

*OpenVMS System Troubleshooting  
for System Managers*

Instructor Guide

*Training*



**COMPAQ**



*OpenVMS System Troubleshooting  
for System Managers*

Instructor Guide

*Training*



**COMPAQ**

© 2000 Compaq Computer Corporation

COMPAQ and the Compaq logo, AXP, CI, DCL, DECevent, DECnet, DECserver 200, DECserver 50, DECserver 90Lplus, DECTPU, DECwindows, DSSI, FDDI, Files-11, HSC50, Memory Channel, Multivendor Customer Services, OpenVMS, OpenVMS Cluster, TPU, TU58 VAX, VAXport, VMS Record Management Services (VMS RMS) Registered in U.S. Patent and Trademark Office.

Microsoft, Windows, Windows NT, and Visual C++ are trademarks of Microsoft Corporation.

Intel, Pentium, Intel Inside, and Celeron are trademarks of Intel Corporation.

Motif, OSF/1, UNIX, the “X” Device, IT DialTone, and The Open Group are trademarks of The Open Group. All other product names mentioned herein may be trademarks or registered trademarks of their respective companies.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change without notice.

The information in this publication is subject to change without notice and is provided “AS IS” WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK ARISING OUT OF THE USE OF THIS INFORMATION REMAINS WITH RECIPIENT. IN NO EVENT SHALL COMPAQ BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, OR LOSS OF BUSINESS INFORMATION), EVEN IF COMPAQ HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE FOREGOING SHALL APPLY REGARDLESS OF THE NEGLIGENCE OR OTHER FAULT OF EITHER PARTY AND REGARDLESS OF WHETHER SUCH LIABILITY SOUNDS IN CONTRACT, NEGLIGENCE, TORT, OR ANY OTHER THEORY OF LEGAL LIABILITY, AND NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

The limited warranties for Compaq products are exclusively set forth in the documentation accompanying such products. Nothing herein should be construed as constituting a further or additional warranty.

## **OpenVMS System Troubleshooting for System Managers**

Instructor Guide

March 2000

## Overview

|                                 |    |
|---------------------------------|----|
| Introduction.....               | 2  |
| Course Description .....        | 3  |
| Intended Audience.....          | 3  |
| Prerequisites .....             | 3  |
| Course Goals .....              | 3  |
| Course Nongoes.....             | 4  |
| Course Resources.....           | 4  |
| Course Organization .....       | 5  |
| Module Organization .....       | 5  |
| Course Conventions.....         | 6  |
| Instructor Notes .....          | 7  |
| OpenVMS Alpha Coverage.....     | 8  |
| Course Schedule .....           | 8  |
| Student Package .....           | 9  |
| Instructor Package .....        | 9  |
| Student References .....        | 9  |
| Instructor Preparation .....    | 9  |
| Course Notes File.....          | 10 |
| Preparing the Systems.....      | 10 |
| Required Equipment .....        | 10 |
| Creating Student Accounts ..... | 11 |

## Module 1 — Dependable Systems Overview

|                                           |   |
|-------------------------------------------|---|
| Introduction.....                         | 2 |
| Objectives .....                          | 2 |
| Resources .....                           | 2 |
| Dependable Systems .....                  | 3 |
| Overview .....                            | 3 |
| System Dependability Categories .....     | 4 |
| Categories of Dependability.....          | 4 |
| Conventional Computing Systems.....       | 4 |
| Highly Available Computing Systems.....   | 5 |
| Mission-Critical Computing Systems .....  | 5 |
| Disaster-Tolerant Computing Systems ..... | 5 |
| System Problems .....                     | 6 |
| Classifying System Problems.....          | 6 |
| Environmental Problems.....               | 6 |
| Hardware Problems.....                    | 6 |
| Software Problems .....                   | 7 |
| Human Errors .....                        | 7 |
| Handling Problems.....                    | 8 |

|                                                          |    |
|----------------------------------------------------------|----|
| Troubleshooting Methodology .....                        | 9  |
| Overview .....                                           | 9  |
| Step 1: Define the Problem .....                         | 10 |
| Purpose.....                                             | 10 |
| What to Do.....                                          | 10 |
| Document the Process.....                                | 11 |
| Step 2: Gather Information.....                          | 12 |
| Purpose.....                                             | 12 |
| Relate Problem to System Changes .....                   | 12 |
| Questions to Ask.....                                    | 13 |
| Step 3: Analyze and Interpret Information.....           | 14 |
| Purpose.....                                             | 14 |
| Analyze the Information .....                            | 14 |
| Interpret the Information.....                           | 15 |
| Example: Interpreting Information .....                  | 16 |
| Step 4: Formulate and Apply a Proposed Solution .....    | 17 |
| Purpose.....                                             | 17 |
| What to Do .....                                         | 17 |
| Propose a Solution .....                                 | 17 |
| Investigate the Proposed Solution.....                   | 18 |
| Apply the Proposed Solution .....                        | 18 |
| Escalate the Problem.....                                | 18 |
| Temporary Solutions.....                                 | 18 |
| Step 5: Verify the Solution.....                         | 20 |
| Purpose.....                                             | 20 |
| What to Do .....                                         | 20 |
| Example: Verifying a Problem Solution.....               | 20 |
| Step 6: Clean Up.....                                    | 21 |
| Purpose.....                                             | 21 |
| What to Do .....                                         | 21 |
| Step 7: Reverify the Solution .....                      | 22 |
| Purpose.....                                             | 22 |
| What to Do .....                                         | 22 |
| Further Actions .....                                    | 22 |
| Step 8: Document the Problem.....                        | 23 |
| Purpose.....                                             | 23 |
| What to Do .....                                         | 23 |
| Step 9: Plan to Avoid or Handle Future Occurrences ..... | 24 |
| Purpose.....                                             | 24 |
| What to Do .....                                         | 24 |
| OpenVMS System Messages.....                             | 25 |
| Overview .....                                           | 25 |
| Format of OpenVMS System Messages .....                  | 26 |
| Example: Error Message .....                             | 26 |
| Severity Levels .....                                    | 27 |
| Accessing Message Descriptions from Online Help.....     | 27 |
| Example: Message Description.....                        | 28 |
| OpenVMS Troubleshooting Tools.....                       | 29 |

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Overview .....                                                    | 29 |
| Error Logging Facility .....                                      | 30 |
| Components .....                                                  | 30 |
| Events Recorded .....                                             | 30 |
| Hardware Types .....                                              | 31 |
| Error Log Buffers .....                                           | 31 |
| Error Log File .....                                              | 32 |
| Example: Displaying Device Errors .....                           | 33 |
| Error Log Utility Reporting .....                                 | 34 |
| Example: Display from DIAGNOSE/FULL (Alpha) .....                 | 35 |
| Operator Log File .....                                           | 37 |
| Location .....                                                    | 37 |
| Messages Recorded in the Operator Log File .....                  | 37 |
| Searching the Operator Log .....                                  | 37 |
| System Bugchecks .....                                            | 39 |
| Types of Bugchecks .....                                          | 39 |
| System Dump File .....                                            | 40 |
| Contents of the System Dump File .....                            | 40 |
| Uses for the System Dump File .....                               | 40 |
| Enabling the System Dump File .....                               | 40 |
| Example: Directory Listing of System Dump File .....              | 41 |
| Where Is the Dump File Written? .....                             | 42 |
| Saving the System Dump File .....                                 | 43 |
| SDA Commands at System Startup .....                              | 44 |
| Example: SDA Commands in System Startup Procedure .....           | 44 |
| Calculating the Correct Size for the System Dump File .....       | 45 |
| Capturing Dump Files on Systems with Limited Disk Space .....     | 46 |
| Using the Paging File for the System Dump .....                   | 46 |
| Copying the System Dump File from SYS\$SYSTEM: PAGEFILE.SYS ..... | 47 |
| Systems with Large Memory and Small Disk Space .....              | 48 |
| Using the System Dump Analyzer to Analyze a Crash .....           | 49 |
| Overview .....                                                    | 49 |
| Accessing the SDA (Alpha) .....                                   | 49 |
| Example: Displaying Crash Information .....                       | 49 |
| Example Notes: Displaying Crash Information .....                 | 50 |
| Example: SDA Command SHOW SUMMARY/IMAGE .....                     | 51 |
| Crash Log Utility Extractor .....                                 | 52 |
| Overview .....                                                    | 52 |
| Accessing CLUE .....                                              | 52 |
| Example: Accessing CLUE .....                                     | 52 |
| Displaying CLUE History Entries .....                             | 53 |
| Displaying Crash Information Using CLUE (OpenVMS VAX) .....       | 53 |
| Displaying Crash Information Using CLUE (Alpha) .....             | 54 |
| System Dump Analyzer Features .....                               | 57 |
| New Features for Existing Commands .....                          | 57 |
| Other SDA Command Changes .....                                   | 57 |
| New SDA Command SET FETCH .....                                   | 57 |
| SDA Display Changes .....                                         | 58 |

|                                     |    |
|-------------------------------------|----|
| VAX Dump File Process.....          | 58 |
| Summary .....                       | 59 |
| Concepts .....                      | 59 |
| Dependable Systems .....            | 59 |
| Troubleshooting Methodology .....   | 59 |
| OpenVMS System Messages .....       | 59 |
| OpenVMS Troubleshooting Tools ..... | 59 |
| Commands .....                      | 60 |
| OpenVMS System Messages .....       | 60 |
| OpenVMS Troubleshooting Tools.....  | 60 |
| Chapter Terms .....                 | 61 |

## **Module 2 — System Initialization Problems**

|                                                           |    |
|-----------------------------------------------------------|----|
| Introduction.....                                         | 2  |
| Objectives .....                                          | 2  |
| Resources .....                                           | 2  |
| System Initialization Concepts .....                      | 3  |
| Overview .....                                            | 3  |
| Phases of OpenVMS System Initialization .....             | 4  |
| Initialization Steps.....                                 | 4  |
| Sequence of Initialization Events .....                   | 7  |
| Site-Specific Startup Command Procedures .....            | 9  |
| Site-Specific Startup Dependencies .....                  | 10 |
| Emergency System Initialization.....                      | 11 |
| Overview .....                                            | 11 |
| Performing a Conversational Boot .....                    | 12 |
| When to Use Conversational Boot.....                      | 12 |
| Conversational Boot Options .....                         | 12 |
| Processor-Specific Procedures .....                       | 12 |
| Examples.....                                             | 12 |
| Using SYSGEN Commands During System Initialization.....   | 13 |
| SYSBOOT> Prompt.....                                      | 13 |
| SYSGEN Commands Commonly Used at the SYSBOOT> Prompt..... | 13 |
| Performing a Minimum System Initialization.....           | 14 |
| Minimum Initialization Operations.....                    | 14 |
| System Processes at Minimum Boot.....                     | 15 |
| Resetting STARTUP_P1.....                                 | 15 |
| Using the AUTOGEN.PAR File for System Parameters.....     | 15 |
| Using the Default System Parameters.....                  | 16 |
| Defining the Console as the Startup Procedure.....        | 17 |
| Overview.....                                             | 17 |
| Defining the Console as the Startup Procedure .....       | 17 |
| Restoring the System Startup.....                         | 17 |
| Using a Full-Screen Editor .....                          | 18 |
| Using EDT from the Console.....                           | 20 |
| Handling a Missing System Account Password.....           | 21 |
| Using the Full Backup Utility at OPA0: .....              | 23 |
| Using an Alternative Site-Independent Startup File.....   | 24 |

|                                                                    |    |
|--------------------------------------------------------------------|----|
| Displaying the Site-Independent Startup Command Procedure .....    | 25 |
| Using Standalone Backup to Save and Restore the System Disk .....  | 26 |
| Standalone Backup.....                                             | 26 |
| Booting Standalone Backup.....                                     | 26 |
| Backing Up the System Disk to Tape .....                           | 27 |
| Restoring the System Disk from Tape .....                          | 28 |
| Handling System Disk Problems.....                                 | 29 |
| Handling System Disk Hardware Problems.....                        | 29 |
| Software Problems .....                                            | 29 |
| Alternative Method for Handling System Disk Software Problems..... | 29 |
| Resolving System Booting Problems .....                            | 31 |
| Overview .....                                                     | 31 |
| Isolating and Resolving System Initialization Problems.....        | 32 |
| System Initialization Landmarks .....                              | 32 |
| When a System Does Not Initialize .....                            | 32 |
| Using Standalone Backup in an Emergency .....                      | 34 |
| Case Study: A Late Night Call.....                                 | 34 |
| Case Study Solution: A Late Night Call .....                       | 35 |
| Avoiding System Initialization Problems .....                      | 37 |
| Overview .....                                                     | 37 |
| Change Management Procedures .....                                 | 38 |
| Questions to Ask Before You Schedule Changes.....                  | 38 |
| Designing a Standard Procedure to Control Changes.....             | 39 |
| Training and Enforcement .....                                     | 39 |
| Summary.....                                                       | 40 |
| System Initialization Concepts .....                               | 40 |
| Emergency Initialization Procedures.....                           | 40 |
| Resolving System Booting Problems .....                            | 40 |
| Avoiding System Initialization Problems.....                       | 40 |
| Chapter Terms .....                                                | 41 |

