance, *Sys Mgr Man: Tuning*, 16–8
on public volumes, *Sys Mgr Man: Essentials*, 8–8
optional
adding or deleting, *Sys Mgr Man: Essentials*, 5–2
System Generation utility (SYSGEN)
and version checking, *Sys Mgr Man: Essentials*, 5–23
AUTOCONFIGURE command (VAX)
in system startup, *Sys Mgr Man: Essentials*, 5–7
changing page, swap, and dump file sizes, *Sys Mgr Man: Tuning*, 15–22, 15–26
changing system parameter file with, *Sys Mgr Man: Tuning*, 14–34
changing system parameters, *Sys Mgr Man: Tuning*, 14–33, 14–34, 18–20
See also System parameters, changing
configuring devices
in system startup, *Sys Mgr Man: Essentials*, 5–7
CONNECT command (VAX), *Sys Mgr Man: Essentials*, 7–7
in system startup, *Sys Mgr Man: Essentials*, 5–7
converting parameters for use with AUTOGEN, *Sys Mgr Man: Tuning*, 14–5
CREATE command, *Sys Mgr Man: Tuning*, 15–18, 15–26
creating a new system parameter file, *Sys Mgr Man: Tuning*, 14–35
creating page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–18
INSTALL command, *Sys Mgr Man: Tuning*, 15–19
in SYPAGSWPFILES.COM, *Sys Mgr Man: Tuning*, 15–21
in system startup, *Sys Mgr Man: Essentials*, 5–7
installing page, swap, and dump files, *Sys Mgr Man: Tuning*, 15–19
in SYPAGSWPFILES.COM, *Sys Mgr Man: Tuning*, 15–21
installing page, swap and dump files
in system startup, *Sys Mgr Man: Essentials*, 5–7
LOAD command (VAX), *Sys Mgr Man: Essentials*, 7–7
managing system parameters, *Sys Mgr Man: Tuning*, 14–4
operator log messages, *Sys Mgr Man: Tuning*, 18–20

System Generation utility (SYSGEN) (cont'd)
showing system parameters, *Sys Mgr Man: Tuning*, 14–32
System hangups, *Sys Mgr Man: Tuning*, 17–21, 17–29
System libraries
decompressing, *Sys Mgr Man: Tuning*, 16–6
System management
centralizing with SYSMAN, *Sys Mgr Man: Essentials*, 2–10
clearing counters, *Sys Mgr Man: Tuning*, 22–25
creating access control lists (ACLs), *Sys Mgr Man: Essentials*, 11–9
deleting device information, *Sys Mgr Man: Tuning*, 22–17
deleting node information, *Sys Mgr Man: Tuning*, 22–20
disabling MOP downline load service, *Sys Mgr Man: Tuning*, 22–23
displaying device information, *Sys Mgr Man: Tuning*, 22–15
displaying LAN device configurations, *Sys Mgr Man: Tuning*, 22–9
displaying LAN device parameters, *Sys Mgr Man: Tuning*, 22–10
displaying node information, *Sys Mgr Man: Tuning*, 22–18
displaying OPCOM messages, *Sys Mgr Man: Tuning*, 22–25
displaying status and counters, *Sys Mgr Man: Tuning*, 22–23, 22–24
enabling MOP downline load service, *Sys Mgr Man: Tuning*, 22–23
environment, *Sys Mgr Man: Essentials*, 2–12 to 2–15
LANACP device database, *Sys Mgr Man: Tuning*, 22–14
LAN Auxiliary Control Program (LANACP) utility, *Sys Mgr Man: Tuning*, 22–5
LAN Control Program (LANCP) utility, *Sys Mgr Man: Tuning*, 22–6
LANCP command files, *Sys Mgr Man: Tuning*, 22–9
LANCP SPAWN function, *Sys Mgr Man: Tuning*, 22–9
LAN devices, *Sys Mgr Man: Tuning*, 22–9
LAN enhancements, *Sys Mgr Man: Tuning*, 22–1
LAN firmware updates, *Sys Mgr Man: Tuning*, 22–14
LAN node database management, *Sys Mgr Man: Tuning*, 22–17
load trace facility, *Sys Mgr Man: Tuning*, 22–25
MOP console carrier, *Sys Mgr Man: Tuning*, 22–26
System management (cont'd)
MOP downline load service management, *Sys Mgr Man: Tuning*, 22–23
MOP trigger boot, *Sys Mgr Man: Tuning*, 22–27
on multiple nodes, *Sys Mgr Man: Essentials*, 2–13
running the LANACP utility, *Sys Mgr Man: Tuning*, 22–6
running the LANCP utility, *Sys Mgr Man: Tuning*, 22–7
setting device information, *Sys Mgr Man: Tuning*, 22–15
setting LAN device parameters, *Sys Mgr Man: Tuning*, 22–12
setting node information, *Sys Mgr Man: Tuning*, 22–18
stopping the LANACP utility, *Sys Mgr Man: Tuning*, 22–6
tasks
clusterwide management, *Sys Mgr Man: Tuning*, 20–3
establishing node in network, *Sys Mgr Man: Tuning*, 21–11
System Management utility (SYSMAN)
accessing disks, *Sys Mgr Man: Essentials*, 8–51
adding startup files to a startup database, *Sys Mgr Man: Essentials*, 5–20, 5–21
ALF commands, *Sys Mgr Man: Essentials*, 6–32
authorization checks in, *Sys Mgr Man: Essentials*, 2–16
changing privileges in, *Sys Mgr Man: Essentials*, 2–17
changing system parameters
active values, *Sys Mgr Man: Tuning*, 14–29
command verification in, *Sys Mgr Man: Essentials*, 2–17
configuring devices (Alpha)
in system startup, *Sys Mgr Man: Essentials*, 5–7
converting parameters for use with AUTOGEN, *Sys Mgr Man: Tuning*, 14–5
creating command procedures for, *Sys Mgr Man: Essentials*, 2–18
deleting startup files, *Sys Mgr Man: Essentials*, 5–21
disabling startup files, *Sys Mgr Man: Essentials*, 5–22
DISKQUOTA commands, *Sys Mgr Man: Essentials*, 8–50
disk quotas with, *Sys Mgr Man: Essentials*, 8–49
Disk Quota utility, *Sys Mgr Man: Essentials*, 8–50
DO command, *Sys Mgr Man: Essentials*, 2–18

System Management utility (SYSMAN) (cont'd)
enabling remote systems to execute commands, *Sys Mgr Man: Essentials*, 2–12
enabling startup files, *Sys Mgr Man: Essentials*, 5–22
features of, *Sys Mgr Man: Essentials*, 2–10
how commands execute, *Sys Mgr Man: Essentials*, 2–11
initialization file, *Sys Mgr Man: Essentials*, 2–19
IO AUTOCONFIGURE command (Alpha)
in system startup, *Sys Mgr Man: Essentials*, 5–7
IO CONNECT command (Alpha), *Sys Mgr Man: Essentials*, 7–8
in system startup, *Sys Mgr Man: Essentials*, 5–7
IO LOAD (Alpha), *Sys Mgr Man: Essentials*, 7–8
loading licenses with, *Sys Mgr Man: Essentials*, 3–8
management environment, *Sys Mgr Man: Essentials*, 2–12
managing a VMScluster, *Sys Mgr Man: Tuning*, 20–16 to 20–23
managing startup, *Sys Mgr Man: Essentials*, 5–18
managing system parameters, *Sys Mgr Man: Tuning*, 14–4, 14–25
modifying the system parameter file, *Sys Mgr Man: Tuning*, 14–28
PARAMETERS command, *Sys Mgr Man: Tuning*, 14–25
summary, *Sys Mgr Man: Tuning*, 14–25
privileges required, *Sys Mgr Man: Essentials*, 2–11
profile, *Sys Mgr Man: Essentials*, 2–16
adjusting, *Sys Mgr Man: Essentials*, 2–16
showing system parameters, *Sys Mgr Man: Tuning*, 14–27
showing the contents of a startup database, *Sys Mgr Man: Essentials*, 5–20
showing the name of the target startup database, *Sys Mgr Man: Essentials*, 5–20
shutdown, *Sys Mgr Man: Essentials*, 4–34
SMISERVER process, *Sys Mgr Man: Essentials*, 2–12
specifying the current startup database, *Sys Mgr Man: Essentials*, 5–20
STARTUP command, *Sys Mgr Man: Essentials*, 5–18
startup logging, *Sys Mgr Man: Essentials*, 4–15
startup restrictions, *Sys Mgr Man: Tuning*, 23–12
timeout periods, *Sys Mgr Man: Essentials*, 2–18
System Management utility (SYSMAN) (cont'd)
use of passwords, *Sys Mgr Man: Essentials*, 2–13, 2–16
using logical names in, *Sys Mgr Man: Essentials*, 2–14
using to centralize system management, *Sys Mgr Man: Essentials*, 2–10
System messages
See also Messages
using when managing a system, *Sys Mgr Man: Essentials*, 2–5
System parameters
See also Parameter files
ACP cache system, *Sys Mgr Man: Essentials*, 8–43
active values, *Sys Mgr Man: Tuning*, 14–3, 14–26
affected by AUTOGEN calculations, *Sys Mgr Man: Tuning*, 14–10
ALPHAVMSSYS.PAR file (Alpha), *Sys Mgr Man: Tuning*, 14–3
automatic setting by AUTOGEN, *Sys Mgr Man: Tuning*, 14–3
booting with default, *Sys Mgr Man: Essentials*, 4–8
categories by function, *Sys Mgr Man: Tuning*, 14–2
changing
checking AUTOGEN's settings, *Sys Mgr Man: Tuning*, 14–10
editing MODPARAMS.DAT file, *Sys Mgr Man: Tuning*, 14–18
recommended method, *Sys Mgr Man: Tuning*, 14–4, 14–18
specifying values in MODPARAMS.DAT file, *Sys Mgr Man: Tuning*, 14–4
with AUTOGEN, *Sys Mgr Man: Tuning*, 14–18
with conversational boot, *Sys Mgr Man: Essentials*, 4–3, 4–6; *Sys Mgr Man: Tuning*, 14–36
with SYSGEN, *Sys Mgr Man: Tuning*, 14–33, 14–34
with SYSMAN, *Sys Mgr Man: Tuning*, 14–28, 14–29
checking before using VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–7
controlling
increasing, *Sys Mgr Man: Tuning*, 14–19
in MODPARAMS.DAT file, *Sys Mgr Man: Tuning*, 14–4, 14–18
specifying absolute values, *Sys Mgr Man: Tuning*, 14–19
specifying maximum values, *Sys Mgr Man: Tuning*, 14–20
specifying minimum values, *Sys Mgr Man: Tuning*, 14–20