## Module 3 — Troubleshooting Printing Problems

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| Introduction.....                                                     | 2  |
| Objectives .....                                                      | 2  |
| Resources .....                                                       | 2  |
| OpenVMS Batch and Print Queue System Concepts .....                   | 3  |
| Overview .....                                                        | 3  |
| OpenVMS Batch and Print Queuing System .....                          | 4  |
| Print Queuing Process.....                                            | 4  |
| Software Components .....                                             | 5  |
| Figure: OpenVMS Batch and Print Queuing System.....                   | 7  |
| Queue Manager Process .....                                           | 8  |
| Queue Manager Process in an OpenVMS Cluster System .....              | 8  |
| Starting the Queue Manager and Creating the Queue Database.....       | 9  |
| Queue Manager Failover.....                                           | 9  |
| Job Controller Process.....                                           | 10 |
| Job Controller Queue Management Functions.....                        | 10 |
| Events That Cause the Job Controller to Perform Queue Functions ..... | 11 |

|                                                       |    |
|-------------------------------------------------------|----|
| Queue Database .....                                  | 12 |
| Purpose.....                                          | 12 |
| Files.....                                            | 12 |
| Queue Master File.....                                | 12 |
| Queue File .....                                      | 13 |
| Queue Journal File .....                              | 13 |
| In-Memory Queue Manager Database .....                | 14 |
| Default Queue Database Location .....                 | 14 |
| Alternate Location for the Queue Database .....       | 14 |
| Performance Considerations .....                      | 15 |
| Moving Queue and Journal Files.....                   | 16 |
| Moving the Master File .....                          | 17 |
| Backup and Restore of the Queue Database.....         | 17 |
| Obtaining Information on Queue System Processes ..... | 19 |
| Displaying Queuing System Processes .....             | 19 |
| Using the Operator Log File .....                     | 19 |
| Print Symbiont.....                                   | 20 |
| Function of the Print Symbiont.....                   | 20 |
| Standard OpenVMS Print Symbionts .....                | 20 |
| System Resources for the Print Symbiont.....          | 20 |
| Displaying the Symbiont Process .....                 | 21 |
| Customizing Print Queues.....                         | 23 |
| Queue Options.....                                    | 23 |
| How to Set Options on a Print Queue .....             | 24 |
| Displaying Print Queue Options.....                   | 25 |
| Autostart Queues .....                                | 27 |
| Initializing Autostart Execution Queues .....         | 27 |
| Making an Autostart Queue Inactive .....              | 28 |
| Disabling All Autostart Execution Queues.....         | 28 |
| Queue Access .....                                    | 29 |
| Restricting Access to a Queue .....                   | 29 |
| Types of Queue Access.....                            | 29 |
| Default Queue Protection.....                         | 29 |
| Displaying Queue Protection .....                     | 29 |
| Remote Print Queue Concepts.....                      | 30 |
| Overview .....                                        | 30 |
| LAT Network Concepts .....                            | 31 |
| Overview.....                                         | 31 |
| Service Nodes .....                                   | 31 |
| Terminal Server Node.....                             | 32 |
| Sample LAT Network.....                               | 32 |
| Service Advertisements.....                           | 33 |
| Types of Service Requests .....                       | 33 |
| Terminal Server Concepts .....                        | 34 |
| Terminal Server Operating Modes .....                 | 35 |
| Terminal Server Differences .....                     | 35 |
| Terminal Server Databases.....                        | 36 |
| Troubleshooting Terminal Servers .....                | 37 |

|                                                          |    |
|----------------------------------------------------------|----|
| Displaying Server Characteristics .....                  | 37 |
| Testing a Terminal Server Port .....                     | 37 |
| Example: Terminal Server Port Test .....                 | 38 |
| LAT Management Tools .....                               | 39 |
| LAT Control Program .....                                | 39 |
| Uses for the LATCP .....                                 | 39 |
| Invoking the LATCP .....                                 | 39 |
| Sample LATCP Commands .....                              | 39 |
| LATCP DCL Command Line Interface .....                   | 39 |
| Troubleshooting Printer Queue Problems .....             | 40 |
| Overview .....                                           | 40 |
| Some Troubleshooting Tips .....                          | 41 |
| Questions to Ask .....                                   | 41 |
| Starting the Investigation .....                         | 41 |
| Isolating the Point of Failure .....                     | 42 |
| Using the COPY Command to Test a Printer .....           | 43 |
| Obtaining Information About the Queue .....              | 44 |
| Checking Mapping to the Server Port .....                | 45 |
| Getting a Test Sheet from the Printer .....              | 45 |
| Checking the Hardware .....                              | 46 |
| Print Job in Pending or Holding State .....              | 46 |
| Example: Print Job Submitted with /AFTER Qualifier ..... | 47 |
| How OpenVMS Formats a Print Job .....                    | 48 |
| Overview .....                                           | 48 |
| How a Print File Is Formatted .....                      | 49 |
| Print Symbiont .....                                     | 49 |
| Driver Reformats the File .....                          | 49 |
| Printer .....                                            | 51 |
| Managing Printer Settings .....                          | 51 |
| Troubleshooting File Output Problems .....               | 52 |
| Problem Definition .....                                 | 52 |
| Getting More Information .....                           | 52 |
| Summary .....                                            | 53 |
| OpenVMS Batch and Print Queue System Concepts .....      | 53 |
| Remote Print Queue Concepts .....                        | 53 |
| Troubleshooting Print Queue Problems .....               | 53 |
| How OpenVMS Formats a Print Job .....                    | 53 |
| Chapter Terms .....                                      | 54 |

## Module 4 — Troubleshooting OpenVMS Cluster Systems

|                                                  |   |
|--------------------------------------------------|---|
| Introduction .....                               | 2 |
| Objectives .....                                 | 2 |
| Resources .....                                  | 2 |
| OpenVMS Cluster System Concepts .....            | 3 |
| Overview .....                                   | 3 |
| OpenVMS Cluster System Software Components ..... | 4 |
| Types of OpenVMS Cluster System Nodes .....      | 6 |
| Satellite Node .....                             | 6 |

|                                                             |    |
|-------------------------------------------------------------|----|
| Boot Node .....                                             | 6  |
| Participating Node.....                                     | 6  |
| Cluster Communication.....                                  | 7  |
| Overview.....                                               | 7  |
| Cluster Group Number.....                                   | 7  |
| Cluster Password.....                                       | 7  |
| Cluster Formation.....                                      | 8  |
| Overview.....                                               | 8  |
| OpenVMS Cluster System Diagnostic Messages .....            | 10 |
| Software Sources of Diagnostic Messages .....               | 10 |
| Connection Manager Messages .....                           | 10 |
| Sample Connection Manager Messages.....                     | 12 |
| VAXport Error Messages.....                                 | 13 |
| Example: VAXport Error Messages .....                       | 14 |
| Error Log Entries for Cluster-Available Devices .....       | 14 |
| HSC Diagnostic Messages and Fault Codes.....                | 15 |
| Using NCP in an OpenVMS Cluster System .....                | 16 |
| NCP SHOW Commands.....                                      | 16 |
| Example: Displaying Node Characteristics .....              | 16 |
| Example Note: Displaying Node Characteristics.....          | 16 |
| Example: Displaying Circuit Characteristics.....            | 17 |
| Example Notes: Displaying Circuit Characteristics.....      | 17 |
| Example: Displaying Line Characteristics.....               | 17 |
| Example: Displaying Executor Characteristics.....           | 18 |
| OpenVMS Cluster System Bugchecks.....                       | 20 |
| CLUEXIT Bugchecks.....                                      | 20 |
| Common CLUEXIT Bugcheck Conditions .....                    | 20 |
| Failure of Nodes to Boot or Join the Cluster .....          | 22 |
| Overview .....                                              | 22 |
| OpenVMS Cluster System Initialization Concepts.....         | 23 |
| Normal Cluster Node Initialization.....                     | 23 |
| Node Boots.....                                             | 25 |
| Node Attempts to Form or Join Cluster .....                 | 25 |
| Node Joins Cluster.....                                     | 26 |
| CI Node Boot Problems .....                                 | 28 |
| Cluster-Related System Initialization Problems .....        | 28 |
| CI Node Fails to Join the Cluster .....                     | 28 |
| Satellite Node Initialization Problems.....                 | 29 |
| Satellite Initialization Landmarks .....                    | 29 |
| Message from Load Host Not Displayed.....                   | 29 |
| Handling OpenVMS Cluster System Hangs.....                  | 32 |
| Overview .....                                              | 32 |
| Recognizing and Handling OpenVMS Cluster System Hangs ..... | 33 |
| Be Sure the OpenVMS Cluster System Is Hung.....             | 33 |
| Cluster Quorum Is Lost.....                                 | 33 |
| Handling Loss of Quorum.....                                | 33 |
| Shared Resource Inaccessible .....                          | 34 |
| Conditions That Cause Shared Resource Problems .....        | 34 |

|                                                                      |    |
|----------------------------------------------------------------------|----|
| Case Studies .....                                                   | 35 |
| Overview .....                                                       | 35 |
| Case Study Problem: OpenVMS Cluster System Booting Problem .....     | 36 |
| Problem Statement .....                                              | 36 |
| Example: Console Errors .....                                        | 36 |
| Example: SHOW CLUSTER /CONTINUOUS Output .....                       | 37 |
| Example: SHOW SYSTEM Output .....                                    | 38 |
| Example: Satellite Boot Errors .....                                 | 39 |
| Example: SHOW CLUSTER /CONTINUOUS Output .....                       | 40 |
| Example: SHOW SYSTEM Output .....                                    | 41 |
| Example: Displaying VAXCLUSTER Parameter .....                       | 43 |
| Case Study Solution: OpenVMS Cluster System Booting Problem .....    | 44 |
| Solution .....                                                       | 44 |
| Configuring a Boot Server .....                                      | 44 |
| Restoring a Cluster Member .....                                     | 44 |
| Case Study Problem: OpenVMS Cluster System Communication Problem ..  | 46 |
| Problem Statement .....                                              | 46 |
| Example: Console Errors .....                                        | 46 |
| Example: SHOW CLUSTER /CONTINUOUS Output .....                       | 48 |
| Case Study Solution: OpenVMS Cluster System Communication Problem .. | 49 |
| Solution .....                                                       | 49 |
| Summary .....                                                        | 50 |
| Concepts .....                                                       | 50 |
| OpenVMS Cluster System Concepts .....                                | 50 |
| Failure of Nodes to Boot or Join the Cluster .....                   | 50 |
| Handling OpenVMS Cluster System Hangs .....                          | 50 |

## Learning Check

## Learning Check Answers

## Appendix A - Setting Up a Remote Print Queue on a Terminal Server





## Introduction

This section describes the contents of the *OpenVMS System Troubleshooting for System Managers* course and suggests ways in which you can most effectively use the materials. This section discusses the following topics:

- Course description
- Intended audience
- Prerequisites
- Course goals
- Course nongoes
- Course resources
- Course organization
- Module organization
- Course conventions

## Course Description

This course uses lectures, case studies, and laboratory exercises to provide experienced OpenVMS system and network managers with problem prevention and troubleshooting techniques designed to improve system availability.

This course presents troubleshooting methodology to diagnose and correct problems in any circumstance. The course also discusses tools that can be useful in OpenVMS system troubleshooting situations and how to ensure that the system is prepared for troubleshooting when problems occur.

Advanced concepts enhance your understanding of how OpenVMS works in several areas relevant to system troubleshooting. There is a heavy emphasis on practical laboratory exercises designed to allow you to apply the troubleshooting methodology to real problems.

The techniques taught in this course apply to all OpenVMS Alpha and OpenVMS VAX systems running the OpenVMS operating system at Version 7.2, including workstations and OpenVMS Cluster systems.

## Intended Audience

This course is intended for experienced OpenVMS system and network managers.

## Prerequisites

You should be able to perform the following tasks before enrolling in this course:

- Perform system startup for a single node or an OpenVMS Cluster node.
- Use OpenVMS utilities such as: BACKUP, INSTALL, SYSGEN, and MONITOR.
- Write and understand DCL command procedures.
- Create and modify print queues.

## Course Goals

To diagnose and correct OpenVMS system problems, you should be able to:

- Apply a troubleshooting methodology to isolate and define system problems in a clustered, networked environment.
- Use appropriate OpenVMS tools and utilities to troubleshoot the system environment.
- Use controls to prevent problems and prepare the system for troubleshooting unexpected problems.

## Course Nongoals

This course does not address all possible system problems, but rather uses a select group of problems to demonstrate troubleshooting techniques.

This course does not address the following topics:

- General OpenVMS or OpenVMS Cluster system management
- OpenVMS operating system internals
- Interpreting system dump files

## Course Resources

You must have access to an OpenVMS Cluster system, including a terminal server and printer, while you do the laboratory exercises. You also need the following additional resources to complete this course:

- *OpenVMS System Manager's Manual*
- *OpenVMS System Management Utilities Reference Manual*
- *OpenVMS VAX System Dump Analyzer Utility Manual*
- *Guidelines for OpenVMS Cluster Configurations*
- *OpenVMS Cluster Systems*
- *DIGITAL OpenVMS System Messages: Companion Guide for Help Message Users*
- *VAX/VMS Internals and Data Structures*
- *OpenVMS AXP Internals and Data Structures*
- *OpenVMS Alpha Upgrade and Installation Manual*
- *OpenVMS VAX Upgrade and Installation Manual*
- *DECams User's Guide*
- *DECnet-Plus for OpenVMS Introduction and User's Guide*
- *DECnet-Plus for OpenVMS Network Management*

Some of these documents are supplied with the student package; the rest should be available as reference in the classroom.

## Course Organization

In this lecture/lab course, you attend both lecture sessions and laboratory sessions. Lecture sessions consist of instructor presentations and class discussions. Laboratory sessions consist of instructor demonstrations or directed individual and/or team study. You should try to complete the written and laboratory exercises during the laboratory sessions.

Your instructor may provide you with a prerequisite test. Take the prerequisite test before beginning the course to find out whether you meet the prerequisites.

## Module Organization

Each module in the *Student Guide* consists of:

- An **introduction** that describes the purpose of the module, provides motivation for mastering its objectives, and outlines its contents.
- One or more **objectives** that identify the skills taught in the module. Objectives are designed to focus your study efforts on a selected number of skills.
- A list of **resources** you may need to complete the unit. Some of these resources are distributed with this course; your instructor should give you access to the others.
- The module **text** that consists of:
  - Descriptive text organized in a list format
  - Illustrations that clarify the relationships among various elements of a OpenVMS Cluster system or summarize steps of a particular process
  - Tables that present information in a summary form
  - Examples containing sample listings from actual interactive sessions
  - A module **summary** that reviews important concepts and skills taught in the module.

## Course Conventions

The following table describes the conventions used in the listings and command tables of the *Student Guide*.

| Convention                                 | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ctrl/X                                     | Press and hold the key labeled Ctrl while you press another key (X). Many control keys have special meanings.                                                                                                                                                                                                                                                                                                                                                                         |
| UPPERCASE                                  | <p>In commands, uppercase characters indicate words you type exactly as they appear. For example, you would type the following commands as they appear:</p> <pre>\$ DIRECTORY \$ TYPE LOGIN.COM</pre>                                                                                                                                                                                                                                                                                 |
| lowercase                                  | <p>Lowercase characters represent elements that you must replace according to the description in the text. For example, you must follow certain rules when you replace file-spec in the following example:</p> <pre>\$ TYPE file-spec</pre>                                                                                                                                                                                                                                           |
| Ellipsis<br>(...)                          | <p>Horizontal ellipses indicate that you can enter additional parameters, values, or information. For example, you can enter any number of file specifications in the following example:</p> <pre>\$ TYPE file-spec, ...</pre> <p>Vertical series of periods or ellipses mean that not all of the data that the system would display in response to the particular command is shown, or that not all the data a user would enter is shown.</p> <pre>\$ TYPE MYFILE.DAT . . . \$</pre> |
| Square brackets<br>([]) and braces<br>({}) | <p>Square brackets indicate that the enclosed item is optional. (Square brackets are not optional, however, in the syntax of some file specifications.) For example, the logical name is optional in the following command:</p> <pre>\$ MOUNT/FOREIGN \$TAPE1 [logical-name]</pre> <p>Braces indicate that you must select from the included items.</p>                                                                                                                               |
| DEFINE SET                                 | <p>The use of the   means to use either the first command or the second command. This may be device specific. On some devices the command is DEFINE and on others SET.</p>                                                                                                                                                                                                                                                                                                            |

## Instructor Notes

Compaq systems enjoy an enviable reputation in the market for dependability and reliability. All computer systems experience problems from time to time and often, if not in the majority of cases, **the problems are caused by lack of training and experience on the part of the system and network manager**. Although this course addresses system problems, you should emphasize the reliability of Compaq systems and not allow the discussion to imply otherwise.

*OpenVMS System Troubleshooting for System Managers* is a course for experienced system and network managers. It is not a substitute for the internal course, *OpenVMS Troubleshooting* which discusses interpreting crash dumps.

**You can think of it as a “graduate course” in system management**, one of several courses offered for those who have completed *OpenVMS System and Network Node Management I*, and *II* in the OpenVMS curriculum map.

Students may come to this course expecting to solve particular problems. This course avoids problems that are:

- Hardware dependent or related.
- OpenVMS version bugs that will be fixed before this course is available.

Therefore, it is most important to set proper expectations at the beginning of the course. **This course uses a set of problems to illustrate the troubleshooting methodology. It does not cover all possible problems.** Problems were chosen for inclusion in this course based on the following criteria:

- *The problem is caused by human error.*
- The problem is realistic.
- The problem is easy to reproduce in the laboratory situation.

To research this course, some of the developers spent two weeks at the Colorado Springs Customer Service Center. Three areas were identified as the source of large numbers of problem calls to the center:

- System initialization (boot) problems leave the system manager feeling helpless. This course shows that even when the system will not boot, the system manager has tools and options to restore the system.
- Because printing involves so many components and there is so much to remember, it is a major source of problem calls. This course addresses the printing process from beginning to end in order to increase the system manager's confidence in his or her ability to solve printing problems.
- OpenVMS Cluster system problems are frequently associated with adding or moving a satellite node.

**Troubleshooting is heroic.** A system that needs troubleshooting is not an available system. The best way to maintain dependable systems is to **avoid the problems in the first place**. The course discusses methods of avoiding problems and puts special emphasis on avoiding the repetition of a problem.

The format of this lecture/laboratory course is the same as that used in the most recent update of the *OpenVMS System Performance Management* course.

- Each section of the module is introduced with an overview page that explains what will be learned in the module and why a system manager needs to learn the materials. It also indicates any topic that is not covered in the module when there is some reason to assume that it might be covered.

Instructors may want to use this page to focus the students' attention on the topics.

- At the end of each *module* is a section that contains new terms learned in the module. Each new term is italicized in the text when it is first mentioned.

## OpenVMS Alpha Coverage

This course has been updated to Version 7.2 of both OpenVMS VAX and OpenVMS Alpha operating systems. The code examples generally are taken from OpenVMS Alpha systems, and when discussing commands or concepts that differ between the two operating systems, the OpenVMS Alpha version is given first.

In general, the two operating systems are functionally equivalent. You should be able to teach this course to students who have all OpenVMS VAX systems, all Alpha systems, or a combination of the two. The base system for the revision of the course was an Alpha 4000 in an NI cluster with a VAX 4000-300 system.

## Course Schedule

See the Table of Contents. Each module contains a suggested time for materials or lecture and a suggested time for practice or laboratory exercises. The modules are found in the Student Guide in the order in which they are typically taught, but you have some flexibility.

Module 1 should be taught first. Module 2 must be taught before Module 4.

Students attending the pilot course suggested that Module 4 be taught right after Module 2.

## Student Package

Each student in the class receives a student package consisting of a Course Guide.

## Instructor Package

Each instructor must have the Instructor Package to teach the course. All directions for creating the laboratory exercises are included in the guide. The instructor package contains an Instructor Guide

## Student References

You should provide a full OpenVMS documentation set in the lab or classroom for reference.

## Instructor Preparation

Instructors qualified to teach *OpenVMS System and Network Node Management II* should be able to prepare for teaching this course. Experience with teaching *OpenVMS System Performance Management* will be helpful. An understanding of VMS internals and data structures will also be helpful.

The instructor needs to be skilled in leading case study discussions.

To prepare for teaching this course, you need:

- *Instructor Guide*
- A full OpenVMS document set

You also need access to systems on which to perform the laboratory exercises to prepare for this course.

## Course Notes File

There will soon be an instructor notes file type place to be used for comments, updates, and corrections regarding this course.

## Preparing the Systems

There is no media with this course. You must create two files. The instructions for creating these files are given in the Instructor's Guide on the appropriate pages.

## Required Equipment

Before offering this course, prepare the systems to support the laboratory exercises that accompany the student materials.

The optimal and minimum configurations to support the laboratory exercises for a class of 16 students are:

|                                                | Optimal       | Minimum          |
|------------------------------------------------|---------------|------------------|
| OpenVMS Alpha or OpenVMS VAX systems           | 1 per student | 1 per 2 students |
| LA50 printer (or equivalent)                   | 4             | 1                |
| Terminal Server                                | 8             | 4                |
| Terminals connected to terminal servers        | 16            | 7                |
| OpenVMS Cluster system with one satellite node | 6             | 4                |

The OpenVMS Alpha and OpenVMS VAX systems are used in Modules 1 and 2. The exercises require that the system be booted and crashed. These systems should not be used by any other class during the time this class is performing the laboratory exercises.

The terminal servers are used in Module 3 and must be on a OpenVMS Cluster system with at least two nodes. All terminal servers can be on the same cluster. One node of this cluster is shut down during the exercises. Students need the terminal server password.

Each group of students needs one terminal on the terminal server to execute terminal server commands and a second terminal to emulate the printer. One exercise requires that a real printer be used.

## Creating Student Accounts

Most of the laboratory exercises require the students to use the system account on the dedicated laboratory systems with full privileges. However, you should create a separate account for each student in the course on the classroom system. Each student should also be allocated a disk quota of at least 1000 blocks if quotas are enabled. None of the operations performed by students require enhanced quotas or special parameter values. Allow these values to default to the values specified for your training center.

The student accounts should have the following privileges, at minimum:

- TMPMBX — Allows use of queues
- NETMBX — Allows use of DECnet



---

# ***Dependable Systems Overview***

## ***Module 1***

## Introduction

The impact of a computing system component failure can range from merely inconveniencing certain users, to the temporary loss of productivity and revenue, to harming the organization's long-term financial health or endangering public safety.

As organizations become more dependent on computing systems, the role of the system manager in providing dependable systems increases. The system manager's responsibilities may include:

- Designing a cost-effective, dependable system
- Establishing procedures and controls to prevent problems
- Troubleshooting and resolving system problems

## Objectives

To provide dependable systems, you should be able to:

- Describe dependable system concepts.
- Describe a method of troubleshooting system problems.
- Describe OpenVMS system messages.
- Use OpenVMS troubleshooting tools.

## Resources

- *OpenVMS VAX System Dump Analyzer Utility Manual*
- *OpenVMS System Messages: Companion Guide for Help Message Users*



---

**Note**

Study Time

Materials: 4 hours

Practice: 2 hours

---

# Dependable Systems

## Overview

A dependable computing system is one that can be counted on to provide services to users when those services are needed.

To participate in decisions regarding system configuration, you should be able to describe the levels of system dependability requirements.

This section discusses:

- System dependability categories
- System problems



### Note

Each section of this course starts with an overview page. The overview includes

- A brief introduction to the topic
  - The reason the target audience needs to know the material
  - A statement regarding any obvious material that is not covered in this section
  - A list of the subtopics in the section
-

## System Dependability Categories

### Categories of Dependability

Different categories of system dependability provide for different business needs. These include:

- Conventional computing systems
- Highly available computing systems
- Mission-critical computing systems
- Disaster-tolerant computing systems



---

#### Note

For further reference, see *Building Dependable Systems: The OpenVMS Approach*. Use this section to build awareness of dependable systems.

Of course, every business wants its systems to be available all the time. Providing the maximum protection from system failure is a costly investment. The most expensive and most dependable systems are not cost-justified in every case. Users always want systems to be available whenever they need to work. It is important to set user expectations regarding the level of system dependability and to keep records showing that expectations have or have not been met.

---

### Conventional Computing Systems

Conventional computing systems provide reliable service during specified hours. Infrequent service interruptions and downtime during other hours for backup and maintenance are expected.

**Business example:** A computer system that provides office automation for a department of a large corporation. The system consists of a two-CPU OpenVMS Cluster. Daily and weekly backups of files to tape are performed and **stored off site**. If service is interrupted by power failure or component failure, the department is prepared to function without the service for a limited period. In the event of a disaster, the department will use the limited facilities of a backup hot site and resort to manual operations.



---

#### Note

There is no excuse for not storing backup tapes off site! A fire or earthquake would make onsite storage inaccessible even if they are stored in a fireproof vault.

---

## Highly Available Computing Systems

Highly available computing systems provide uninterrupted computing services during essential time periods. This requires redundant hardware and a backup power supply. Disk backup and routine maintenance are performed during non-business hours.

**Business example:** Business functions that require uninterrupted computing services, either during essential time periods, or during most hours of the day and most days of the week throughout the year. An automobile registration system that must always be available during business hours could be built using an OpenVMS Cluster with shadowed disks and an uninterruptible power system. During non-business hours, the system can be shut down for maintenance and disk backup. In the event of a disaster, the system can be established at a hot backup site within 16 hours.

## Mission-Critical Computing Systems

Mission-critical computing systems provide for functions where the business's financial goals depend directly on how well its computing systems support the firm's applications. This may require fault-tolerant computing systems.

**Business examples:**

- A telephone catalog sales company requires that customers can call in orders 7 days a week, 24 hours a day.
- The part of the automobile registration system that allows police officers to radio in license numbers for registration checking.

## Disaster-Tolerant Computing Systems

Disaster-tolerant computing systems provide for functions that must be immune from disasters such as earthquakes, fires, floods, hurricanes, power failures, and acts of terrorism. This requires geographically separated, redundant hardware configurations.

**Business example:**

The control system for a nuclear power plant.

## System Problems

### Classifying System Problems

System problems can be classified according to their cause:

- Environmental
- Hardware failure
- Software failure
- Human errors

### Environmental Problems

Environmental problems include:

- Power fluctuations and loss
- Temperature
- Air quality
- Physical security

### Hardware Problems

All hardware eventually fails due to friction, metal stress, and other factors.

*Mean-time-between-failures* (MTBF) is the average time that passes before a computing component fails such that remedial action is required.

To reduce the impact of hardware failure:

- Buy reliable hardware.
- Provide correct maintenance and environmental conditions.
- Provide redundancy. Avoid a single point of failure.
- Anticipate failures and replace or service equipment before it fails.



---

#### Note

Hardware (not just computer hardware) always fails eventually due to friction, metal stress, and so on. **Compaq has a reputation for making very reliable hardware.** Do not allow this category of the discussion to reflect negatively on the company. Most computer hardware becomes obsolete long before it wears out.

Discuss shadowing the system disk as an example of preventing a single point of failure.

---

## Software Problems

Theoretically, software could be built that would not fail.



---

### Note

Theoretically, software could be built that would never fail. In the real world this seldom happens.

System managers have little control over software development. They do, however, sometimes have some control over what software is installed on the system.

There should be control procedures to follow before software is installed that provide for:

- Testing
  - Proper backup requirements
  - Problem documentation and reporting
  - System performance impact
- 

## Human Errors

The keys to minimizing human errors are:

- Training (including documentation and practice)
- Making the system easy to use



---

### Note

Human errors can be prevented or significantly reduced by training and/or controls.

To avoid operator errors, instructions for handling tasks that occur infrequently should be posted near the terminal where they are performed. For example, something that happens about once a month should be fully documented. Once a month is not often enough for the operator to retain complex instructions.

**BACKUPS, even daily backups should be documented and initialed in a log book. Look at this log book every week.**

Equipment in a data center should be clearly labeled so the operator knows immediately what system it is.

---

## Handling Problems

Even the best managed systems cannot avoid all problems. Good problem management can minimize the impact of the problem on the business.

- Detect problems as soon as possible. Have an established procedure for problem reporting.
- Prepare for troubleshooting by installing the appropriate tools.
- Follow good troubleshooting techniques.
- Prevent problems from recurring.



---

### Note

Many tools are available to detect and prevent problems, and more can be written using DCL. For example, you can easily write a command procedure to look for disks that are overcrowded or to check that all files have been backed up each morning.

The best way to handle a problem is to avoid having one in the first place. Some problems cannot be avoided, but the impact of the problem can be minimized.

Gradually introduce the idea that while firefighting is heroic, it is not the best way to provide dependable systems.

Dependable systems are systems where problems are avoided.

After the immediate problem is fixed, hold a post-problem review session with personnel involved. Determine what caused the problem in the first place. Can anything be done to prevent the problem from occurring again? If you cannot prevent the problem, then at least prepare to correct the problem quickly.

- Make sure the problem solution is documented so that it can be quickly reviewed.
  - Have the proper tools handy: hardware tools such as screw drivers and software tools such as a system dump file.
  - Train personnel to recognize the problem and implement the solution.
-

# Troubleshooting Methodology

## Overview

When a system problem occurs, the system manager must apply a systematic method to solving the problem.

A suggested troubleshooting method consists of nine steps:

1. Define the problem.
2. Gather information about the problem.
3. Analyze and interpret the information.
4. Propose a solution, or refer the problem to the responsible person.
5. Verify the solution.
6. Clean up any parameters or files created for testing.
7. Reverify the solution.
8. Document the problem and archive the solution.
9. Investigate methods of preventing the problem from occurring again.

These nine steps are discussed in this section.

## Step 1: Define the Problem

### Purpose

The problem must be defined exactly before it can be analyzed and explained.

- The following is not a helpful problem definition:  
“My files will not print.”
- A clearer problem definition would be:  
“The printer in my office jams every time I try to print a file.”

It is important to **define the problem** without assuming what caused the problem.

For example, defining the problem as “the printer is broken” when files are not printing directs attention to the printer as the source of the problem when, in fact, it may be that the queue is the problem.

### What to Do

When you encounter a system problem, ask the following preliminary questions of yourself or the user who reported the problem:

- What was the user trying to do when the problem occurred?
- What command was entered?
- What result was obtained?
- Is the problem accompanied by one or more error messages? What are the exact messages? In what order are they displayed?
- In the event of a hardware error (such as a printer problem), are any warning lights on?



---

#### Note

It is a good idea to insist that users write the error messages down and turn them in with the problem report. Often, when people are forced to read an error message carefully, they find that they can solve the problem themselves.

---

## Document the Process

As you ask questions and gather information about the problem, write the answers down.

This information is helpful to you when you analyze the information, and it saves time for other people to whom you may later escalate the problem.



---

### Note

The correct order of the questions will come with time and experience. The system manager will learn to order the questions based on:

- The skill of the users
  - The activity of the shop
-

## Step 2: Gather Information

### Purpose

After you have a clear problem statement, gathering additional information helps you narrow both the scope of the problem and the possible causes.

You may need to gather information several times throughout the process to isolate the source of the problem.

### Relate Problem to System Changes

It is most important to know what changes have been made to the system and when they were made.