System parameters
controlling (cont'd)
with ADD_ prefix, *Sys Mgr Man: Tuning*, 14–19
with MAX_ prefix, *Sys Mgr Man: Tuning*, 14–20
with MIN_ prefix, *Sys Mgr Man: Tuning*, 14–20
creating a new parameter file
with SYSGEN, *Sys Mgr Man: Tuning*, 14–35
current values, *Sys Mgr Man: Tuning*, 14–3, 14–26
default values, *Sys Mgr Man: Tuning*, 14–3
definition, *Sys Mgr Man: Tuning*, 14–1
DUMPBUG, *Sys Mgr Man: Tuning*, 15–3
DUMPSTYLE, *Sys Mgr Man: Tuning*, 15–3, 15–11
dynamic, *Sys Mgr Man: Tuning*, 14–3
effect on other parameters, *Sys Mgr Man: Tuning*, 14–4
ERLBUFFERPAGES, *Sys Mgr Man: Tuning*, 15–6
ERRORLOGBUFFERS, *Sys Mgr Man: Tuning*, 15–6
file extensions, *Sys Mgr Man: Tuning*, 16–7
GBLPAGES, *Sys Mgr Man: Tuning*, 16–12
GBLSECTIONS, *Sys Mgr Man: Tuning*, 16–12
initialization at boot time, *Sys Mgr Man: Tuning*, 14–36
in memory
See Active system parameters
MODPARAMS.DAT file
See MODPARAMS.DAT file
MULTIPROCESSING, *Sys Mgr Man: Tuning*, 26–3
MVTIMEOUT, *Sys Mgr Man: Essentials*, 8–60, 8–62
NPAGEDYN, *Sys Mgr Man: Tuning*, 26–7
on disk, *Sys Mgr Man: Tuning*, 14–3
See also Current system parameters
See Current system parameters
in ALPHAVMSSYS.PAR file (Alpha), *Sys Mgr Man: Tuning*, 14–36
in VAXVMSSYS.PAR file (VAX), *Sys Mgr Man: Tuning*, 14–36
parameter files
See Parameter files
QUANTUM, *Sys Mgr Man: Tuning*, 26–8
recommended method for changing, *Sys Mgr Man: Tuning*, 14–4, 14–5
RMS_EXTEND_SIZE, *Sys Mgr Man: Tuning*, 16–7
SAVEDUMP, *Sys Mgr Man: Tuning*, 15–2, 15–3
SCSNODE, *Sys Mgr Man: Tuning*, 23–10
showing
System parameters
showing (cont'd)
with conversational boot, *Sys Mgr Man: Essentials*, 4–3, 4–6; *Sys Mgr Man: Tuning*, 14–36
with SYSGEN, *Sys Mgr Man: Tuning*, 14–32
with SYSMAN, *Sys Mgr Man: Tuning*, 14–27
SMP_CPUS, *Sys Mgr Man: Tuning*, 26–3, 26–5
STARTUP_P1, *Sys Mgr Man: Essentials*, 4–14
STARTUP_P2, *Sys Mgr Man: Essentials*, 4–14
storing your changes for use with AUTOGEN, *Sys Mgr Man: Tuning*, 14–5
symmetric multiprocessing, *Sys Mgr Man: Tuning*, 26–3
TAPE_MVTIMEOUT, *Sys Mgr Man: Essentials*, 8–60, 8–62
tasks for managing, *Sys Mgr Man: Tuning*, 14–1
TTY_DEFCHAR, *Sys Mgr Man: Essentials*, 7–10
TTY_DEFCHAR2, *Sys Mgr Man: Essentials*, 7–10, 7–12
types of, *Sys Mgr Man: Tuning*, 14–2
dynamic, *Sys Mgr Man: Tuning*, 14–2
general, *Sys Mgr Man: Tuning*, 14–2
major, *Sys Mgr Man: Tuning*, 14–2
UAFALTERNATE, *Sys Mgr Man: Essentials*, 4–10
user definable, *Sys Mgr Man: Tuning*, 14–3
VAXVMSSYS.PAR file (VAX), *Sys Mgr Man: Tuning*, 14–3
vector processing, *Sys Mgr Man: Tuning*, 26–5
VECTOR_MARGIN, *Sys Mgr Man: Tuning*, 26–7
VECTOR_PROC, *Sys Mgr Man: Tuning*, 26–5
VIRTUALPAGECNT, *Sys Mgr Man: Essentials*, 13–32
when incorrect values prevent the system from booting, *Sys Mgr Man: Essentials*, 4–8
WSMAX, *Sys Mgr Man: Essentials*, 13–30; *Sys Mgr Man: Tuning*, 26–7
System passwords, *Sys Mgr Man: Essentials*, 11–3
dictionary of, *Sys Mgr Man: Essentials*, 11–2
System shutdown
See also SHUTDOWN.COM command procedure
adjusting quorum when shutting down a node, *Sys Mgr Man: Essentials*, 4–29
after software installation, *Sys Mgr Man: Essentials*, 3–13
allowing batch and print jobs to complete before, *Sys Mgr Man: Essentials*, 13–55
caution about timing of system halt, *Sys Mgr Man: Tuning*, 15–2

System shutdown (cont'd)
checking for existence of system files before, *Sys Mgr Man: Essentials*, 4–29
customizing, *Sys Mgr Man: Essentials*, 4–32
with SYSHUTDWN.COM command procedure, *Sys Mgr Man: Essentials*, 4–27
defining the minimum number of minutes before shutdown, *Sys Mgr Man: Essentials*, 4–33
emergency procedure
OPCCRASH, *Sys Mgr Man: Essentials*, 4–27, 4–36
emergency procedures
console, *Sys Mgr Man: Essentials*, 4–27
for an entire VMScluster, *Sys Mgr Man: Essentials*, 4–29
notification of, *Sys Mgr Man: Essentials*, 4–33
options
automatic reboot, *Sys Mgr Man: Essentials*, 4–29
manual reboot, *Sys Mgr Man: Essentials*, 4–29
specifying time interval between DISABLE AUTOSTART/QUEUES and STOP /QUEUES/ON_NODE commands, *Sys Mgr Man: Essentials*, 13–56
time of shutdown, *Sys Mgr Man: Essentials*, 4–33
orderly, *Sys Mgr Man: Essentials*, 4–27
order of events, *Sys Mgr Man: Essentials*, 4–31
procedures for performing, *Sys Mgr Man: Essentials*, 4–27
saving AUTOGEN feedback data, *Sys Mgr Man: Essentials*, 4–29
SHUTDOWN.COM, *Sys Mgr Man: Essentials*, 4–27
example, *Sys Mgr Man: Essentials*, 4–30
how to use, *Sys Mgr Man: Essentials*, 4–28
when to use, *Sys Mgr Man: Essentials*, 4–27
stopping queues before, *Sys Mgr Man: Essentials*, 13–55
with SYSMAN, *Sys Mgr Man: Essentials*, 4–34
System startup
See also Booting
See also Site-specific startup command procedure
See also Startup command procedure
See also Startup phases
analyzing a crash dump, *Sys Mgr Man: Essentials*, 5–13
assigning systemwide logical names, *Sys Mgr Man: Essentials*, 5–8
booting with minimum, *Sys Mgr Man: Essentials*, 4–14
System startup (cont'd)
CONFIGURE phase, *Sys Mgr Man: Essentials*, 7–7
configuring devices, *Sys Mgr Man: Essentials*, 4–4, 7–6
special (Alpha), *Sys Mgr Man: Essentials*, 5–7
special (VAX), *Sys Mgr Man: Essentials*, 5–7
controlling when booting, *Sys Mgr Man: Essentials*, 4–12
creating systemwide announcements, *Sys Mgr Man: Essentials*, 5–14
databases, *Sys Mgr Man: Essentials*, 5–18
defining location of systemwide login procedure, *Sys Mgr Man: Essentials*, 5–17
definition of logical names, *Sys Mgr Man: Essentials*, 5–4, 5–5
description, *Sys Mgr Man: Essentials*, 4–12
determining order of phases, *Sys Mgr Man: Essentials*, 5–18
displaying startup commands as they execute, *Sys Mgr Man: Essentials*, 4–14
enabling autostart, *Sys Mgr Man: Essentials*, 5–12
enabling operator console, *Sys Mgr Man: Essentials*, 5–5
enabling operator log file, *Sys Mgr Man: Essentials*, 5–5
events, *Sys Mgr Man: Essentials*, 4–4
order of, *Sys Mgr Man: Essentials*, 5–4
possibility of future change in order, *Sys Mgr Man: Essentials*, 5–5
execution of AUTOCONFIGURE command, *Sys Mgr Man: Essentials*, 5–4
execution of login procedures, *Sys Mgr Man: Essentials*, 5–16
execution of site-specific startup command procedures, *Sys Mgr Man: Essentials*, 5–5
freeing dump information from page file, *Sys Mgr Man: Tuning*, 15–4
in an emergency
with default system parameters, *Sys Mgr Man: Essentials*, 4–8
without startup and login procedures, *Sys Mgr Man: Essentials*, 4–8
without the UAF, *Sys Mgr Man: Essentials*, 4–10
installing images, *Sys Mgr Man: Essentials*, 5–4; *Sys Mgr Man: Tuning*, 16–10
installing page and swap files, *Sys Mgr Man: Essentials*, 5–5, 5–6; *Sys Mgr Man: Tuning*, 15–5, 15–20, 15–21
limiting the number of interactive users, *Sys Mgr Man: Essentials*, 5–16
LMF database, *Sys Mgr Man: Essentials*, 5–5
loading of device drivers, *Sys Mgr Man: Essentials*, 5–4

System startup (cont'd)
loading of licenses, *Sys Mgr Man: Essentials*, 5–5
loading of Product Authorization Keys (PAKs), *Sys Mgr Man: Essentials*, 5–5
location of files used in, *Sys Mgr Man: Essentials*, 4–4
logging with SYSMAN, *Sys Mgr Man: Essentials*, 4–15
making remote InfoServer devices available, *Sys Mgr Man: Tuning*, 23–14
making remote InfoServer disks available, *Sys Mgr Man: Essentials*, 5–13
managing with SYSMAN, *Sys Mgr Man: Essentials*, 5–18
messages
indicating execution of site-independent startup, *Sys Mgr Man: Essentials*, 4–5
indicating execution of site-specific startup, *Sys Mgr Man: Essentials*, 4–5
indicating lack of installed page file, *Sys Mgr Man: Essentials*, 5–6
mounting disk for page and swap files, *Sys Mgr Man: Essentials*, 5–6
mounting the queue database disk, *Sys Mgr Man: Essentials*, 12–6
performing site-specific operations, *Sys Mgr Man: Essentials*, 5–10
phases
See Startup phases
purging the operator log file, *Sys Mgr Man: Essentials*, 5–14
saving contents of system dump file, *Sys Mgr Man: Essentials*, 5–13; *Sys Mgr Man: Tuning*, 15–15
setting
device characteristics in, *Sys Mgr Man: Essentials*, 5–12
printer device characteristics, *Sys Mgr Man: Essentials*, 7–14
terminal device characteristics, *Sys Mgr Man: Essentials*, 7–11
setting up a LAT network, *Sys Mgr Man: Essentials*, 5–15
starting InfoServer Client for OpenVMS software, *Sys Mgr Man: Essentials*, 5–13
starting of system processes, *Sys Mgr Man: Essentials*, 5–5
starting queues, *Sys Mgr Man: Essentials*, 5–12
starting SMISERVER process, *Sys Mgr Man: Essentials*, 5–5
starting the DECnet network, *Sys Mgr Man: Essentials*, 5–15
starting the License Management Facility (LMF), *Sys Mgr Man: Essentials*, 5–5
starting the queue manager, *Sys Mgr Man: Essentials*, 12–3
System startup (cont'd)
startup command procedures, *Sys Mgr Man: Essentials*, 5–2
startup of CONFIGURE process, *Sys Mgr Man: Essentials*, 5–5
submitting batch jobs, *Sys Mgr Man: Essentials*, 5–14
suppressing autoconfiguration, *Sys Mgr Man: Essentials*, 5–8, 7–9
tasks, *Sys Mgr Man: Essentials*, 4–1
VMS$PHASES.DAT database, *Sys Mgr Man: Essentials*, 5–4
System time
See Time
resetting after January 1st, *Sys Mgr Man: Tuning*, 20–18
System tuning
See Tuning
System users (security category)
defining with MAXSYSGROUP parameter, *Sys Mgr Man: Essentials*, 11–8
qualifications for, *Sys Mgr Man: Essentials*, 11–8
System version
registering images with dependencies on, *Sys Mgr Man: Essentials*, 5–22
System volumes
definition, *Sys Mgr Man: Essentials*, 8–8
Systemwide logical names, *Sys Mgr Man: Essentials*, 5–8
SYSTEST account
initial modification, *Sys Mgr Man: Essentials*, 6–9
in UAF, *Sys Mgr Man: Essentials*, 6–7
logging into for UETP, *Sys Mgr Man: Tuning*, 17–2, 17–4
privileges required for UETP, *Sys Mgr Man: Tuning*, 17–23
quotas required to run UETP, *Sys Mgr Man: Tuning*, 17–23
SYSTEST directory
creating for UETP, *Sys Mgr Man: Tuning*, 17–6
function during UETP, *Sys Mgr Man: Tuning*, 17–4
SYSTEST_CLIG account
in UAF, *Sys Mgr Man: Essentials*, 6–7
reenabling for UETP, *Sys Mgr Man: Tuning*, 17–10
requirements for UETP, *Sys Mgr Man: Tuning*, 17–10, 17–37
SYSUAF.DAT files
definition, *Sys Mgr Man: Essentials*, 6–2
moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8

SYSUAFALT.DAT file, *Sys Mgr Man: Essentials*, 4–10
SYSUAF logical name
defining during system startup, *Sys Mgr Man: Essentials*, 5–8
defining to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8

T

Tailoring a system disk
with VMSTAILOR and DECW$TAILOR, *Sys Mgr Man: Essentials*, 5–1
Tailoring utilities (VMSTAILOR and DECW$TAILOR), *Sys Mgr Man: Essentials*, 5–1
Tape cartridge drives
preparing for UETP, *Sys Mgr Man: Tuning*, 17–7
Tape commands
DISMOUNT, *Sys Mgr Man: Essentials*, 8–44
MOUNT, *Sys Mgr Man: Essentials*, 8–18
Tape files
See also Tape file system
accessing at file level, *Sys Mgr Man: Essentials*, 9–13
accessing for read and write operations, *Sys Mgr Man: Essentials*, 9–14
append operation, *Sys Mgr Man: Essentials*, 9–18
closing after opening for read access, *Sys Mgr Man: Essentials*, 9–17
closing after opening for write access, *Sys Mgr Man: Essentials*, 9–18
copying, *Sys Mgr Man: Essentials*, 9–21
definition, *Sys Mgr Man: Essentials*, 8–7
locating for read and write access, *Sys Mgr Man: Essentials*, 9–15
modifying characteristics, *Sys Mgr Man: Essentials*, 9–8
reading, *Sys Mgr Man: Essentials*, 9–17
update operation, *Sys Mgr Man: Essentials*, 9–18
writing to, *Sys Mgr Man: Essentials*, 9–18
Tape file system
checking
continuation volume, *Sys Mgr Man: Essentials*, 8–42
expiration date field, *Sys Mgr Man: Essentials*, 9–18
locating files, *Sys Mgr Man: Essentials*, 9–15
overwriting existing information, *Sys Mgr Man: Essentials*, 9–17
protection on, *Sys Mgr Man: Essentials*, 8–19
writing files to tape volume, *Sys Mgr Man: Essentials*, 9–18
Tape mass storage control protocol
See TMSCP
Tapes
See also Tape volumes
accessing file level, *Sys Mgr Man: Essentials*, 9–14
allocating drives, *Sys Mgr Man: Essentials*, 8–9, 9–22
basic concepts of, *Sys Mgr Man: Essentials*, 8–6
blocks, *Sys Mgr Man: Essentials*, 8–6
bpi, *Sys Mgr Man: Essentials*, 8–6
commands, *Sys Mgr Man: Essentials*, 8–27
copying files from, *Sys Mgr Man: Essentials*, 9–21
deallocating drives, *Sys Mgr Man: Essentials*, 8–10
dismounting, *Sys Mgr Man: Essentials*, 10–16
DOS–11, *Sys Mgr Man: Essentials*, 9–24
enabling write cache, *Sys Mgr Man: Essentials*, 8–24
file names, *Sys Mgr Man: Essentials*, 9–16
file protection
See Protection
files
See Tape files
file system, *Sys Mgr Man: Essentials*, 8–7
getting information about, *Sys Mgr Man: Essentials*, 7–17
initializing, *Sys Mgr Man: Essentials*, 10–13
IRG (interrecord gap), *Sys Mgr Man: Essentials*, 8–6
label format, *Sys Mgr Man: Essentials*, 8–24
labeling, *Sys Mgr Man: Essentials*, 10–21
loading on drive, *Sys Mgr Man: Essentials*, 9–22
management
tasks, *Sys Mgr Man: Essentials*, 7–16
markers on, *Sys Mgr Man: Essentials*, 8–6
modifying device characteristics, *Sys Mgr Man: Essentials*, 7–17, 8–42
mounting, *Sys Mgr Man: Essentials*, 10–15
MTACP process, *Sys Mgr Man: Essentials*, 8–6
preparing for UETP, *Sys Mgr Man: Tuning*, 17–2, 17–5, 17–6
protection, *Sys Mgr Man: Essentials*, 8–14
reading from, *Sys Mgr Man: Essentials*, 9–17
record blocking, *Sys Mgr Man: Essentials*, 8–6
advantages, *Sys Mgr Man: Essentials*, 8–7
record size
specifying, *Sys Mgr Man: Essentials*, 8–26
sequential organization of, *Sys Mgr Man: Essentials*, 8–6
specifying block size, *Sys Mgr Man: Essentials*, 8–24
standard-labeled
mounting, *Sys Mgr Man: Essentials*, 8–24