- Has a new layered product been installed?
- Have system parameters been changed?
- Was the system just reinitialized?
- Has new hardware been installed?



#### Note

Focus this discussion on what has changed.

Providing dependable systems requires a change management strategy that records and documents all changes to the system no matter how trivial. One of the problems we will solve in the lab exercises deals with a change made to the SYSTARTUP\_VMS.COM file that causes the procedure to loop and prevents the system from booting. It is not difficult to visualize a situation in which the file is changed by a system manager who then goes on vacation or leaves the company. A month later the system is rebooted. Problems! Who remembers that the file was updated?

Operator shift reports should include problem descriptions, changes, and so on. These reports should be reviewed each day by the system manager and others. These reports should be saved.

System managers should document all changes they make to the system.

## Questions to Ask

The following questions can guide you in gathering information about the problem:

- What system or systems are involved?
- What preceded the problem? What were the previous operations you performed before you detected the problem?
- What was happening on the system at the time the error occurred?
- What operation were you performing when the error occurred?
  - Can you relate the error to a system application, utility, or command?
  - Does the application utility or command have limitations or restrictions?
- Has this happened before, or is this the first time you noticed the problem?
- When did the error occur? Watch for cyclic peak periods such as start of day, noon, end of day, and overnight batch runs.
- What is the scope of the problem: system, cluster, LAN, WAN?
- What applications are affected?

## Step 3: Analyze and Interpret Information

### Purpose

You may need to analyze and interpret information repeatedly as you gather more specific information about the problem you are solving. Continual information gathering throughout the troubleshooting process, followed by gathering, analyzing and interpreting the information helps you to isolate the source of the problem and eventually solve the problem.

### Analyze the Information

Group the information into the following four categories to get a clear picture of the problem. You may need to reword the sample questions to fit your situation, but you should attempt to find information for each category.

| Category | Sample Question to Ask                   |
|----------|------------------------------------------|
| What     | What do we observe?                      |
| When     | When is the problem observed?            |
| Where    | Where is the problem observed?           |
| Extent   | How serious or extensive is the problem? |



---

#### Note

Group the information so you can rule out certain problems and begin to isolate the source of the problem. Isolating the source of the problem may be all that you can do to resolve a fault.

*Extent* refers to the bounds of a problem's impact on a system or group of systems.

- Are all users affected or only one user? Only privileged users? One UIC group?
  - Are all machines affected? Is the problem on one node, an OpenVMS Cluster, or is it network-wide?
-

## Interpret the Information

To interpret your data, establish a basis for comparison. Evaluate the elements of information and the relationships among the elements to get an understanding of the problem.

- Determine what conditions are required to cause the problem to happen again.
- Try to identify a similar situation that could experience the same problem, but does not experience it. Then ask yourself what is different about that other situation? Asking this question makes it possible to identify the factors that are particular to the problem.
- Another technique is to ask, "What has changed?" Look at the differences between the problem situation and the situation that does not have the problem, and determine whether any of these differences represents something that has changed.
- Try to make the problem move. Have the user log in to another node, and see if the same thing occurs. What is different on that node, and what is the same?
- If you are unable to replicate a problem, you will probably not be able to find a solution.
  - Write a problem report that includes all relevant information.
  - Save the report to use if the problem happens again.
- Try to reduce the situation to the simplest set of circumstances that will still demonstrate the problem.
- You may discover that there is more than one problem, requiring more than one solution. Stay focused! Solve one problem at a time.



### Note

If a problem cannot be replicated, it is not that the problem did not happen, it is just that you do not know the set of conditions required to cause the problem. The problem may be caused by a set of conditions that do not occur very frequently.

In general, computer problems or incidents do not involve just one transaction. They usually apply to a group of transactions, users, devices, and so on. However it may be a very small group.

A problem that happens once and never happens again is not really a problem. Fix the consequences and go on. However, always document the problem in case it happens again. If it happens again, the first incident combined with the second incident may identify the conditions.

---

### Example: Interpreting Information

Consider two identical printers connected to the same computer system. The printers are set up the same way, and they are located side by side. The paper in printer A jams frequently, but the paper in printer B rarely jams. What is different about the two printers?

Investigation might reveal that printer B always prints on plain paper, but printer A is often loaded with hole-punched paper. Most of the jams occur when hole-punched paper is used. It is more important to fan hole-punched paper before loading. Because various people load the printer, the paper is not always fanned correctly.



#### Note

This “hardware” problem is actually a result of having users who are inadequately trained.

Placing a picture above the printer that demonstrates the proper method of fanning the paper before loading might remedy this situation.

You might want to mention another option, that of forbidding users to load the printer. Put up a sign telling users to call the system manager to load the printer and then have the printer sit there empty until the system manager has time to load the printer.

---

## Step 4: Formulate and Apply a Proposed Solution

### Purpose

Use the information you have gathered to formulate a proposed solution to the problem.

Implement the solution, if appropriate.

### What to Do

At this step, you should do one or more of the following:

- Propose a solution to the problem.
- Investigate the solution further by testing other relationships.
- Apply the proposed solution or escalate the problem for resolution.
- Consider installing a temporary solution or workaround.



---

**Note**

Computer systems are easy to troubleshoot compared with a situation where you must be very careful before trying a proposed solution.

For example, suppose you were trying to figure out why a plane crashed or why a large gun exploded.

After you investigate the problem, you may have a good idea of what caused the problem. For example, a user cannot print to a printer. You investigate and find that the protection on the queue does not allow access. Your proposed solution is to change the protection on the queue. You change the protection on the queue and the user still cannot print the file. Your proposed solution did not work, but at least a plane didn't crash. However you may want to back out your change or at least find out why the queue protection was set that way in the first place.

---

### Propose a Solution

After you have analyzed and interpreted the information, you may be able to propose a specific solution:

- New solution
- Known solution



---

**Note**

It is very important to change only one thing at a time. If you decide to modify three system parameters, change the protection on two queues, and rebuild a disk, how can you know what the real solution was? Unless it is absolutely necessary, change one thing, test the change, and repeat the process.

---

## **Investigate the Proposed Solution**

After you formulate a proposed solution, investigate the solution by looking for other relationships.

If *X* is the problem, then *Y* should be true/false.

For example, if you think a user in group A cannot print a file on a print queue because the queue protection only allows a user in group C to print the file, then:

- Test to see if another user in group A can print to the queue. If the result of this test is negative, your proposed solution is supported.
- Test to see if a user in group C can print to the queue. A positive result supports the proposed solution.

## **Apply the Proposed Solution**

Implement the solution.

For example change the protection on the queue to allow world access.

## **Escalate the Problem**

If the required solution is beyond the bounds of your authority or expertise, you may need to seek help from another person.

An escalation process should be in place and available to all personnel who are responsible for system operations. A listing of who to contact should be part of the standard operating procedure for any operations center.

## **Temporary Solutions**

If you cannot solve a problem quickly and system availability is critical, you may have to implement a temporary solution that enables the system to remain running while allowing you to work on the problem.

These temporary measures are not ideal solutions and may cause a decrease in system performance. However, for most users, decreased service during the repair phase is preferable to no service at all.



---

**Note**

Even a temporary solution should be fully documented. Too often, the system manager will apply a quick fix to a situation, then forget about the original problem. Months later, the “temporary” fix may cause additional problems, because these quick solutions are often not well planned.

**An example:** A user is having trouble accessing a network node. You find that the address has changed, and do an NCP SET ADDRESS 1.234. Later, when the system is rebooted and the permanent (and unchanged) database is read, the problem returns.

**Another example:** A user is having trouble with the system parameters on a workstation. You modify the parameters in SYSGEN (they happen to be dynamic parameters), but forget to make the change in MODPARAMS.DAT.

---

## Step 5: Verify the Solution

### Purpose

Verify your solution to make sure that it explains every aspect of the observed problem. Verification is possible in most problem situations.

### What to Do

Attempt to reproduce the problem under the same circumstances, and make sure the problem does not occur.

Attempt to reproduce the problem under slightly different circumstances. For example, test the solution from both a privileged and non-privileged account.

Does the solution survive system initialization?

- The best way to determine this is to initialize the system.
- In many cases, you may not be able to initialize the system to test your solution.
- Try to ascertain that the change is permanent.
- Make a note to test the solution after the next system initialization occurs.

If the problem occurs after the proposed solution has been implemented, go back to investigating the problem.

### Example: Verifying a Problem Solution

After changing the protection on the queue to allow all users access to the queue:

- Check to make sure that non-privileged users from groups A, B, and C can print to the queue.
- Check to make sure the change will survive system initialization.

## Step 6: Clean Up

### Purpose

After you solve a system problem and verify the solution, clean up the system.

This step ensures that any parameters or tools you used in solving the problem do not interfere with the operation of the system.

### What to Do

To clean up the system:

- Remove any solutions that were temporarily installed or enabled for troubleshooting.
- Delete test files.



---

#### Note

Do your cleanup one step at a time, verifying the solution each time you remove a piece of your troubleshooting “apparatus.” If the problem reappears, you will know what change caused it to occur.

You may have installed a terminal at the end of the printer cable to test the terminal server.

You may need to run a program to test or look at the setting on something.

---

## Step 7: Reverify the Solution

### Purpose

After you clean up temporary test facilities and solutions you used to solve the problem, confirm that your solution is still correct.

It is possible that some of the tools you used for testing masked the problem.

### What to Do

Attempt to create the problem again under the same and slightly different circumstances.

Step 6 and step 7 should be done together.

- Remove one test file or temporary solution
- Test

### Further Actions

If the proposed solution has solved the problem, you should:

- Cancel any requests you may have made for service from outside vendors.
- Notify users that the problem has been corrected.

If the solution does not solve the problem, you should:

- Gather more information.
- Analyze and interpret the data again.

## Step 8: Document the Problem

### Purpose

Keeping records of problem solutions and the processes that led to those solutions can provide insight to future system problems.

If you keep a log of the troubleshooting operation, you will find it easy to document the solution.

### What to Do

Include the following information in your documentation:

- When the problem occurred
- Symptoms of the problem
- Explanation for its occurrence
- How you solved the problem
- Tools used to solve the problem
- Temporary means put in place to enable users to work while the problem was being solved



---

**Note**

This is possibly the most important step, and is a central truth that students should take away from the course. A notebook with system problems and their solutions is a goldmine of information and can save the system manager time and frustration.

---

## Step 9: Plan to Avoid or Handle Future Occurrences

### Purpose

After the problem is solved and the systems are fully available, take steps to prevent the same problem from interrupting service again. If this is not feasible, prepare to recognize and solve the problem quickly.

### What to Do

Study the problem and its cause to determine if you can do something to prevent the problem from occurring again.

- Software solution
- Hardware solution
- Manual control or training

If prevention is not possible or too expensive, prepare yourself and at least one other person to recognize and handle the problem.



---

#### Note

Troubleshooting is costly in work time lost because the system or part of the system is unavailable. Often something can be put into place to prevent the problem from happening. To determine if the preventive measure is cost justified, it is necessary to have an estimate of the cost of system disruption.

The order of the preventive measure represents, in general, the cost of the solution. In general, and depending on the problem, a software solution is cheaper than a hardware solution and both are cheaper than a manual procedure. However, this may not always be true.

If the prevention is too expensive, the next best thing is to be prepared to handle the problem if it occurs again. If one person knows how to handle a problem, then what happens when the problem occurs when that person is on vacation?

If a problem is expected to occur frequently (daily), then training people to handle the problem is justified. A problem that is expected to occur infrequently should be documented for easy reference because people cannot be expected to retain unused information for long periods of time. Problem documentation is also used to train new workers.

For example, about once a month a line printer jams while printing a long report and the operator needs to stop the printer, fix the paper, back the print job up for the number of damaged pages, and restart the job. Because this operation takes several steps, post step-by-step instructions for handling the situation right over the terminal where the operator works.

---

# OpenVMS System Messages

## Overview

The OpenVMS operating system provides a centralized error message facility. When you enter a command at the terminal or execute an image that results in an error, the system locates the message associated with the error and displays it. The system also sometimes generates a message when a command has completed successfully.

You can get more information on system messages through online help.

This section discusses OpenVMS system messages.

## Format of OpenVMS System Messages

Most messages displayed by the OpenVMS operating system have the following format:

- Main message  
%FACILITY-L-IDENT, message-text
- Secondary message  
-FACILITY-L-IDENT, message-text

Where:

- FACILITY specifies the abbreviated name of the software component that issued the message.
- L shows the severity level of the condition that caused the message.
- IDENT is a symbol of up to 15 characters that represents the message.
- Message-text explains the cause of the message.

Some OpenVMS facilities produce messages with a slightly different format.

### Example: Error Message

In the following error message, the main message is a warning (W) and the secondary message is an error (E).

```
$ TYPE MYFILE.DAT
```

```
%TYPE-W-SEARCHFAIL, error searching for  
DISK01:[SMITH]MYFILE.DAT; -RMS-E-FNF, file not found
```

## Severity Levels

The following table summarizes the severity levels of OpenVMS system messages.

| This code...    | Means...                                                                                                                                      |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| S (Success)     | The system successfully performed your request. In some cases, the command processing continues after the message is issued.                  |
| I (Information) | The system performed your request. The message provides information about the process.                                                        |
| W (Warning)     | The command may have performed some, but not all, of your request. The message may suggest that you verify the command or the program output. |
| E (Error)       | The output or program result is incorrect, but the system may attempt to continue execution.                                                  |
| F (Fatal)       | The system cannot continue to execute the request.                                                                                            |

## Accessing Message Descriptions from Online Help

In OpenVMS Version 5.5 and higher, online message help is included as part of the main help library.

Use the following command to access a message description:

```
$ HELP/MESSAGE ident
```

The OpenVMS installation kit includes an optional online help version of the *DIGITAL OpenVMS System Messages: Companion Guide for Help Message Users*. Because the system message online help is optional this document provides information on how to add or delete it.

## Example: Message Description

The following example displays the help information for a message with the identification ACCVIO.

```
$ HELP/MESSAGE ACCVIO
```

```
ACCVIO, access violation, reason mask='xx', virtual  
      address='xxxxxxxx', PC='xxxxxxxx', PSL='xxxxxxxx'
```

```
Facility:  TPU, DEC Text Processing Utility (DECTPU)
```

```
Explanation: An internal DECTPU failure occurred.
```

```
User Action: Contact a Digital support representative.
```

```
Press RETURN to continue ...
```



### Note

If there is more than one description associated with an error message, they are displayed in the order that they were entered into the help library. In the case of ACCVIO, for example, the second description is for the more familiar system (rather than TPU) error message.

Make a note that students should not give up if the first description does not appear to be related to what they were doing when the problem occurred.

---

# OpenVMS Troubleshooting Tools

## Overview

The Error Log utility is a system management tool that selectively reports the contents of one or more error log files.