Tapes (cont'd)
structure of, *Sys Mgr Man: Essentials*, 8–6
testing with UETP, *Sys Mgr Man: Tuning*, 17–31, 17–33
UETP test image, *Sys Mgr Man: Tuning*, 17–34
volume label
overriding protection, *Sys Mgr Man: Essentials*, 8–25
volume protection
See Protection
volumes
See also Tape volumes
volume sets, *Sys Mgr Man: Essentials*, 8–6
write cache
enabling, *Sys Mgr Man: Essentials*, 8–24
writing files to, *Sys Mgr Man: Essentials*, 9–22
Tape volumes, *Sys Mgr Man: Essentials*, 9–23
See also Disk volumes
See also Tape files
See also Tapes
See also Tape volumes
See also Volumes
accessibility protection, *Sys Mgr Man: Essentials*, 8–19
accessing files on, *Sys Mgr Man: Essentials*, 9–13, 9–17
access to, *Sys Mgr Man: Essentials*, 9–14
continuation, *Sys Mgr Man: Essentials*, 8–39
copying files, *Sys Mgr Man: Essentials*, 9–21
from, *Sys Mgr Man: Essentials*, 9–21
to, *Sys Mgr Man: Essentials*, 9–20
to and from, *Sys Mgr Man: Essentials*, 9–24
deallocating, *Sys Mgr Man: Essentials*, 9–23
dismounting, *Sys Mgr Man: Essentials*, 9–23
file-structured, *Sys Mgr Man: Essentials*, 8–20
foreign, *Sys Mgr Man: Essentials*, 8–20
header labels, *Sys Mgr Man: Essentials*, 8–24
initializing, *Sys Mgr Man: Essentials*, 9–22
mounting, *Sys Mgr Man: Essentials*, 8–21, 8–24, 8–27
mounting volume sets, *Sys Mgr Man: Essentials*, 8–37
mounting with automatic switching disabled, *Sys Mgr Man: Essentials*, 8–40
mount verification, *Sys Mgr Man: Essentials*, 8–58
overriding UIC, *Sys Mgr Man: Essentials*, 8–25
private, *Sys Mgr Man: Essentials*, 8–9
reading attributes of header labels, *Sys Mgr Man: Essentials*, 9–17
reading files on, *Sys Mgr Man: Essentials*, 9–17
searching for files on, *Sys Mgr Man: Essentials*, 9–5
Tape volumes (cont'd)
specifying record size, *Sys Mgr Man: Essentials*, 8–26
standard-labeled, *Sys Mgr Man: Essentials*, 9–21
copying files from, *Sys Mgr Man: Essentials*, 9–21
wildcard characters supported, *Sys Mgr Man: Essentials*, 9–16
write-enabling, *Sys Mgr Man: Essentials*, 8–28
write-locking, *Sys Mgr Man: Essentials*, 8–37
write rings, *Sys Mgr Man: Essentials*, 8–37
writing files to, *Sys Mgr Man: Essentials*, 9–22
writing to files on, *Sys Mgr Man: Essentials*, 9–18
TAPE_MVTIMEOUT system parameter, *Sys Mgr Man: Essentials*, 8–60, 8–62
TCP/IP (Transmission Control Protocol/Internet Protocol)
applications, *Sys Mgr Man: Tuning*, 21–13
TDF (time differential factor), *Sys Mgr Man: Essentials*, 5–35
map for determining, *Sys Mgr Man: Essentials*, 5–35
tables of, *Sys Mgr Man: Tuning*, B–1
TELNET, *Sys Mgr Man: Tuning*, 21–13
Template files
for site-specific startup, *Sys Mgr Man: Essentials*, 5–3
Temporary working directory
specifying alternate working device for, *Sys Mgr Man: Essentials*, 3–14
Terminal queues, *Sys Mgr Man: Essentials*, 13–3
Terminals
console, *Sys Mgr Man: Essentials*, 2–21
controlling access through system passwords, *Sys Mgr Man: Essentials*, 11–3
determining physical type, *Sys Mgr Man: Essentials*, 7–13
disconnecting without terminating a process, *Sys Mgr Man: Essentials*, 7–11
documenting owners of, *Sys Mgr Man: Essentials*, 7–11
for security alarms, *Sys Mgr Man: Tuning*, 18–29
keeping sessions on multiple, *Sys Mgr Man: Essentials*, 7–11
LAT, *Sys Mgr Man: Essentials*, 7–13
determining characteristics of, *Sys Mgr Man: Essentials*, 7–13
disconnecting, *Sys Mgr Man: Essentials*, 7–12
managing
tasks for, *Sys Mgr Man: Essentials*, 7–10
operator
See Operator terminals
preparing for UETP, *Sys Mgr Man: Tuning*, 17–3, 17–5, 17–8, 17–14

Terminals (cont'd)
remote, *Sys Mgr Man: Essentials*, 7–12
SET TERMINAL/INQUIRE command, *Sys Mgr Man: Essentials*, 7–13
setting characteristics, *Sys Mgr Man: Essentials*, 7–10
default values, *Sys Mgr Man: Essentials*, 7–10
in system startup, *Sys Mgr Man: Essentials*, 5–12, 7–11
simulating users for UETP, *Sys Mgr Man: Tuning*, 17–34
testing with UETP, *Sys Mgr Man: Tuning*, 17–31, 17–33
UETP output, *Sys Mgr Man: Tuning*, 17–33
UETP test image, *Sys Mgr Man: Tuning*, 17–34
virtual
See Virtual terminals
Terminal servers, *Sys Mgr Man: Tuning*, 24–5
defined, *Sys Mgr Man: Tuning*, 24–1
on OpenVMS system, *Sys Mgr Man: Tuning*, 24–14
Time
See also TDF (time differential factor)
modifying system time, *Sys Mgr Man: Tuning*, 20–17
resetting after January 1st, *Sys Mgr Man: Tuning*, 20–18
updating in a VMScluster, *Sys Mgr Man: Tuning*, 20–18
Time differential factor
See TDF
Time differential factor (TDF)
See also Time
Time formats, *Sys Mgr Man: Essentials*, 5–43
predefined, *Sys Mgr Man: Essentials*, 5–48
specifying, *Sys Mgr Man: Essentials*, 5–46
Timeout periods
in SYSMAN, *Sys Mgr Man: Essentials*, 2–18
mount verification OPCOM message, *Sys Mgr Man: Essentials*, 8–60
Timer queue entry limit, *Sys Mgr Man: Essentials*, 6–44
TLZ04 tape drives
supported by UETP, *Sys Mgr Man: Tuning*, 17–7
TMSCP (tape mass storage control protocol), *Sys Mgr Man: Tuning*, 20–2
TN3270, *Sys Mgr Man: Tuning*, 21–13
TP_SERVER process
stopping permanently, *Sys Mgr Man: Tuning*, 25–20
stopping to dismount a disk, *Sys Mgr Man: Tuning*, 25–14
TQELM process limit, *Sys Mgr Man: Essentials*, 6–44
Track
definition, *Sys Mgr Man: Tuning*, A–2
Trailer labels
on tape files, *Sys Mgr Man: Essentials*, 8–7
Trailer pages, *Sys Mgr Man: Essentials*, 13–33
file, *Sys Mgr Man: Essentials*, 13–37
job, *Sys Mgr Man: Essentials*, 13–37
Transaction group
example, *Sys Mgr Man: Tuning*, 25–22
Transaction logs
changing the size of, *Sys Mgr Man: Tuning*, 25–9
checking the size of, *Sys Mgr Man: Tuning*, 25–8
creating, *Sys Mgr Man: Tuning*, 25–4
moving, *Sys Mgr Man: Tuning*, 25–11
planning, *Sys Mgr Man: Tuning*, 25–3
Transactions
monitoring, *Sys Mgr Man: Tuning*, 25–6
Transferring files remotely, *Sys Mgr Man: Tuning*, 21–14
Translation modes
card reader, *Sys Mgr Man: Essentials*, 7–19
Transmission Control Protocol/Internet Protocol
See TCP/IP
Transmission speed
setting for terminals, *Sys Mgr Man: Essentials*, 7–10, 7–11
Transports, *Sys Mgr Man: Tuning*, 23–6
LASTport, *Sys Mgr Man: Tuning*, 23–5
Trigger boot
MOP downline load service, *Sys Mgr Man: Tuning*, 22–27
Troubleshooting
adding or deleting a device control library module, *Sys Mgr Man: Essentials*, 13–83
autostart queues, *Sys Mgr Man: Essentials*, 13–81
booting problems, *Sys Mgr Man: Essentials*, 4–14, 4–16
general printer problems, *Sys Mgr Man: Essentials*, 13–77
holding jobs, *Sys Mgr Man: Essentials*, 13–78
if a device is not recognized by the system, *Sys Mgr Man: Essentials*, 7–7
jobs that will not execute, *Sys Mgr Man: Essentials*, 13–78
jobs with characteristic mismatch, *Sys Mgr Man: Essentials*, 13–80
OPCOM problems, *Sys Mgr Man: Essentials*, 2–22
pending jobs, *Sys Mgr Man: Essentials*, 13–78
print jobs with stock mismatch, *Sys Mgr Man: Essentials*, 13–80