The System Dump Analyzer is a system management tool that is used to:

- Interpret the contents of the system dump file
- Copy the contents of the dump file to another file
- Analyze the running system

The system and network manager is responsible for:

- Maintaining the error log file and using the Error Log utility to detect hardware problems
- Making sure that the system writes a dump file whenever the system fails, that the dump file is large enough to contain all the information to be saved, and that the dump file is saved for analysis

Reading system dump files is an advanced topic that is covered in a specialized course.

This section discusses:

- Error Logging facility
- Operator log files
- System bugchecks
- System dump files
- Capturing dump files on systems with limited disk space
- Using the System Dump Analyzer to analyze a crash
- Using the Crash Log Utility Extractor

## Error Logging Facility

### Components

The Error Logging facility includes:

- Executive routines that detect errors and events and write relevant information into error log buffers in memory
- A process called ERRFMT, which starts when the system is booted
- Periodic emptying the error log buffers
- Formatting of standard error messages
- Stores the formatted information in a file on the system disk
- The Error Log utility, which is used to selectively report the contents of the error log file

### Events Recorded

The OpenVMS operating system records several categories of events.

| <b>This category...</b> | <b>Includes these events...</b>                                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Errors                  | Device errors, device timeouts, machine checks, bus errors, memory errors, asynchronous write errors, undefined interrupts, and bugchecks         |
| Volume changes          | Volume mounts and dismounts                                                                                                                       |
| System events           | Cold startups, warm startups, system failure startups, messages from the Send Message to Error Logger (\$SENDERR) system service, and time stamps |

## Hardware Types

The Error Log utility functions with most OpenVMS supported hardware, such as:

- CPUs
- Disks
- Tape drives
- Some communications devices



### Note

Compaq hardware is very reliable. Be careful to emphasize reliability as a feature of Compaq hardware. However, Compaq hardware can be made more reliable by providing the proper environment for the hardware.

Monitoring hardware errors can predict that a component is about to fail. (The Error Log utility is used for this.)

Another note: Customer service keeps records regarding the mean time between failures (MTBF) for hardware components. For customers with a service contract, customer service may replace a part when the anticipated MTBF approaches.

---

## Error Log Buffers

The routines that detect errors and events write the information into error log buffers.

- Error log buffers are located in nonpaged pool in system space.
- The information in the error log buffers is written to the error log file by the ERRFMT process.
- Error log buffers are explicitly copied to the system dump file when the system crashes.
- The number of error log buffers and the size in pages are controlled by the following system parameters:
  - ERRORLOGBUFFERS
  - ERLBUFFERPAGES
- A burst of errors may fill up the buffers faster than the ERRFMT process can remove them. These error messages are lost.

**Note**

During system initialization (boot), a group of buffers is allocated in contiguous nonpageable system address space. The number of buffers allocated is specified by the system parameter `ERRORLOGBUFFERS`. Prior to VMS Version 5.0, the number was fixed at 2.

The number of pages in each error log allocation buffer is specified by the system parameter `ERLBUFFERPAGES`, default 2.

The group of buffers is treated as a ring. Initially, error message buffers are reserved in the first allocation buffer. When it fills, error message buffers are reserved in the second buffer. After a buffer fills, the `ERRFMT` process is awakened to copy the buffer's contents to the error log file so the buffer can be reused.

System parameters:

- **ERLBUFFERPAGES** specifies the number of pages of memory to allocate for each buffer requested by the `ERRORLOGBUFFERS` parameter. The default is 2 pages and a maximum value of 32 pages. (OpenVMS Alpha systems use pagelets for these values, which are 512 bytes, the same size as an OpenVMS VAX page.)
- **ERRORLOGBUFFERS** specifies the amount of physical memory (page) reserved for the system error log entries. If too low, messages might not be written to the error log file. If too high, too many physical pages can be consumed by the error log buffers. If you increase `ERRORLOGBUFFERS`, you also need to increase the size of the dump file by an equal number of blocks.

---

**Error Log File**

The OpenVMS operating system automatically records device and CPU error messages in the error log file `SYS$ERRORLOG:ERRLOG.SYS`.

The error log file increases in size and takes up space on the system disk until it is explicitly renamed or deleted.

Run a command procedure periodically to:

- Rename the current copy of the error log file to a file named, for example, `ERRLOG.17-DEC-95`
- Optionally back up the renamed error log file to off-line media
- Optionally delete the renamed error log file

**Note**

Not all hardware failures give warning error counts.

---

## Example: Displaying Device Errors

Use the SHOW ERROR command to display device errors.

```
$ SHOW ERROR
```

| Device                     | Error Count |
|----------------------------|-------------|
| PEA0:                      | 15          |
| \$1\$DUA1: (HSC01, HSC02)  | 1           |
| \$1\$DUA4: (HSC01, HSC02)  | 3           |
| \$1\$DUA10: (HSC01, HSC02) | 3           |
| \$1\$MUA0: (HSC01, HSC02)  | 2923        |
| \$1\$MUA3: (HSC01, HSC02)  | 3901        |
| \$1\$MUA1: (HSC02, HSC01)  | 3179        |
| \$1\$MUA2: (HSC02, HSC01)  | 3806        |
| \$1\$DUA8: (HSC02, HSC01)  | 17          |

The SHOW ERROR command only lists errors accumulated by one OpenVMS Cluster node and only the errors accumulated since the last time the system was booted.



### Note

It is most important to mention that the DCL SHOW ERROR command only displays the errors accumulated by one node and only the errors accumulated since the last system boot.

If, for example, you use this command to look at errors on a disk from a node that has not run a process that required I/O to that disk, you might miss an accumulation of errors that could be seen from another node in the cluster.

- DECEvent software displays errors accumulated by all nodes in the cluster and is not limited to errors accumulated since the last reboot. This software reads the error log file and alerts the customer to potential hardware problems. You should familiarize yourself with DECEvent by reading the *OpenVMS System Management Utilities Reference Manual*.
- DECEvent is provided free by Multivendor Customer Services to customers holding service contracts and/or equipment under warranty. It is not available as a separate product.

## Error Log Utility Reporting

To selectively report the contents of an error log file, use the DECEvent command:

```
$ DIAGNOSE [/qualifier(s)] file-specification
```



---

### Note

The detailed contents of the DIAGNOSE utility reports are most meaningful to Multivendor Customer Services personnel. Proper maintenance of the error log file is important to maintaining dependable systems and is the responsibility of the system manager. The error log file is used to:

- Identify hardware components with potential for failure. This allows the system manager to move vital work to other components before complete failure. Components so identified can be repaired or replaced prior to causing system downtime.
- Help diagnose a component failure, thereby decreasing the time to repair, and get the equipment back on line.

Dependable systems should include an extra disk in the system configuration. Each day disks should be monitored. When a disk begins to log errors, the files on that disk should be moved to the spare disk and Multivendor Customer Services called to repair the failing disk.

If the site cannot afford to maintain a spare disk, the same thing can be done by moving files to other disks or by making decisions about the relative importance of files. For instance, a large data file that is not used until the end of the month might be archived on tape and the disk space freed for files from the failing disk that are required sooner.

---

**Example: Display from DIAGNOSE/FULL (Alpha)**

The following is the output from an DIAGNOSE/FULL command.

```
***** ENTRY 1 *****
Logging OS          1. OpenVMS
System Architecture  2. Alpha
OS version          V7.2
Event sequence number 1583.
Timestamp of occurrence 18-JAN-1999 09:21:18
System uptime in seconds 58004.
Error mask          x00000000
Flags               x0001 Dynamic Device Recognition
Host name           COGENT

Alpha HW model      DEC 3000 Model 400
System type register x00000004 DEC 3000
Unique CPU ID       x00000002
mpnum               x000000FF
mperr               x000000FF

Event validity      -1. Unknown validity code
Event severity      -1. Unknown severity code
Entry type          100.
Major Event class   3. IO Subsystem
IO Minor Class      1. MSCP
IO Minor Sub Class  5. Logged Message
---- Device Profile ----
Vendor
Product Name        RAID 0 - Host Based
Unit Name           COGENT$DPA
Unit Number         10.
Device Class         x0001 Disk

---- IO SW Profile ----
VMS DC$_CLASS       1.
VMS DT$_TYPE        175.
```

---- MSCP Logged Msg ----

Logged Message Type Code      22. RAID Message

RAID Event Type                8. Remove Member

Distinguished Member         0.

Member Index                 1.

RAID Urgency                 4. Global Disk Error

RAID Status                 x00180009 Bit 00 - Reduced

                             Bit 03 - Striped

                             Bit 19 - FE Dis FE

                             Bit 20 - BC Buff Copy Off

RAIDset Name                 KGB

\*\*\*\*\*

## Operator Log File

### Location

The operator log file is located in the SYSS\$MANAGER directory.



---

**Note**

The definitions in SYLOGICALS.COM affect the OPCOM process. This is one reason why the system logical names should not be defined in SYSTARTUP\_VMS.COM.

---

### Messages Recorded in the Operator Log File

The following types of messages are recorded in the operator log file:

- Initialization of the operator log file
- Status reports for devices attached to the system
- Operator terminals enabled and disabled
- Volume mounts and dismounts



---

**Note**

These messages do not show up unless operator assistance is needed.

---

- User requests and operator replies
- Changes to system parameters through the System Generation utility
- Security alarm messages
- DECnet status messages

### Searching the Operator Log

You can use the DCL SEARCH command to search one or more files for specified strings and display the lines containing those strings.

This command is especially useful when searching for a problem.

- Format:  
**SEARCH file-specification[,,,] search-string[,,,]**
- Parameters:
  - *file-specification* specifies one or more files to be searched.
  - *search-string* specifies the character string to be located.

An example is the use of the command to search OPERATOR.LOG for messages concerning the job controller or queue manager.

```
$ SEARCH SYS$MANAGER:OPERATOR.LOG/WINDOW=5 -  
_$ QUEUE_MANAGE, JOB_CONTROL
```

The qualifier /WINDOW=*n* specifies the number of lines to be displayed. *n*=5 is an appropriate value.

## System Bugchecks

### Types of Bugchecks

When the OpenVMS operating system detects an internal inconsistency, such as a corrupted data structure or an unexpected exception, it generates a *bugcheck*.

- If the inconsistency is not severe enough to prevent continued system operation, the bugcheck is nonfatal and results in an error log entry.
- If the error is serious enough to jeopardize system operation and data integrity, a fatal bugcheck is generated. This typically results in aborting normal system operation (crash), recording the contents of memory to a dump file for later analysis, and rebooting the system.



#### Note

A bugcheck generates a call to a system routine that handles severe problems.

The OpenVMS glossary defines bugcheck as an internal inconsistency that OpenVMS cannot resolve. If a nonfatal bugcheck is declared, an error log entry is made. If the system cannot continue to run, a fatal bugcheck is declared and the system produces a crash dump.

Machine checks are the hardware equivalent of bugchecks.

---

## System Dump File

### Contents of the System Dump File

When the operating system fails, the OpenVMS operating system writes to the system dump file the contents of the following:

- Error log buffers
- Processor registers
- Physical memory



---

**Note**

The crashing system tries to save as much context information as possible before completely crashing. You can use this dump file to analyze why the system crashed. For example, the system manager can often examine the code that was being executed just before the crash, including the line that caused the crash.

---

### Uses for the System Dump File

Use the system dump file to:

- Investigate the cause of a system crash
- Investigate system hangs

### Enabling the System Dump File

The OpenVMS operating system writes the contents of memory into the system dump file when the operating system fails, provided:

- The system parameter DUMPBUG is set.
- There is adequate space for the dump file.



---

**Note**

DUMPBUG enables or disables the writing of error log buffers and memory contents to SYS\$SYSTEM:SYSDUMP.DMP when a fatal bugcheck occurs. This parameter should be set to 0 only when the executive is being debugged.

It is important to emphasize that the system dump overwrites either SYSDUMP.DMP or the paging file. It does not use the regular RMS services to create a new file or to expand an existing file.

The system manager must plan ahead and have an existing file large enough to be overwritten.

---

### Example: Directory Listing of System Dump File

The following example shows the directory listing of a system dump file.

```
VRROOM> DIRECTORY/FULL SYS$SYSTEM:SYSDUMP.DMP
```

```
Directory SYS$SYSROOT:[SYSEXE]
```

```
SYSDUMP.DMP;1      File ID: (2318,2,0)
```

```
Size: 327689/327690 Owner: [SYSTEM]
```

```
Created: 01-DEC-1998 14:09:11.93
```

```
Revised: 15-DEC-1998 14:12:04.66 (3)
```

```
Expires: <None specified>
```

```
Backup: <No backup recorded>
```

```
Effective: <None specified>
```

```
Recording: <None specified>
```

```
File organization: Sequential
```

```
Shelved state: Online
```

```
File attributes: Allocation: 327690, Extend: 0, Global buffer count: 0
```

```
    No version limit, Backups disabled, MoveFile disabled
```

```
Record format: Undefined, maximum 0 bytes, longest 0 bytes
```

```
Record attributes: None
```

```
RMS attributes: None
```

```
Journaling enabled: None
```

```
File protection: System:RWED, Owner:RWED, Group:RW, World:
```

```
Access Cntrl List: None
```

```
Total of 1 file, 327689/327690 blocks.
```



#### Note

Use this directory listing to familiarize the student with the system dump file. One issue we will be discussing is backing up the dump file. Remember that a system crash overwrites the existing dump file. If you want to refer to the original dump again, you must restore it from tape.

By default, the system dump file is set NOBACKUP. To use the Backup utility to copy the system dump file, it is necessary to use the /IGNORE=NOBACKUP qualifier.

## Where Is the Dump File Written?

The file SYSS\$SYSTEM:SYSDUMP.DMP is furnished as an empty file in the OpenVMS software distribution kit.

When the system fails, OpenVMS overwrites the contents of the previous dump file. If the system dump file is too small, all of the memory contents are not saved.

If there is no SYSDUMP.DMP, the system writes the contents of memory to the system paging file SYSS\$SYSTEM:PAGEFILE.SYS.



---

### Note

How is a dump file produced when the system crashes due to power failure? The dump file is only produced if the CPU has a backup battery power supply sufficient to allow the dump file to be produced. Most CPUs do have this power supply.

Mention that it is not often necessary to have a crash dump after a power failure because we know what caused the system to crash.

It is important that space is allocated for the expected size of the dump file before anything is written to the file. When the system crashes, the dump is overwritten. Remember the system is going down, not executing elaborate code to extend a file.

On most systems when the system dump file is too small, the system's page table is not included in the dump.

The dump file is only written to the page file on the system disk. When a dump is present in the page file, the SHOW MEMEORY/FILES command places the letter D next to the total size field.

---

## Saving the System Dump File

If you do not save the contents of the system dump file by copying the contents to another file, it will be overwritten the next time the system fails.