Troubleshooting (cont'd)
problems deleting a queue, form, or characteristic, *Sys Mgr Man: Essentials*, 13–81
queue manager, *Sys Mgr Man: Essentials*, 12–14
queue problems, *Sys Mgr Man: Essentials*, 13–77
stalled output queue, *Sys Mgr Man: Essentials*, 13–81
startup problems, *Sys Mgr Man: Essentials*, 4–14
system dump file for, *Sys Mgr Man: Tuning*, 15–11
system failure, *Sys Mgr Man: Tuning*, 15–11
system hang, *Sys Mgr Man: Tuning*, 15–16
system startup problems, *Sys Mgr Man: Essentials*, 4–15
UETP, *Sys Mgr Man: Tuning*, 17–21
TT2$M_DISCONNECT characteristic
enabling, *Sys Mgr Man: Essentials*, 7–12
relationship to TTY_DEFCHAR2 system parameter, *Sys Mgr Man: Essentials*, 7–12
setting up virtual terminals, *Sys Mgr Man: Essentials*, 7–12
TTDRIVER device driver
loading, *Sys Mgr Man: Essentials*, 7–11
TTY_DECCHAR system parameter, *Sys Mgr Man: Essentials*, 7–10
TTY_DEFCHAR2 system parameter, *Sys Mgr Man: Essentials*, 7–10
relationship to TT2Y$M_DISCONNECT characteristic, *Sys Mgr Man: Essentials*, 7–12
setting up virtual terminals, *Sys Mgr Man: Essentials*, 7–12
Tuning
See also Performance
considering hardware capacity, *Sys Mgr Man: Tuning*, 16–6
definition, *Sys Mgr Man: Tuning*, 16–4
evaluating success, *Sys Mgr Man: Tuning*, 16–6
minimizing with AUTOGEN feedback, *Sys Mgr Man: Tuning*, 14–10
predicting when required, *Sys Mgr Man: Tuning*, 16–5
with AUTOGEN, *Sys Mgr Man: Tuning*, 14–5
TYPE command
tape, *Sys Mgr Man: Essentials*, 9–17

U

UAFALTERNATE logical name, *Sys Mgr Man: Essentials*, 4–10
UAFALTERNATE system parameter, *Sys Mgr Man: Essentials*, 4–10
UAFs (user authorization files)
booting with alternate, *Sys Mgr Man: Essentials*, 4–10
checking quotas for software installation, *Sys Mgr Man: Essentials*, 3–7
description of, *Sys Mgr Man: Essentials*, 6–2
general maintenance, *Sys Mgr Man: Essentials*, 6–6
initial contents, *Sys Mgr Man: Essentials*, 6–6
initial modification, *Sys Mgr Man: Essentials*, 6–9
listing records in, *Sys Mgr Man: Essentials*, 6–23
logical name defining location, *Sys Mgr Man: Essentials*, 5–8
login check, *Sys Mgr Man: Essentials*, 6–5
modifying
user record, *Sys Mgr Man: Essentials*, 6–23
network proxy, *Sys Mgr Man: Essentials*, 6–35
records
creating multiple default, *Sys Mgr Man: Essentials*, 6–24
resource limits for VAX and Alpha, *Sys Mgr Man: Essentials*, 6–40
returning to the default, *Sys Mgr Man: Essentials*, 4–11
SYSMAN checks, *Sys Mgr Man: Essentials*, 2–16
SYSUAF.DAT, *Sys Mgr Man: Essentials*, 6–2
user priorities, *Sys Mgr Man: Essentials*, 6–31
UETCONT00.DAT file, *Sys Mgr Man: Tuning*, 17–31
creation of, *Sys Mgr Man: Tuning*, 17–31
UETINIDEV.DAT file, *Sys Mgr Man: Tuning*, 17–31, 17–32
creation of, *Sys Mgr Man: Tuning*, 17–31
format, *Sys Mgr Man: Tuning*, 17–32
UETININET.DAT file, *Sys Mgr Man: Tuning*, 17–36
UETINIT00.EXE image, *Sys Mgr Man: Tuning*, 17–31
UETINIT01.EXE image, *Sys Mgr Man: Tuning*, 17–21, 17–31
UETLOAD00.DAT file, *Sys Mgr Man: Tuning*, 17–34
UETNETS00.EXE file, *Sys Mgr Man: Tuning*, 17–36
UETP$NODE_ADDRESS logical name, *Sys Mgr Man: Tuning*, 17–12
UETP (User Environment Test Package)
aborting execution, *Sys Mgr Man: Tuning*, 17–15
compact disc drives supported by, *Sys Mgr Man: Tuning*, 17–7

UETP (User Environment Test Package) (cont'd)
DECnet installation defaults, *Sys Mgr Man: Tuning*, 17–30
description, *Sys Mgr Man: Tuning*, 17–1
displaying tests as they run, *Sys Mgr Man: Tuning*, 17–18
initialization phase, *Sys Mgr Man: Tuning*, 17–31
interpreting output, *Sys Mgr Man: Tuning*, 17–17
master command procedure, *Sys Mgr Man: Tuning*, 17–30
normal completion, *Sys Mgr Man: Tuning*, 17–15
optical disk drives supported by, *Sys Mgr Man: Tuning*, 17–7
organization, *Sys Mgr Man: Tuning*, 17–30
required privileges, *Sys Mgr Man: Tuning*, 17–23
required quotas, *Sys Mgr Man: Tuning*, 17–23
requirements for small disk systems, *Sys Mgr Man: Tuning*, 17–11
running all phases, *Sys Mgr Man: Tuning*, 17–3
running individual phase, *Sys Mgr Man: Tuning*, 17–13
running multiple passes, *Sys Mgr Man: Tuning*, 17–14, 17–20
starting, *Sys Mgr Man: Tuning*, 17–13
system resource requirements, *Sys Mgr Man: Tuning*, 17–2, 17–4
testing vector processors, *Sys Mgr Man: Tuning*, 17–12
TLZ04 tape drive, *Sys Mgr Man: Tuning*, 17–7
typical failures reported by, *Sys Mgr Man: Tuning*, 17–21
VAX Vector Instruction, *Sys Mgr Man: Tuning*, 17–12
UETP.COM procedure, *Sys Mgr Man: Tuning*, 17–30
terminating, *Sys Mgr Man: Tuning*, 17–15
UETP.LOG file, *Sys Mgr Man: Tuning*, 17–15, 17–20, 17–29, 17–35
UETPHAS00.EXE program, *Sys Mgr Man: Tuning*, 17–30, 17–31
UETUNAS00.EXE UETP test image, *Sys Mgr Man: Tuning*, 17–19
UFDs (user file directories)
included in MFD, *Sys Mgr Man: Tuning*, A–4
UIC associated with, *Sys Mgr Man: Tuning*, A–4
UICs (user identification codes), *Sys Mgr Man: Essentials*, 6–14
default protection
changing, *Sys Mgr Man: Essentials*, 9–7
directory protection, *Sys Mgr Man: Essentials*, 9–11
UICs (user identification codes) (cont'd)
disk usage kept in quota file, *Sys Mgr Man: Tuning*, A–9
for UETP, *Sys Mgr Man: Tuning*, 17–6
identifiers, *Sys Mgr Man: Essentials*, 11–10
interpreting, *Sys Mgr Man: Essentials*, 11–7
member number, *Sys Mgr Man: Essentials*, 6–18
overriding for tape volumes, *Sys Mgr Man: Essentials*, 8–25
protection
of queues, *Sys Mgr Man: Essentials*, 13–22
public volumes, *Sys Mgr Man: Essentials*, 8–15
[0,0], *Sys Mgr Man: Essentials*, 8–49
Update procedures
See also Mandatory updates
definition, *Sys Mgr Man: Essentials*, 3–5
restrictions, *Sys Mgr Man: Essentials*, 3–5
Upgrade procedures
and system version dependent applications, *Sys Mgr Man: Essentials*, 5–23
definition, *Sys Mgr Man: Essentials*, 3–5
on Alpha systems, *Sys Mgr Man: Essentials*, 3–2
POLYCENTER Software Installation utility, *Sys Mgr Man: Essentials*, 3–2, 3–29
restrictions, *Sys Mgr Man: Essentials*, 3–5
steps in, *Sys Mgr Man: Essentials*, 3–5
using with existing OpenVMS, *Sys Mgr Man: Essentials*, 3–2
Usage count
DIRECTORY/SIZE command, *Sys Mgr Man: Essentials*, 8–49
DISKQUOTA display, *Sys Mgr Man: Essentials*, 8–49
USE command
in conversational boot, *Sys Mgr Man: Essentials*, 4–7
in SYSGEN, *Sys Mgr Man: Tuning*, 14–32, 14–34
User accounts
changing quotas or privileges, *Sys Mgr Man: Essentials*, 6–23
deleting, *Sys Mgr Man: Essentials*, 6–25
disabling, *Sys Mgr Man: Essentials*, 6–27
listing records of, *Sys Mgr Man: Essentials*, 6–23
maintaining, *Sys Mgr Man: Essentials*, 6–24
modifying, *Sys Mgr Man: Essentials*, 6–23
restricting use, *Sys Mgr Man: Essentials*, 6–27
setting up, *Sys Mgr Man: Essentials*, 6–6
User authorization, *Sys Mgr Man: Essentials*, 6–4
User authorization files
See UAFs
User disks
preparing for UETP, *Sys Mgr Man: Tuning*, 17–2, 17–6