- Use the System Dump Analyzer (SDA) utility COPY command to save the contents of the system dump file in a file with another name.
- Format:  
SDA> COPY output-file-specification
- Example:  
SDA> COPY SYS\$CRASH:SAVEDUMP.DMP
- To save disk space, back up the copy of the dump file to tape and archive. Do **not** delete the original dump file.



---

### Note

It is very important that the students understand how and why to save a copy of the dump file. Copies of the file should be made for the following reasons:

- In a situation where the system is crashing frequently, valuable information can be lost if the system dump file is not saved. Often when the system crashes and reboots successfully, the situation is ignored. Only after it has happened multiple times does anyone start investigating.
- The system manager may want to give less privileged users access to the SAVEDUMP.DMP for analysis purposes; for example, a Compaq representative.
- SYSDUMP.DMP is set no backup. When the SDA command COPY is used to create a file to save the dump file, the new file is not set NOBACKUP and will be included by the Backup utility without using the /IGNORE=NOBACKUP qualifier.

You should emphasize two points:

- A crash dump overwrites the dump file.
- In the event of frequent system crashes, archived copies of the dump file are needed.

Customers are often worried about conserving disk space.

When looking at this figure, discuss disk space. The system manager may want to archive the SAVEDUMP.DMP to tape and purge the directory. The SYSDUMP.DMP must not be deleted. If SYSDUMP.DMP is deleted and the system crashes again, the dump file will be written to the paging file. This causes problems if the system manager does not expect the dump file to be written to the paging file. A discussion of writing the dump file to the paging file is coming up.

---

## SDA Commands at System Startup

SDA commands included in the site-specific startup procedure save the dump file and display crash information.

When invoked in the site-specific startup procedure, SDA executes the specified commands only if the system is booting immediately after a system failure.



---

**Note**

When invoked in the site-specific startup procedure, SDA executes the specified commands only if the system is booting immediately after a system failure. SDA examines a flag in the dump file's header that indicates whether it has already processed the file. If the flag is set, SDA merely exits. If the flag is clear, SDA executes the specified commands and sets the flag. This flag is clear when the operating system initially writes a crash dump, except for those resulting from an operator-requested shutdown (for instance, SYS\$SYSTEM:OPCCRASH.COM or SYS\$SYSTEM:SHUTDOWN.COM).

---

### Example: SDA Commands in System Startup Procedure

Include the following commands in the system startup command procedure to be executed only upon initialization following a crash:

```
$!  
$! Print dump listing if system just failed.  
$!  
$ ANALYZE/CRASH_DUMP SYS$SYSTEM:SYSDUMP.DMP  
COPY SYS_MAN_EXTRA:SAVEDUMP.DMP ! Save the dump file  
SET OUTPUT SYSDUMP.LIS ! Create a listing file  
SHOW CRASH ! Display crash info  
SHOW STACK ! Show current stack  
SHOW SUMMARY/IMAGE ! List all active processes  
SHOW PROCESS/PCB/PHD/REG ! Display current process  
SHOW SYMBOL/ALL ! Print system symbol table  
EXIT  
$!
```



---

**Note**

The lines after the copy command cause the system to display crash information in a list file.

Since for some customers the time it takes to reboot the system is an issue, in these cases you might want to point out the following.

- These steps are not necessary.
  - They take time if they are placed in SYSTARTUP\_VMS.COM.
-

## Calculating the Correct Size for the System Dump File

To calculate the correct size for a full system dump file:

- Enter the DCL SHOW MEMORY command to determine the total size of physical memory in pages on your system (M).

```
$ SHOW MEMORY
      System Memory Resources on 15-JAN-1999 17:32:05.00
Physical Memory Usage (pages):  Total  Free  In Use
Modified
Main Memory (160.00Mb)    327680  227196  92080  8404
.
.
.
```

- Determine the number of error log buffers from the setting of the system parameter ERRORLOGBUFFERS (N) and the number of error log buffer pages from the setting of the system parameter ERLBUFFERPAGES (P).

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SHOW ERRORLOGBUFFERS
Parameter Name  Current Default  Min.   Max.   Unit   Dynamic
-----
ERRORLOGBUFFERS      4      4      2     64 Buffers
SYSGEN> SHO ERLBUFFERPAGES
Parameter Name  Current Default  Min.   Max.   Unit   Dynamic
-----
ERLBUFFERPAGES      2      2      2    32 Pages
```

- Use the formula:

$$SIZE = M + (N * P) + 2$$

- Example:

$$SIZE = 327680 + (4 * 2) + 2 = 327,690 \text{ pages}$$

16 disk blocks per page on Alpha systems.

1 disk block per page on VAX systems.



### Note

This is only important when using the paging file to capture the system dump file. You need to make sure that you have enough space in the paging files on the system disk to hold the entire dump file and room for the paging of boot. The space may be in multiple paging files, but all must be on the system disk. ERRORLOGBUFFERS specifies the number of error log buffers. ERLBUFFERPAGES dictates how many pages are allocated for each of the error log buffers.

Multiplying the two provides the memory demands (and the demands placed on the system dump file) for error log storage.

This calculation is not clear in the OpenVMS documentation set. See AUTOGEN.COM source code for an explanation.

## Capturing Dump Files on Systems with Limited Disk Space

### Using the Paging File for the System Dump

If the SYSS\$SYSTEM:SYSDUMP.DMP does not exist, the OpenVMS operating system writes the physical memory dump into SYSS\$SYSTEM:PAGEFILE.SYS, which overwrites the contents of the paging file.

Set the system parameter SAVEDUMP (1=save) to save the information in the paging file related to the system crash when the system is rebooted.

- The paging file must be at least 1000 blocks larger than the minimum size for the dump file.
- On reboot, free the space occupied by the system dump file in the paging file or the system may hang.

Sample output from the SYSGEN SHOW SAVEDUMP command:

```
SYSGEN> SHOW SAVEDUMP
```

| Parameter Name | Current | Default | Min. | Max. | Unit    | Dynamic |
|----------------|---------|---------|------|------|---------|---------|
| SAVEDUMP       | 0       | 0       | 0    | 1    | Boolean |         |

The DCL COPY command can also be used to copy the system dump file from the paging file to another file. This, however, does not free the space in the paging file and may cause the system to hang.

## Copying the System Dump File from SYS\$SYSTEM: PAGEFILE.SYS

When the system dump file is written to the paging file, you must copy the file to another location when the system is rebooted to free the paging file space.

- Use SDA commands in the site-specific STARTUP.COM file.
- This frees the space in the paging file and allows the system to continue.

```
$ ANALYZE/CRASH_DUMP SYS$SYSTEM:PAGEFILE.SYS
.
.
.
COPY SYS$SYSTEM:SAVEDUMP.DMP    ! Save the dump file
```

```
! and release space in page file
EXIT
```



### Note

The page file should be as large as the sum of what AUTOGEN recommends for PAGEFILE and DUMPFIL. Otherwise, a minimum boot may be required.

---

## Systems with Large Memory and Small Disk Space

On systems with large memory and very limited disk space, saving the entire dump file may be impossible.

Set the system parameter DUMPSTYLE to 1 to set AUTOGEN to create a dump file large enough to save only a subset of the memory dump.



---

### Note

**DUMPSTYLE(D)** specifies the method of writing system dumps. Specify one of the following values.

- 0 The entire contents of physical memory is written to the dump file. This is the default.
- 1 Selected portions of memory are written to the dump files as space permits.

To find out how much space is needed for DUMPSTYLE=1, run AUTOGEN from GETDATA to TESTFILES.

If the dump file is large enough to contain the entire contents of physical memory, DUMPSTYLE should be set to 0 to ensure that all of the information in physical memory is saved in the dump file.

If you have a large memory system and the dump file is too small to contain a complete system dump of physical memory, set DUMPSTYLE to 1 to specify a partial memory dump.

To delete the dump file, rename the dump file to SAVEDUMP.OLD and reboot the system.

See SYSGEN>HELP SYS DUMPSTYLE for information on compressed and DOSD (Dump Off System Disk) values of DUMPSTYLE.

---

## Using the System Dump Analyzer to Analyze a Crash

### Overview

The System Dump Analyzer (SDA) utility extracts information from the system crash dump file (usually SYSSYSTEM:SYSDUMP.DMP) and presents it in a usable format.

### Accessing the SDA (Alpha)

To access the System Dump Analyzer utility, type ANALYZE/CRASH *dumpfile* at the DCL prompt.

### Example: Displaying Crash Information

Use the SHOW CRASH command at the SDA> prompt to display information about the most recent crash.

```
OpenVMS (TM) Alpha system dump analyzer
...analyzing a selective memory dump...
%SDA-W-NOTSAVED, SPT window not saved in the dump file
Dump taken on 15-JAN-1999 15:40:51.03
❶ INVEXCEPTN, Exception while above ASTDEL
SDA> SHOW CRASH
System crash information
-----
Time of system crash: 15-JAN-1999 13:47:10.54
Version of system: OpenVMS (TM) Alpha Operating System,
Version V7.2
System Version Major ID/Minor ID: 3/0
VMScluster node: VRR00M, a DEC 3000 Model 500
Crash CPU ID/Primary CPU ID: 00/00
Bitmask of CPUs active/available: 00000001/00000001
CPU bugcheck codes:
    CPU 00 -- INVEXCEPTN, Exception while above ASTDEL
-----
System State at Time of Exception
-----
Exception Frame:
-----
    R2 = FFFFFFFF 80885FB0 SYS$PEDRIVER_NPRW+05DB0
    R3 = 00000000 0000000A
    R4 = 61616161 61616161
    R5 = FFFFFFFF 80A556C0
    R6 = 66666666 66666666
    R7 = 77777777 77777777
    PC = FFFFFFFF 809F37CC
    PS = 20000000 00000804
❷ 809F37BC:  ADDQ    SP, #X10, SP
    809F37C0:  RET     R31, (R26)
    809F37C4:  BIS     R31, R31, R0
    809F37C8:  BIS     R31, #X0A, R28
    PC => 809F37CC:  STL     R28, (R0)
```

```
809F37D0: LDQ    R3, #X0008(SP)
809F37D4: ADDQ    SP, #X10, SP
809F37D8: RET     R31, (R26)
809F37DC: 61616161
PS =>
MBZ SPAL  MBZ IPL VMM MBZ CURMOD INT PRVMOD
0 20 000000000000 08 0 0 KERN 1 KERN
.
.
.
```

### Example Notes: Displaying Crash Information

- ❶ The reason for the crash
- ❷ Code that was executing at the time of the crash. PC points to the current instruction.



---

**Note**

On an OpenVMS VAX system, information on the code executing at the time of the crash is not given.

---

## Example: SDA Command SHOW SUMMARY/IMAGE

The following example shows the output of the SDA SHOW SUMMARY/IMAGE command.

```
$ ANALYZE/CRASH SYS$SYSTEM:SYSDUMP.DMP
OpenVMS (TM) VAX System dump analyzer
Dump taken on 15-JAN-1999 14:52:44.24
NETDLLERR, DECnet Datalink Layer detected a fatal error
SDA> SHOW SUMMARY/IMAGE
Current process summary
-----
Extended Indx Process name Username State Pri PCB  PHD Wkset
-- PID --
40800041 0001 SWAPPER      HIB  16 813545A0 81354400 0
40800044 0004 STARTUP     SYSTEM PFW  5 817EAE00 81E6AC00 777
      DSA0:[SYS17.SYSCOMMON.][SYSEXE]VMOUNT.EXE;1
40800046 0006 CONFIGURE  SYSTEM HIB  9 817ECE00 81E22600 182
      DSA0:[SYS17.SYSCOMMON.][SYSEXE]CONFIGURE.EXE;1
SDA>
```



### Note

You can take time now to let the students explore SDA. The online help explains the function of each of the SHOW options.

If you need a system crash dump file, you can use the following procedure to crash the system:

1. Enter console mode (press the HALT button on most CPUs)
2. At the >>> prompt, type D/G F FFFFFFFF (deposit -1 to the program counter)
3. At the >>> prompt, type C (continue)

## Crash Log Utility Extractor

### Overview

The Crash Log Utility Extractor (CLUE) is a utility for managing and examining system dump files. A history file lets you store information about previous system crashes.

### Accessing CLUE

To access CLUE on an OpenVMS Alpha system:

1. Start the System Dump Analyzer (SDA) utility.
2. Enter CLUE at the SDA> prompt.

To access CLUE on an OpenVMS VAX system enter MCR CLUE at the DCL prompt.

### Example: Accessing CLUE

The following example shows how to access CLUE on an OpenVMS Alpha system:

```
$ ANALYZE/CRASH SYS$SYSTEM:SYSDUMP.DMP
```

```
OpenVMS (TM) Alpha system dump analyzer
...analyzing a selective memory dump...
```

```
Dump taken on 15-JAN-1999 13:47:10.54
INVEXCEPTN, Exception while above ASTDEL
```

```
SDA> CLUE
```

```
CLUE Alpha - Type CLUE HELP for further Information
```

```
CLUE commands: CANASTA, CLEANUP, CONFIG, CRASH, DEBUG,
ERRLOG, HELP, HISTORY,
```

```
          KPB, MCHK, MEMORY, PROCESS, SCSI, STACK, VCC, XQP
```

```
SDA>
```

If there is no valid dump, SDA returns an error message and drops back to the DCL prompt.

## Displaying CLUE History Entries

On an OpenVMS VAX system, use the /DISPLAY command in CLUE to list the entries in the history file.

There is no equivalent command on an OpenVMS Alpha system.

```
$ MCR CLUE
```

| # | Node   | Time              | Type     | Process Name | Module  | Offset |
|---|--------|-------------------|----------|--------------|---------|--------|
| 1 | BROWNY | 9-DEC-1998 13:56  | OPERATOR | SYSTEM       | UNKNOWN | 0      |
| 2 | BROWNY | 16-DEC-1998 13:57 | OPERATOR | SYSTEM_1     | UNKNOWN | 0      |
| 3 | BROWNY | 16-DEC-1998 07:12 | OPERATOR | SYSTEM       | UNKNOWN | 0      |
| 4 | BROWNY | 15-DEC-1998 16:23 | OPERATOR | _RTA1:       | UNKNOWN | 0      |
| 5 | BROWNY | 15-DEC-1998 15:36 | OPERATOR | SYSTEM       | UNKNOWN | 0      |

```
CLUE> /DISPLAY
```

## Displaying Crash Information Using CLUE (OpenVMS VAX)

From the CLUE\_DISPLAY> prompt, use the SHOW CRASH command to display information about a selected system crash.