User disks (cont'd)
space requirements for UETP, *Sys Mgr Man: Tuning*, 17–5
test error during UETP, *Sys Mgr Man: Tuning*, 17–24
testing with UETP, *Sys Mgr Man: Tuning*, 17–33
UETP test image, *Sys Mgr Man: Tuning*, 17–34
User Environment Test Package
See UETP
User file directories
See UFDs
User files
on public volumes, *Sys Mgr Man: Essentials*, 8–8
placement, *Sys Mgr Man: Essentials*, 8–9
User Identification Code
See UIC
User loads
defined for UETP DECnet test, *Sys Mgr Man: Tuning*, 17–36
defining for the UETP load test, *Sys Mgr Man: Tuning*, 17–14
equation used to determine for UETP load test, *Sys Mgr Man: Tuning*, 17–18, 17–19
User Mail profile, *Sys Mgr Man: Essentials*, 6–38
User mode
logical names, *Sys Mgr Man: Tuning*, 16–15
User names
as identifiers, *Sys Mgr Man: Essentials*, 11–10
entering when logging in, *Sys Mgr Man: Essentials*, 2–10
User resources, *Sys Mgr Man: Essentials*, 6–39
Users
interactive
limiting number of, *Sys Mgr Man: Essentials*, 5–16
limiting the number of interactive, *Sys Mgr Man: Essentials*, 5–16; *Sys Mgr Man: Tuning*, 16–4
protection code categories, *Sys Mgr Man: Essentials*, 11–8
restricting login hours for, *Sys Mgr Man: Tuning*, 16–4
security categories of, *Sys Mgr Man: Essentials*, 11–8
sending messages to with OPCOM, *Sys Mgr Man: Essentials*, 2–22
sending requests to an operator, *Sys Mgr Man: Essentials*, 2–25
validation of, *Sys Mgr Man: Essentials*, 2–16
User-specified job retention
changing, *Sys Mgr Man: Essentials*, 13–73

V

Validation of users, *Sys Mgr Man: Essentials*, 2–16
VAXcluster environments
defining the number of nodes for AUTOGEN, *Sys Mgr Man: Tuning*, 14–21
VAXcluster systems
adjusting quorum after shutting down a node, *Sys Mgr Man: Essentials*, 4–29
shutting down an entire VAXcluster, *Sys Mgr Man: Essentials*, 4–29
VAX systems
downline loading, *Sys Mgr Man: Tuning*, 23–2
VAX Vector Instruction Emulation facility (VVIEF), *Sys Mgr Man: Tuning*, 17–12
definition, *Sys Mgr Man: Tuning*, 26–5
determining presence of, *Sys Mgr Man: Tuning*, 26–9, 26–10
loading, *Sys Mgr Man: Tuning*, 26–10
unloading, *Sys Mgr Man: Tuning*, 26–10
VAXVMSSYS.PAR file, *Sys Mgr Man: Essentials*, 4–2; *Sys Mgr Man: Tuning*, 14–3
initializing parameters at boot time, *Sys Mgr Man: Tuning*, 14–36
VCRs (vector count registers), *Sys Mgr Man: Tuning*, 26–4
Vector capability
determining availability within a system, *Sys Mgr Man: Tuning*, 26–9
placing an ACL on, *Sys Mgr Man: Tuning*, 26–8
Vector-capable, *Sys Mgr Man: Tuning*, 26–4
Vector consumer
determining the identity of, *Sys Mgr Man: Tuning*, 26–9
managing, *Sys Mgr Man: Tuning*, 26–6
marginal, *Sys Mgr Man: Tuning*, 26–7
obtaining information about, *Sys Mgr Man: Tuning*, 26–8
Vector context switch
obtaining information about, *Sys Mgr Man: Tuning*, 26–9
Vector CPU time
obtaining information regarding process, *Sys Mgr Man: Tuning*, 26–9
Vector-present processors, *Sys Mgr Man: Tuning*, 26–4
adding to system, *Sys Mgr Man: Tuning*, 26–5
identifying, *Sys Mgr Man: Tuning*, 26–9
removing from system, *Sys Mgr Man: Tuning*, 26–5
when unavailable, *Sys Mgr Man: Tuning*, 26–6
Vector processing, *Sys Mgr Man: Tuning*, 26–4 to 26–9
configuring system, *Sys Mgr Man: Tuning*, 26–5

Vector processing (cont'd)
definition, *Sys Mgr Man: Tuning*, 26–4
establishing batch queues for, *Sys Mgr Man: Tuning*, 26–7
managing, *Sys Mgr Man: Tuning*, 26–5
obtaining information about, *Sys Mgr Man: Tuning*, 26–8
obtaining number of vector processors, *Sys Mgr Man: Tuning*, 26–9
resource requirements, *Sys Mgr Man: Tuning*, 26–7
system performance, *Sys Mgr Man: Tuning*, 26–4
tasks for managing, *Sys Mgr Man: Tuning*, 26–5
tuning system, *Sys Mgr Man: Tuning*, 26–7
VAX support for, *Sys Mgr Man: Tuning*, 26–4
Vector registers, *Sys Mgr Man: Tuning*, 26–4
Vectors
definition, *Sys Mgr Man: Tuning*, 26–4
VECTOR_MARGIN system parameter, *Sys Mgr Man: Tuning*, 26–7
VECTOR_PROC system parameter, *Sys Mgr Man: Tuning*, 26–5
Verification
mount
See Mount verification
startup, *Sys Mgr Man: Essentials*, 4–15
turning on during system startup, *Sys Mgr Man: Essentials*, 4–14
Version dependencies
registering images, *Sys Mgr Man: Essentials*, 5–22
Version limits
setting on files, *Sys Mgr Man: Essentials*, 8–53
Version numbers, *Sys Mgr Man: Essentials*, 9–15
Virtual devices
mounting during system startup, *Sys Mgr Man: Tuning*, 23–14
VIRTUALPAGECNT system parameter
optimizing batch queues for Sort/Merge utility, *Sys Mgr Man: Essentials*, 13–32
Virtual terminal protocol, *Sys Mgr Man: Tuning*, 21–13
Virtual terminals
connecting, *Sys Mgr Man: Essentials*, 7–11
determining physical terminal type, *Sys Mgr Man: Essentials*, 7–13
enabling, *Sys Mgr Man: Essentials*, 7–11
purpose of, *Sys Mgr Man: Essentials*, 7–11
TT2$M_DISCONNECT characteristic, *Sys Mgr Man: Essentials*, 7–12
TTY_DEFCHAR2 system parameter, *Sys Mgr Man: Essentials*, 7–12
VLRs (vector length registers), *Sys Mgr Man: Tuning*, 26–4
VMB.EXE file, *Sys Mgr Man: Essentials*, 4–17
role in boot process, *Sys Mgr Man: Essentials*, 4–2
VMRs (vector mask registers), *Sys Mgr Man: Tuning*, 26–4
VMS$LAYERED.DAT file, *Sys Mgr Man: Essentials*, 5–19
function in startup procedure, *Sys Mgr Man: Essentials*, 5–18
VMS$PHASES.DAT file
in startup procedure, *Sys Mgr Man: Essentials*, 5–4, 5–18
VMS$VMS.DAT file
in startup procedure, *Sys Mgr Man: Essentials*, 5–18
VMScluster environments
dual-architecture
installing images, *Sys Mgr Man: Tuning*, 20–21
preparing for UETP, *Sys Mgr Man: Tuning*, 17–11
security audit log files in, *Sys Mgr Man: Tuning*, 18–31
test failure during UETP, *Sys Mgr Man: Tuning*, 17–26
VMScluster systems
See also VAXcluster environments
adjusting quorum after shutting down a node, *Sys Mgr Man: Essentials*, 4–29
autostart queues in, *Sys Mgr Man: Essentials*, 13–4
benefits of, *Sys Mgr Man: Tuning*, 20–1
common parameter files in, *Sys Mgr Man: Tuning*, 14–22
cross-architecture booting, *Sys Mgr Man: Tuning*, 22–23
defining in SYSMAN, *Sys Mgr Man: Essentials*, 2–15
defining location of queue database files, *Sys Mgr Man: Essentials*, 12–5
device names in, *Sys Mgr Man: Essentials*, 7–1
disks in, *Sys Mgr Man: Essentials*, 8–2
dismounting a volume, *Sys Mgr Man: Essentials*, 8–45
dual-architecture
installing images, *Sys Mgr Man: Tuning*, 20–21
executing commands in, *Sys Mgr Man: Essentials*, 2–11
generic batch queues in, *Sys Mgr Man: Essentials*, 13–7
generic output queues in, *Sys Mgr Man: Essentials*, 13–12
generic queues in, *Sys Mgr Man: Essentials*, 13–2
in network environment, *Sys Mgr Man: Tuning*, 21–6

VMScluster systems (cont'd)
LAN management enhancements, *Sys Mgr Man: Tuning*, 22–1
LAN MOP coexisting with DECnet MOP, *Sys Mgr Man: Tuning*, 22–20
limiting nodes that can run the queue manager, *Sys Mgr Man: Essentials*, 12–9
local and nonlocal, *Sys Mgr Man: Essentials*, 2–15
managing with SYSMAN, *Sys Mgr Man: Tuning*, 20–16 to 20–23
migrating from DECnet MOP to LAN MOP, *Sys Mgr Man: Tuning*, 22–20
monitoring with SHOW CLUSTER, *Sys Mgr Man: Tuning*, 20–4
mounting volumes in, *Sys Mgr Man: Essentials*, 8–21
moving the queue manager to another node, *Sys Mgr Man: Essentials*, 12–10
node restriction for a startup command procedure, *Sys Mgr Man: Essentials*, 5–19
print and batch queues in, *Sys Mgr Man: Tuning*, 20–2
remote monitoring limitation, *Sys Mgr Man: Tuning*, 18–44
security, *Sys Mgr Man: Tuning*, 20–16
setting up for LAN MOP, *Sys Mgr Man: Tuning*, 22–22
shared resources, *Sys Mgr Man: Tuning*, 20–1
shutting down an entire VMScluster, *Sys Mgr Man: Essentials*, 4–29
specifying location of queue database, *Sys Mgr Man: Essentials*, 12–6
specifying preferred order of queue manager nodes, *Sys Mgr Man: Essentials*, 12–9
SYSMAN management environment, *Sys Mgr Man: Essentials*, 2–12
system management, *Sys Mgr Man: Tuning*, 20–3
using CLUSTER_CONFIG.COM, *Sys Mgr Man: Tuning*, 20–2
using CLUSTER_CONFIG to set up LAN MOP, *Sys Mgr Man: Tuning*, 22–21
VMSINSTAL.COM command procedure
See also Installation procedures
See also Installing software
Alternate System Root option, *Sys Mgr Man: Essentials*, 3–18
restriction, *Sys Mgr Man: Essentials*, 3–18
Alternate Working Device option, *Sys Mgr Man: Essentials*, 3–14
answer file, *Sys Mgr Man: Essentials*, 3–14
Autoanswer option, *Sys Mgr Man: Essentials*, 3–14
BACKUP qualifiers, *Sys Mgr Man: Essentials*, 3–16
command line syntax, *Sys Mgr Man: Essentials*, 3–9
VMSINSTAL.COM command procedure (cont'd)
completing installation, *Sys Mgr Man: Essentials*, 3–13
correcting problems, *Sys Mgr Man: Essentials*, 3–9
creating new answer file, *Sys Mgr Man: Essentials*, 3–14
destination parameter, *Sys Mgr Man: Essentials*, 3–12
File Log option, *Sys Mgr Man: Essentials*, 3–17
Get Save Set option, *Sys Mgr Man: Essentials*, 3–15, 3–16
getting help in, *Sys Mgr Man: Essentials*, 3–9
installing layered products, *Sys Mgr Man: Essentials*, 3–6
options, *Sys Mgr Man: Essentials*, 3–13
option list parameter, *Sys Mgr Man: Essentials*, 3–12
specifying, *Sys Mgr Man: Essentials*, 3–12
table of, *Sys Mgr Man: Essentials*, 3–11
preparing to use, *Sys Mgr Man: Essentials*, 3–6
product list parameter, *Sys Mgr Man: Essentials*, 3–9
product save-set format, *Sys Mgr Man: Essentials*, 3–16
Release Notes option, *Sys Mgr Man: Essentials*, 3–17
saving answers, *Sys Mgr Man: Essentials*, 3–14
source parameter, *Sys Mgr Man: Essentials*, 3–11
starting, *Sys Mgr Man: Essentials*, 3–9
system failure
conditions for, *Sys Mgr Man: Essentials*, 3–13
system shutdown following, *Sys Mgr Man: Essentials*, 3–13
temporary working directory
specifying location of, *Sys Mgr Man: Essentials*, 3–14
VMSKITBLD.COM command procedure
ADD option, *Sys Mgr Man: Essentials*, 2–31
BUILD option, *Sys Mgr Man: Essentials*, 2–27
compared to CLUSTER_CONFIG.COM command procedure, *Sys Mgr Man: Essentials*, 2–31
completing a system disk created with the BUILD option, *Sys Mgr Man: Essentials*, 2–29
configuring a system root added with, *Sys Mgr Man: Essentials*, 2–32
copying system files from the system disk, *Sys Mgr Man: Essentials*, 2–26
COPY option, *Sys Mgr Man: Essentials*, 2–29
options, *Sys Mgr Man: Essentials*, 2–26

VMSKITBLD.COM command procedure (cont'd)
reliance on .TEMPLATE version of site-specific command procedures, *Sys Mgr Man: Essentials*, 5–3
VMSMAIL logical name
defining to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
VMSMAIL_PROFILE.DATA file, *Sys Mgr Man: Essentials*, 6–38
moving to reduce system disk I/O, *Sys Mgr Man: Tuning*, 16–8
VMSMAIL_PROFILE logical name
defining during system startup, *Sys Mgr Man: Essentials*, 5–8
VMSTAILOR
See Tailoring utilities
Volatile databases
network, *Sys Mgr Man: Tuning*, 21–11
VOLPRO privilege, *Sys Mgr Man: Essentials*, 8–14
VOLSET.SYS file
See Volume set list file
Volume identifier field, *Sys Mgr Man: Essentials*, 8–39
Volume integrity, *Sys Mgr Man: Essentials*, 8–58
Volume labels
assigning to devices, *Sys Mgr Man: Essentials*, 5–11
changing, *Sys Mgr Man: Essentials*, 3–33
definition, *Sys Mgr Man: Essentials*, 10–55
format, *Sys Mgr Man: Essentials*, 8–24
specifying for magnetic tape, *Sys Mgr Man: Essentials*, 10–14
used with BACKUP command, *Sys Mgr Man: Essentials*, 10–55
Volume protection
See Protection, volume
Volume Protection Override (VOLPRO)
See VOLPRO privilege
Volumes
See also Disk volumes
See also Private volumes
See also Public volumes
See also Tape volumes
See also Volume sets
adding to an existing disk set, *Sys Mgr Man: Essentials*, 8–34
advantages of using separate, *Sys Mgr Man: Essentials*, 8–30
availability
OPCOM message, *Sys Mgr Man: Essentials*, 8–59
binding into volume set, *Sys Mgr Man: Essentials*, 8–22
canceling mount verification, *Sys Mgr Man: Essentials*, 8–62
continuation, *Sys Mgr Man: Essentials*, 8–42
Volumes (cont'd)
controlling cache size, *Sys Mgr Man: Essentials*, 8–58
disabling mount messages, *Sys Mgr Man: Essentials*, 8–23
disk
See Disk volumes
dismounting, *Sys Mgr Man: Essentials*, 8–43, 10–16
restriction, *Sys Mgr Man: Tuning*, 16–18
Files–11, *Sys Mgr Man: Essentials*, 8–3
foreign
copying files to and from, *Sys Mgr Man: Essentials*, 9–24
mounting, *Sys Mgr Man: Essentials*, 8–21, 9–13
group, *Sys Mgr Man: Essentials*, 8–8
initializing, *Sys Mgr Man: Essentials*, 8–10, 10–13
mounting, *Sys Mgr Man: Essentials*, 8–28, 10–15
See also MOUNT command
continuation tape, *Sys Mgr Man: Essentials*, 8–40
if device is unavailable, *Sys Mgr Man: Essentials*, 8–27
operator functions, *Sys Mgr Man: Essentials*, 8–18
public, *Sys Mgr Man: Essentials*, 8–20
steps, *Sys Mgr Man: Essentials*, 8–27
mount verification aborted
OPCOM message, *Sys Mgr Man: Essentials*, 8–63
mount verification timeout, *Sys Mgr Man: Essentials*, 8–60
OPCOM message, *Sys Mgr Man: Essentials*, 8–60
operator-assisted mount, *Sys Mgr Man: Essentials*, 8–28
private, *Sys Mgr Man: Essentials*, 8–9
protection, *Sys Mgr Man: Essentials*, 8–14
public
See Public volumes
rebuild
determining if needed, *Sys Mgr Man: Essentials*, 7–4
rebuilding, *Sys Mgr Man: Essentials*, 8–50
recovering from errors, *Sys Mgr Man: Essentials*, 8–60
sharing, *Sys Mgr Man: Essentials*, 8–23
substituting, *Sys Mgr Man: Essentials*, 8–27
system
See System volumes
tape
See Tape volumes

Volume security profile
reserved file, *Sys Mgr Man: Tuning*, A–9
SECURITY.SYS, *Sys Mgr Man: Tuning*, A–9
Volume set list file
definition, *Sys Mgr Man: Tuning*, A–9
reserved file, *Sys Mgr Man: Tuning*, A–9
used by Analyze/Disk_Structure utility, *Sys Mgr Man: Tuning*, A–9
VOLSET.SYS, *Sys Mgr Man: Tuning*, A–9
Volume sets
backing up, *Sys Mgr Man: Essentials*, 10–40
CD–ROM
partially mounted ISO 9660, *Sys Mgr Man: Essentials*, 8–35
characteristics, *Sys Mgr Man: Essentials*, 8–30
creating, *Sys Mgr Man: Essentials*, 8–22, 8–30
definition, *Sys Mgr Man: Essentials*, 8–2, 8–30
disk, *Sys Mgr Man: Essentials*, 8–31 to 8–34
accessing, *Sys Mgr Man: Essentials*, 8–31, 8–32
adding to, *Sys Mgr Man: Essentials*, 8–34
adding volumes, *Sys Mgr Man: Essentials*, 8–34
assigning name to, *Sys Mgr Man: Essentials*, 8–31
creating, *Sys Mgr Man: Essentials*, 8–31
from existing volumes, *Sys Mgr Man: Essentials*, 8–33
from new volumes, *Sys Mgr Man: Essentials*, 8–32
shadowed, *Sys Mgr Man: Essentials*, 8–33
creating files on, *Sys Mgr Man: Essentials*, 8–32
directory structure, *Sys Mgr Man: Essentials*, 8–31
index file, *Sys Mgr Man: Essentials*, 8–32
mounting, *Sys Mgr Man: Essentials*, 8–31, 8–32, 8–33
naming, *Sys Mgr Man: Essentials*, 8–31
information stored in VOLSET.SYS, *Sys Mgr Man: Tuning*, A–9
mounting, *Sys Mgr Man: Essentials*, 8–31
See also MOUNT command
privileges to access, *Sys Mgr Man: Essentials*, 8–32
processing continuation volumes, *Sys Mgr Man: Essentials*, 8–37
restoring, *Sys Mgr Man: Essentials*, 10–47
restriction for system disk, *Sys Mgr Man: Essentials*, 8–31
root volume, *Sys Mgr Man: Essentials*, 8–31
tape, *Sys Mgr Man: Essentials*, 8–6, 8–37 to 8–40
automatic volume switching, *Sys Mgr Man: Essentials*, 8–39
continuation volumes, *Sys Mgr Man: Essentials*, 8–38
Volume sets
tape (cont'd)
creating, *Sys Mgr Man: Essentials*, 8–37
mounting, *Sys Mgr Man: Essentials*, 8–37
mounting continuation volumes, *Sys Mgr Man: Essentials*, 8–40
mounting with automatic switching disabled, *Sys Mgr Man: Essentials*, 8–40
Volume shadowing, *Sys Mgr Man: Essentials*, 10–40, 10–47
minimerge (assisted merge), *Sys Mgr Man: Essentials*, 10–41
mounting disks in host-based shadow set, *Sys Mgr Man: Essentials*, 10–41
RAID, *Sys Mgr Man: Essentials*, 10–40
Volume structure
ISO 9660, *Sys Mgr Man: Essentials*, 8–3
Volume switching
automatic, *Sys Mgr Man: Essentials*, 8–39
VTA0 device
connecting, *Sys Mgr Man: Essentials*, 7–11
VVIEF
See VAX Vector Instruction Enulation facility
VVIEF$DEINSTAL.COM command procedure, *Sys Mgr Man: Tuning*, 26–10
VVIEF$INSTAL.COM command procedure, *Sys Mgr Man: Tuning*, 26–5, 26–10
VVIEF (VAX Vector Instruction Emulation facility)
See VAX Vector Instruction Emulation Facility

W

Waiting job status, *Sys Mgr Man: Essentials*, 13–69
WANs (wide area networks)
connections, *Sys Mgr Man: Tuning*, 21–8
multipoint, *Sys Mgr Man: Tuning*, 21–9
point-to-point, *Sys Mgr Man: Tuning*, 21–8
Welcome messages
defining, *Sys Mgr Man: Essentials*, 5–14
displaying, *Sys Mgr Man: Essentials*, 5–15
login, *Sys Mgr Man: Essentials*, 2–10
Wide area networks (WANs)
See WANs
Wildcard characters
asterisk (*)
using with tape volumes, *Sys Mgr Man: Essentials*, 9–16
in file names, *Sys Mgr Man: Essentials*, 9–15
with OpenVMS extended names, *Sys Mgr Man: Essentials*, 9–16
with standard names, *Sys Mgr Man: Essentials*, 9–16
with tape volumes, *Sys Mgr Man: Essentials*, 9–15

Working directory
temporary
in VMSINSTAL.COM command procedure, *Sys Mgr Man: Essentials*, 3–14
Working sets
adjusting for vectorized applications, *Sys Mgr Man: Tuning*, 26–7
default size, *Sys Mgr Man: Essentials*, 6–44
extent, *Sys Mgr Man: Essentials*, 6–44
limits and quotas
choosing a value for batch queues, *Sys Mgr Man: Essentials*, 13–30, 13–31
specifying for batch queues, *Sys Mgr Man: Essentials*, 13–21, 13–29
specifying for output queues, *Sys Mgr Man: Essentials*, 13–21
quota, *Sys Mgr Man: Essentials*, 6–44
Work load
adjusting system parameters for, *Sys Mgr Man: Tuning*, 16–5
distributing, *Sys Mgr Man: Tuning*, 16–3
designing efficient applications, *Sys Mgr Man: Tuning*, 16–4
restricting login hours, *Sys Mgr Man: Tuning*, 16–4
importance of knowing, *Sys Mgr Man: Tuning*, 16–2
monitoring, *Sys Mgr Man: Tuning*, 16–2
tools used for, *Sys Mgr Man: Tuning*, 16–2
strategies for managing, *Sys Mgr Man: Tuning*, 16–3
Workstations
backing up, *Sys Mgr Man: Essentials*, 10–36
managing queues on, *Sys Mgr Man: Essentials*, 13–2
OPCOM behavior on, *Sys Mgr Man: Essentials*, 2–21
OPCOM startup on, *Sys Mgr Man: Essentials*, 2–21
printer queue configuration, *Sys Mgr Man: Essentials*, 13–8
setting up media on, *Sys Mgr Man: Essentials*, 8–11
starting SMISERVER process, *Sys Mgr Man: Essentials*, 2–12
use of Snapshot facility with, *Sys Mgr Man: Essentials*, 4–21
World users (security category), *Sys Mgr Man: Essentials*, 11–8
Writable images, *Sys Mgr Man: Tuning*, 16–11
Write access
See Access types, write
for disk directory files, *Sys Mgr Man: Essentials*, 9–11
for disk files, *Sys Mgr Man: Essentials*, 9–6
granting through protection codes, *Sys Mgr Man: Essentials*, 11–8
Writeboot utility (WRITEBOOT), *Sys Mgr Man: Essentials*, 4–17
error messages, *Sys Mgr Man: Essentials*, 4–18
Write cache
enabling for tape device, *Sys Mgr Man: Essentials*, 8–24
Write-enabling a tape, *Sys Mgr Man: Essentials*, 8–28
Write-locked devices
mount verification, *Sys Mgr Man: Essentials*, 8–61
Write-locking
disk volumes, *Sys Mgr Man: Essentials*, 8–60
tape volumes, *Sys Mgr Man: Essentials*, 8–37
Write operation
See Access types, write
Write rings
on tape volumes, *Sys Mgr Man: Essentials*, 8–37
WSDEFAULT process limit
choosing a value for batch queues, *Sys Mgr Man: Essentials*, 13–30
specifying a value for batch queues, *Sys Mgr Man: Essentials*, 13–21, 13–29
specifying a value for output queues, *Sys Mgr Man: Essentials*, 13–21
specifying values, *Sys Mgr Man: Essentials*, 13–21 to 13–30
WSDEF process limit, *Sys Mgr Man: Essentials*, 6–44
WSEXTENT process limit, *Sys Mgr Man: Essentials*, 6–44
choosing a value for batch queues, *Sys Mgr Man: Essentials*, 13–30
for efficient sorting, *Sys Mgr Man: Essentials*, 13–31
specifying a value for batch queues, *Sys Mgr Man: Essentials*, 13–22, 13–29
specifying a value for output queues, *Sys Mgr Man: Essentials*, 13–22
value for efficient backups, *Sys Mgr Man: Essentials*, 10–10
WSMAX system parameter, *Sys Mgr Man: Essentials*, 13–30; *Sys Mgr Man: Tuning*, 26–7
WSQUO process limit, *Sys Mgr Man: Essentials*, 6–44
WSQUOTA process limit
choosing a value for batch queues, *Sys Mgr Man: Essentials*, 13–30
specifying a value for batch queues, *Sys Mgr Man: Essentials*, 13–22, 13–29
specifying a value for output queues, *Sys Mgr Man: Essentials*, 13–22
value for efficient backups, *Sys Mgr Man: Essentials*, 10–10

X

XAR keyword
with MOUNT/PROTECTION command, *Sys Mgr Man: Essentials*, 8–23
XARs (extended attribute records)
protection fields, *Sys Mgr Man: Essentials*, 8–17
mount option, *Sys Mgr Man: Essentials*, 8–17
XDELTA debugging tool
entering through IPC, *Sys Mgr Man: Essentials*, 8–63
X terminal client, *Sys Mgr Man: Tuning*, 23–4