```
CLUE_DISPLAY >SHOW CRASH 1
```

```
#1 CRASH DATA:
```

|                                  |                         |
|----------------------------------|-------------------------|
| BUGCHECK TYPE                    | INVEXCEPTN              |
| TIME                             | 10-DEC-1998 14:56:16.25 |
| VMS VERSION                      | V7.2                    |
| Crash CPU ID / Primary CPU ID    | 0/0                     |
| Number of CPUs                   | 1                       |
| Bitmask of CPUs active/available | 00000001/00000001       |
| Bitmask of bugchecking CPUs      | 00000001                |
| Current Process Name             | NONE                    |

Press RETURN for more

```
CLUE_DISPLAY >
```



### Note

The CLUE display continues to give register information. On an OpenVMS VAX system, better information is given by the SDA ANALYZE CRASH command.

## Displaying Crash Information Using CLUE (Alpha)

Use the CLUE CRASH command at the SDA prompt to display information about the most recent crash.

SDA> CLUE CRASH

Crashdump Summary Information:

-----

Crash Time: 10-JAN-1999 13:47:10.54

Bugcheck Type: INVEXCEPTN, Exception while above ASTDEL

Node: VRR00M (Clustered)

CPU Type: DEC 3000 Model 500

VMS Version: V7.2

Current Process: NULL

Current Image: <not available>

Failing PC: FFFFFFFF 809F37CC

Failing PS: 20000000 00000804

Module: <not available>

Offset: 00000000

Boot Time: 3-JAN-1999

System Uptime: 2 02:48:48.54

Crash/Primary CPU: 00/00

Saved Processes: 24

Pagesize: 8 KByte (8192 bytes)

Physical Memory: 96 MByte (12288 PFNs)

Dumpfile Pagelets: 107821 blocks

Dump Flags: olddump,writecomp,errlogcomp,dump\_style

EXESGL\_FLAGS: poolpging,init,bugdump

Stack Pointers:

KSP = FFFFFFFF 825F1B68 ESP = FFFFFFFF 825F3000 SSP = FFFFFFFF 825EF000

USP = FFFFFFFF 825EF000

General Registers:

R0 = 00000000 00000FBA R1 = 00000000 00000000 R2 = FFFFFFFF FFFFFFFD

R3 = FFFFFFFF 8082F660 R4 = FFFFFFFF 825F1BC0 R5 = FFFFFFFF 825F1D28

R6 = FFFFFFFF 825F1D80 R7 = 20000000 00000804 R8 = 88888888 88888888

R9 = 99999999 99999999 R10 = AAAAAAAAAA AAAAAAAAAA R11 = BBBBBBBB BBBBBBBB  
R12 = CCCCCCCC CCCCCCCC R13 = DDDDDDDD DDDDDDDD R14 = EEEEEEEE EEEEEEEE  
R15 = EEEEEEEE EEEEEEEE R16 = 00000000 000001CC R17 = FFFFFFFF 825F1BC0  
R18 = FFFFFFFF 825F1D80 R19 = FFFFFFFF 80804200 R20 = FFFFFFFF 8096E4B4  
R21 = FFFFFFFF 8096E4B7 R22 = FFFFFFFF 809F378C R23 = FFFFFFFF 80822340

Press RETURN for more.

SDA>



---

**Note**

The SDA SHOW CRASH command seems to give more usable information, both on the OpenVMS VAX and Alpha systems.

---

On the Alpha, the CLUE CONFIG command can be useful as it gives the sort of information that SHOW DEVICE would return from the console. For example:

SDA> CLUE CONFIG

System Configuration:

-----

System Information:

System Type DEC 3000 Model 500 Primary CPU ID 00  
Cycle Time 6.6 nsec (150 MHz) Pagesize 8192 Byte

Memory Configuration:

| Cluster | PFN | Start | PFN Count | Range (MByte) | Usage   |
|---------|-----|-------|-----------|---------------|---------|
| #00     | 0   | 256   | 0.0 MB    | - 2.0 MB      | Console |
| #01     | 256 | 12032 | 2.0 MB    | - 96.0 MB     | System  |

Per-CPU Slot Processor Information:

|               |              |                        |                       |
|---------------|--------------|------------------------|-----------------------|
| CPU ID        | 00           | CPU State              | rc,pa,pp,cv,pv,pmv,pl |
| CPU Type      | EV4 P2.0/2.1 | Halt PC                | 00000000 00000000     |
| PAL Code      | 5.41         | Halt PS                | 00000000 00000000     |
| CPU Revision  | F001         | Halt Code              | 00000000 00000000     |
| Serial Number | .....        | Bootstrap or Powerfail |                       |

Press RETURN for more.

**Note**

This is only available on the Alpha system.

---

## System Dump Analyzer Features

The System Dump Analyzer (SDA) has a new command, new features to existing commands, display changes, and other command changes.

### New Features for Existing Commands

The following commands have new features and qualifiers:

- **SHOW CLUSTER /NODE=name**
- **SHOW CONNECTIONS /SYSAP /NODE=name**
- **SHOW LAN/CLIENT=name /DEVICE=name /UNIT=name**
- **SHOW LOCK/CACHED**
- **SHOW PAGE\_TABLE /GPT /PT /SOS1 /S2 /SPTW**
- **SHOW PFN\_DATA/ADDRESS=<PFN-entry-address>**
- **SHOW PROCESS/ADDRESS=pcb-address /INDEX=n /RDE=id /SEMAPHORE /THREADS /WORKING\_SET\_LIST**
- **SHOW RESOURCE/CACHED**
- **SHOW SUMMARY/THREAD**

### Other SDA Command Changes

Other SDA command changes include:

- **Page table requests**  
Certain page table requests require SDA to cover large virtual address spaces in small increments looking for valid page table entries. This will noticeably slow your system. Try using a more narrow range of interest to prevent this.
- **Bugcheck usability changes**  
An attempt has been made to make the dumps that were partially written when a system crashed more usable. This reduces the number of times SDA signals an incomplete dump file.
- **Bugcheck transition pages**  
BUGCHECK no longer forces pages of virtual memory to be written into a dump file. Page table entries are not altered, and transition pages are treated as holes in address space that is inaccessible by SDA.

### New SDA Command SET FETCH

The new SDA command **SET FETCH** sets the default size of data manipulated by the EXAMINE and the EVALUATE commands. SET FETCH has no qualifiers.

## SDA Display Changes

The following commands have display changes:

- **SHOW CRASH**  
This command's exception display is changed to use the new 64-bit signal array.
- **SHOW PFN\_DATA**  
The output from this command is converted to a two-line format to accommodate the expansion of virtual addresses from 32 bits to 64 bits. The expansion was also needed to display additional information.
- **SHOW PAGE\_TABLE**
- **SHOW PROCESS/WORKING SET**
- **SHOW PROCESS/PHD**
- **SHOW PROCESS/PROCESS\_SECTION\_TABLE**

## VAX Dump File Process

The OpenVMS VAX SDA now uses RMS file access to process the dump file. (Previously, it mapped the dump file into the working set.)

This change, however, has made some of the SDA commands noticeably slower.

A value of 16,000 is now sufficient for the system parameter virtual page count VIRTUALPAGECNT to analyze any size dump.

## Summary

### Concepts

#### Dependable Systems

- A dependable computing system is one that can be counted on to provide services to users when those services are needed.
- System problems can be caused by
  - Environmental problems
  - Hardware failure
  - Software failure
  - Human error

#### Troubleshooting Methodology

- A suggested troubleshooting method consists of nine steps:
  1. Define the problem.
  2. Gather information about the problem.
  3. Analyze and interpret the information.
  4. Propose a solution, or refer the problem to the responsible person.
  5. Verify the solution.
  6. Clean up any parameters or files created for testing.
  7. Reverify the solution.
  8. Document the problem and archive the solution.
  9. Investigate methods of preventing the problem from occurring again.

#### OpenVMS System Messages

- Most system messages follow a standard format.
- Online help provides more information on error messages.

#### OpenVMS Troubleshooting Tools

- The system dump file captures the contents of memory when the operating system experiences a fatal bugcheck.
- Nonfatal bugchecks write messages to the error log file.

## Commands

### OpenVMS System Messages

#### **HELP /MESSAGE ident**

DCL command to access a system message description.

### OpenVMS Troubleshooting Tools

#### **SHOW ERROR**

DCL command to display device errors.

#### **DIAGNOSE file-specification**

DCL command to selectively report contents of an error log file.

#### **SEARCH file-specification[,,,] search-string[,,,]**

DCL command to search one or more files for specified strings.

#### **ANALYZE/CRASH\_DUMP file-specification**

DCL command to invoke the System Dump Analyzer utility.

#### **SHOW MEMORY**

DCL command to display memory usage.

#### **SHOW parameter-name**

SYSGEN command to display parameter values.

## Chapter Terms

### **availability**

The percentage or amount of scheduled time that a computing system provides application service.

### **bugcheck**

An internal inconsistency that the OpenVMS operating system cannot resolve. If a nonfatal bugcheck is declared, an error log entry is made. If the system cannot continue to run, a fatal bugcheck is declared and the system produces a crash dump.

### **fault tolerance**

The ability of a computing system to withstand faults and errors while continuing to provide the required services. A high level of failover transparency is usually implied by this term.

### **mean-time-between-failures (MTBF)**

The average time that passes before a computing component fails such that remedial action is required.

### **reliability**

The ability of a computing system to operate without failing.



---

# ***System Initialization Problems***

## ***Module 2***

## Introduction

During routine system operation, you may shut down and restart your system to perform various maintenance functions. User modifications to files on the system disk, hardware failures, etc., can cause the startup procedure to fail.

To correct the problem, you should be able to use the emergency startup procedures to reboot the system.

This module discusses how to identify problems when booting a standalone system. Cluster problems are discussed in a Module 4.

This module also reinforces the idea of change management, which is one of the key concepts in this course.



---

**Note**

Time

Materials: 3 hours

Practice: 4 hours

---

## Objectives

To handle and prevent system initialization problems, you should be able to:

- Describe system initialization concepts.
- Use emergency system initialization procedures and the standalone Backup utility.
- List the steps to follow in solving system initialization problems.
- Avoid system initialization problems.

## Resources

- *OpenVMS VAX Version x.x Upgrade and Installation Manual*
- *OpenVMS Alpha Version x.x Upgrade and Installation Manual*
- *OpenVMS Alpha Internals and Data Structures*
- *VAX/VMS Internals and Data Structures*

# System Initialization Concepts

## Overview

System initialization, or booting, is the process of loading system software from the system disk into processor memory. Part of the initialization process is specific to the type of processor. Other parts are common across the entire family of computers. This section focuses on the common parts of the initialization process.

System initialization takes place in a series of steps or stages as different programs are executed. As each program executes, messages that provide information on how far the initialization process has progressed are written to the console.

If the initialization process fails, you must determine at what stage it failed so you can begin the troubleshooting process.

To troubleshoot system initialization problems, you should be able to describe the phases of system initialization which are discussed in this module.

## Phases of OpenVMS System Initialization

### Initialization Steps

OpenVMS system initialization occurs in a series of steps.

1. The console subsystem is CPU specific. Regardless of its implementation, the console subsystem performs the following tasks:
  - Initializes the CPU
  - Locates physically contiguous good memory
  - Loads the primary bootstrap image into memory and transfers control to it
1. The primary bootstrap program (APB for Alpha CPUs, VMB for VAX CPUs) runs standalone and performs the following tasks:
  - Sizes memory
  - Initializes context for the adapter and device unit (system disk) containing the secondary bootstrap program
  - Loads the secondary bootstrap and transfers control to it
2. SYSBOOT.EXE, the secondary bootstrap program (a file on system disk), runs standalone with memory management disabled and performs the following tasks:
  - Reads the system parameters, by default from [SYSn.SYSEXEX]ALPHAVMSSYS.PAR (VAXVMSSYS.PAR for VAX systems)
  - Lays out the system virtual address space based on the system parameters
  - Loads the system base image, SYS.EXE, into memory
  - Also loads the system device driver, port driver, auxiliary driver, SYS\$CPU\_ROUTINES\_xxyy (SYSLOA on VAX systems), and OpenVMS Cluster code
  - Turns control over to the EXE\$INIT routine in module INIT (EXE\_INIT on VAX systems)
  - Displays the SYSBOOT> prompt during conversational boot

3. EXE\$INIT performs the following initialization tasks that must occur before process context is available:
  - Turns on memory management
  - Invokes initialization routines loaded by SYSBOOT.EXE
  - Initializes both the scheduler and memory management
  - Starts secondary CPUs
  - Runs the routine that starts the swapper
  - Displays the OpenVMS banner OpenVMS <sup>TM</sup> Alpha Operating System, Version V7.2
4. EXE\$SWAPINIT performs the following tasks:
  - Initializes paged pool and the pageable logical name database
  - Creates the SYSINIT process
5. SYSINIT.EXE performs the following tasks:
  - Loads RMS and other loadable images
  - Initializes the OpenVMS Cluster software for an OpenVMS Cluster node
  - Opens the swap and page files and records their extents
  - Opens Files-11 Extended QIO processor (XQP) as a global section
  - Loads the system message file
  - Creates the site-independent startup process
6. The startup process has a full process context and can execute DCL commands. It performs the following tasks:
  - Executes the site-independent SYS\$SYSTEM:STARTUP.COM
  - Creates OPCOM, job controller, and SMISERVER processes
  - Creates systemwide logical names
  - Runs SYSGEN to autoconfigure the I/O database
  - Installs the images specified in the VMSIMAGES.DAT file
  - Enables licenses
  - Executes the site-specific SYS\$MANAGER:SYSTARTUP\_VMS.COM, and so forth
  - Enables logins

**Note**

The purpose of this page is to give the student some familiarity with the overall system boot process. The section contains more information than the students are expected to retain.

Another objective here is to get the students to have respect for these system initialization programs so they do not delete them from the system disk. Customer calls to the CSC (Customer Service Center) indicate that customers, worried about disk space on the system disk, do delete system initialization files. Of course nothing happens until they try to reboot the system several days later. By that time they may have forgotten what they did. This information is reinforced by a laboratory exercise.

- **What does the term load mean?**
- Bring the program into memory.
- **What does turn over control mean?**
- The instruction pointer is set to point to the first instruction in the routine. This will then be the next instruction to execute.
- **What does it mean that APB and VMB run standalone?**
- There is no OpenVMS operating system when it is running: no memory management, no CPU sharing, and so on. APB or VMB is in the boot block.

Also, you might want to mention that OpenVMS postpones an initialization task to as late a stage of initialization as possible. This is because tasks become easier to do as more and more of the operating system becomes available.

For more information see *OpenVMS AXP Internals and Data Structures*. The OpenVMS banner displayed by EXE\$INIT is normally the first message you see. APB, VMB, and SYSBOOT display nothing, except for SYSBOOT> prompt.

Mention that SYSTARTUP\_VMS.COM should be written to display console messages describing what is happening

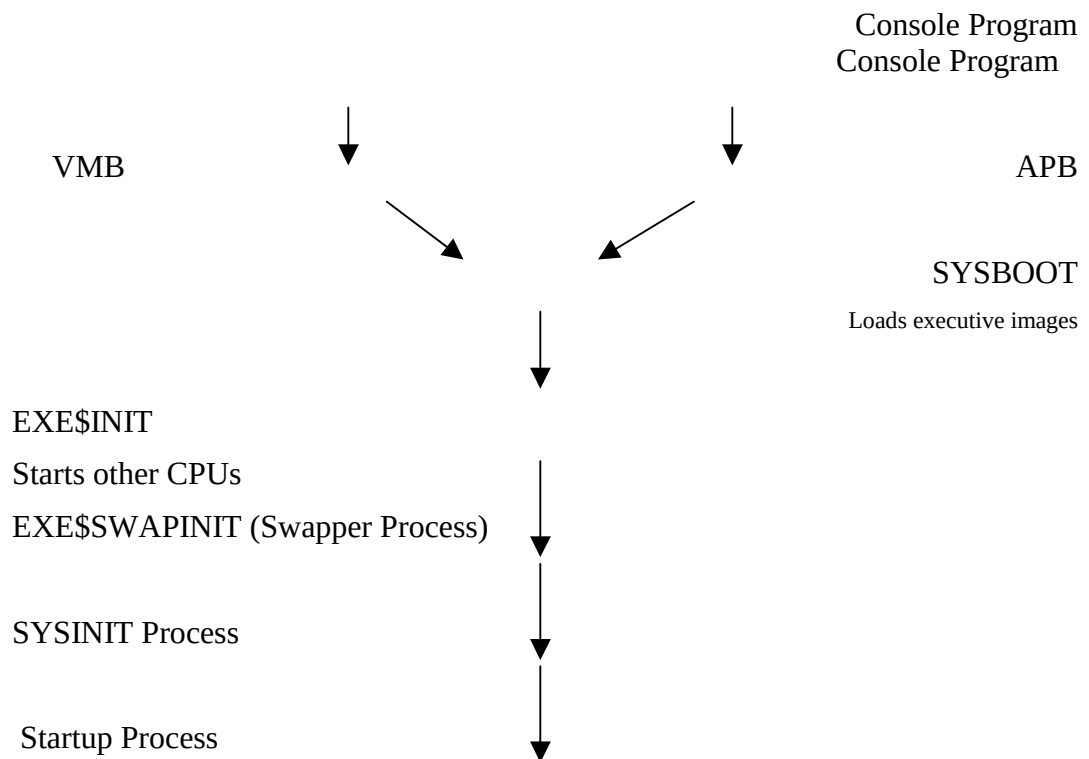
---

## Sequence of Initialization Events

The following figure shows the sequence of OpenVMS initialization steps on different CPUs.

### VAX 6000 Series

### Alpha





---

**Note**

The console subprogram is processor specific. Students should refer to the hardware manual that came with their system to reference the commands for their particular system.

The *OpenVMS System and Network Node Management III* course contains a discussion of the console subprogram. Students should review this material before coming to class.

**SYSBOOT**

If a conversational bootstrap has been requested, SYSBOOT allows the system manager to change system parameters in several ways:

- Use default parameters stored in SYSBOOT (USE DEFAULT)
- Change individual parameters with the SET command
- Load a set of parameters from a parameter file (USE filespec)

The SYS\$CPU\_ROUTINES\_xxyy image is equivalent to the VAX image SYSLOA\_xxyy, where xx is the system type designator and yy is the processor type designator.

**How to Use this Page**

Put this page up on the overhead projector at the beginning of the discussion of this section. As you discuss the various stages of system boot, indicate them on the figure. Then use the figure to illustrate how far system boot has progressed after various types of conversational boot. For example:

- Boot minimum
- Boot OPA0:
- When you see the SYSBOOT prompt

This figure will also be useful when discussing system initialization in the Module 4.

---

## Site-Specific Startup Command Procedures

The site-specific command procedures in order of execution are:

- SYCONFIG.COM
- SYLOGICALS.COM
- SYPAGSWPFILES.COM
- SYSECURITY.COM
- NET\$STARTUP.COM
- SYSTARTUP\_VMS.COM

Use the site-specific command procedures to do the following:

- Start or enable batch and print queues.
- Set terminal speeds and other device characteristics.
- Create site-specific system logical names.
- Install additional privileged and shareable images.
- Load user-written device drivers.
- Mount volumes other than the system disk.
- Load the console driver.
- Start DECnet software.
- Start the software that runs the LAT protocol.
- Produce an error log report.
- Announce system availability.
- Load the operating systems for print servers.

## Site-Specific Startup Dependencies

Follow these rules for the site-specific startup command procedure:

- Start the network before the LAT.
- Start print queues after the LAT.
- Check requirements for layered products.



### Note

The order of these steps may vary greatly from this template.

To see the site-specific startup files, have the students type:

```
Set default sys$startup
Dir sy*.com
```

AUTOSTART Queues are addressed in module 3.

Because this is a time consuming part of the system initialization, it is a good time to suggest that the students should write this command procedure to display console messages to help them monitor progress.

```
$ WRITE SYS$OUTPUT "message text"
```

Some steps may be done using SYSMAN.

Watch dependencies very carefully. You may want to run some startup commands in a batch procedure. Make sure you start a “system” batch queue first.

---

# Emergency System Initialization

## Overview

The startup and login procedures provided by Compaq should always work; however, certain user interventions, or hardware problems can cause them to fail. When this happens, you use the emergency startup procedures to boot the system.

When system files are accidentally deleted or corrupted, you use standalone Backup to restore the system disk and boot the system.

System disk software and hardware problems can make the entire system unavailable.

This section discusses:

- Performing a conversational boot
- Using SYSGEN commands during system initialization
- Performing a minimum system initialization
- Defining the console as the startup procedure
- Using standalone Backup to save and restore the system disk
- Handling system disk problems



### Note

This is the “tools” section of this chapter.

You are standing in front of a system that does not boot properly. You do not have to give up and call for help.

The purpose of this section is to give the system manager some tools for solving system boot problems. Later in this chapter there is a list of steps to take to try to fix the system that do not boot. The lab exercises will use these techniques.

Students should be familiar with all these procedures, although they may not have applied them in a systematic way to solve problems.

---

## Performing a Conversational Boot

### When to Use Conversational Boot

Perform a conversational boot when you want to stop the boot procedure before it completes. Generally this is because of error messages during the boot.

The boot procedure pauses after it loads SYSBOOT.EXE and displays the SYSBOOT> prompt.

### Conversational Boot Options

At the SYSBOOT> prompt, you can enter SYSGEN commands to do the following:

- Look at system parameter values.
- Change system parameter values.
  - Specify a minimum startup.
  - Specify another parameter file.
  - Specify another site-independent startup command procedure.
  - Define the console as the startup procedure so you can control the final stages of the system initialization.
  - Turn on verify.

### Processor-Specific Procedures

The procedures to perform a conversational boot are processor-specific. See the documentation that was shipped with your system to find the procedures for a particular Alpha or VAX computer.

### Examples

#### VAX 6000 series

```
>>> BOOT /R5=1 [device]
```

#### Alpha

```
>>> BOOT -FLAGS 0,1 [device]
```

## Using SYSGEN Commands During System Initialization

### SYSBOOT> Prompt

When you use a conversational boot, you access the SYSBOOT> prompt.

You can use a subset of the SYSGEN commands at the SYSBOOT> prompt.

### SYSGEN Commands Commonly Used at the SYSBOOT> Prompt

| Command     | Description                                                                                                                                                                        |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CONTINUE    | Resumes the boot procedure.                                                                                                                                                        |
| HELP        | Displays a summary of the SYSBOOT commands.                                                                                                                                        |
| SET         | Establishes the value of a system parameter.                                                                                                                                       |
| SET/STARTUP | Sets the name of the system startup command procedure. (The default startup procedure is SYS\$SYSTEM:STARTUP.)                                                                     |
| SHOW        | Displays current, default, maximum, and minimum values for specific system parameters. Displays, with appropriate qualifiers, characteristics of parameters grouped by categories. |
| USE         | Specifies a parameter file to be used as a source of values. (You must enter the entire file specification, including device and directory.)                                       |



#### Note

If you reach the SYSBOOT prompt you know that all the files up to this point are intact. This is good news because you have some useful options.

Students are expected to know how to run SYSGEN and how to use the commands.

However, you should discuss the situation where system parameter changes have fixed the system so that it does not boot. Although you hope the system manager knows about the changes, it does happen that someone changes system parameters, does not reboot the system, and goes on vacation.

To resolve such problems, boot conversationally to the SYSBOOT> prompt and use an alternate parameter file or change the value of a given parameter. Also point out that they can display parameter values to help them find the problem.

The system parameter file is ALPHAVMSSYS.PAR on Alpha systems, and VAXVMSYS.PAR on OpenVMS VAX systems.

**Danger** – Changes you make are written to the system parameter file.

## Performing a Minimum System Initialization

To perform a minimum system initialization, enter the following at the SYSBOOT> prompt:

```
SYSBOOT> SET STARTUP_P1 "MIN"
```

```
SYSBOOT> CONTINUE
```

### Minimum Initialization Operations

A minimum system initialization performs the following operations:

- Starts the processes that control error logging, the job controller, and the operator's log
- Installs known images
- Defines the number of interactive users as 8
- Logs off

A minimum system initialization does not invoke the site-specific startup command procedures, nor does it automatically configure all devices.

A minimum system initialization provides a DCL environment in which you can resolve problems with the site-specific startup command procedure. To continue resolving problems with the site-specific startup command procedure, log in to the system account or other privileged account.



#### Note

Again show the figure of the system programs and show where a minimum boot takes you. Booting the system minimally gets you to the DCL prompt and allows you to resolve problems with the SYSTARTUP\_VMS.COM procedure.

A minimum boot passes a parameter of “MIN” to STARTUP.COM as P1. STARTUP.COM checks this parameter and modifies its actions if P1 is “MIN”.

---

## System Processes at Minimum Boot

The following processes are active after a minimum system initialization:

- SWAPPER
- FASTPATH\_SERVER
- ERRFMT
- OPCOM
- JOB\_CONTROL

## Resetting STARTUP\_P1

The value of STARTUP\_P1 is saved and affects future boot operations. After you resolve the problems with the startup command procedure and boot the system, you must reset the value to NULL.

To reset the value to NULL, run SYSGEN.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET STARTUP_P1 ""
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```



---

**Note**

This block is to acquaint the student with the state of the system at the time of a minimum boot.

---

## Using the AUTOGEN.PAR File for System Parameters

If changes to system parameters have been made without using AUTOGEN, and the system does not initialize, you may be able to avoid restoring the system disk by using the AUTOGEN.PAR file.

Perform a conversational boot, and at the SYSBOOT> prompt, enter the command USE SYS\$SYSTEM:AUTOGEN.PAR to instruct the system to use this file.

```
SYSBOOT> USE SYS$SYSTEM:AUTOGEN.PAR
SYSBOOT> SET STARTUP_P1 "MIN"
SYSBOOT> CONTINUE
```

## Using the Default System Parameters

If you suspect that the reason your system does not initialize is because of changes to system parameters, try using the default system parameters to initialize the system.

Perform a conversational boot, and at the SYSBOOT> prompt enter the command USE DEFAULT to instruct the system to use all the default system parameters. Then boot the system minimally.

```
SYSBOOT> USE DEFAULT
```

```
SYSBOOT> SET STARTUP_P1 "MIN"
```

```
SYSBOOT> CONTINUE
```

In most cases, this allows the system to initialize.



### Note

Warning! If the student's have not been using AUTOGEN, AUTOGEN . PAR may be from the install of OpenVMS or the last product that forced AUTOGEN to run.

If the students have not been using AUTOGEN, installing a product that forces AUTOGEN to run may be the cause of their booting problem. **Always change SYSGEN parameters by running AUTOGEN.**

If the problem was caused by AUTOGEN, the students can use the following to solve the problem:

```
SYSBOOT> USE SYS$SYSTEM:ALPHAVMSSYS.OLD or VAX  
VMSSYS.OLD.
```

---

## Defining the Console as the Startup Procedure

### Overview

You can resolve problems with the site-independent startup command procedure by defining the console as the startup procedure.

You can use this technique to:

- Reset a forgotten system password
- Perform a full backup or restore operation
- Use an alternate startup file

### Defining the Console as the Startup Procedure

To define the console as the startup device, boot conversationally until you reach the SYSBOOT> prompt, and then enter:

1. SYSBOOT> SHOW/STARTUP
2. SYSBOOT> SET/STARTUP OPA0:
3. SYSBOOT> CONTINUE

The original startup setting is displayed and changed to be the console. The boot process continues through the SYSINIT process, and you will see the DCL prompt. The site-independent startup command procedure has not been executed.

This is an important problem-resolution method, often called “Booting OPA0.”

Be sure to go back to the figure of the initialization events and show where you are when you boot OPA0:

### Restoring the System Startup

After the problems have been resolved, restore the site-independent startup command procedure as the startup command procedure:

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

To run the site-independent startup command procedure:

```
$ reboot ! if that symbol has been defined
```

## Using a Full-Screen Editor

If you use a workstation as a console, you can use an editor only in line mode when the system is booted to use OPA0:. The following procedure allows you to use a full-screen editor:

1. Perform a conversational boot of the workstation.
2. At the SYSBOOT> prompt:

```
SYSBOOT> SET/STARTUP OPA0:
SYSBOOT> SET STARTUP_P1 "MIN"
SYSBOOT> CONTINUE
```

The system boots with the console as the command procedure.

3. When you see the \$ prompt, do the following:  
\$ SPAWN  
\$ SET DEFAULT [SYS0.SYSCOMMON.SYSEXEC] ! Set default to the  
directory  
! containing startup.com  
\$ @STARTUP ! This will boot the system minimum  
ALPHA VAX  
\$ MCR SYSMAN \$ MCR SYSGEN  
SYSMAN>IO AUTO SYSGEN> AUTO ALL  
SYSMAN>EXIT SYSGEN> EXIT  
\$ DEFINE/SYSTEM DECV\$IGNORE\_DECNET "TRUE"  
\$ @SYSS\$MANAGER:DECV\$STARTUP
4. When DECV\$STARTUP.COM executes, log out of the workstation terminal so that DECwindows can start up on it.
5. When DECwindows starts on the workstation terminal, log in to the DECwindows session.
6. Use the full-screen editor to make modifications to the SYSTARTUP\_VMS.COM command procedure.

7. Set startup back to the SYS\$SYSTEM:STARTUP.COM.  
SYSGEN> USE CURRENT  
SYSGEN> SET/STARTUP SYS\$SYSTEM:STARTUP.COM  
SYSGEN> SET STARTUP\_P1 ""  
SYSGEN> WRITE CURRENT  
SYSGEN> EXIT
8. Halt the system and reboot.

**Note**

Calls to Multivendor Customer Services indicate that using a line editor is a problem for many people. This is true even for people who remember when line editors were common. For people who started in the field recently, a line editor is a bizarre experience. You might want to discuss a few line editors commands here.

The system manager must prepare the system account to use the full-screen editor before the emergency situation occurs. This cannot be done during a minimum boot. Installing this procedure prepares the system for troubleshooting. It is an example of minimizing the impact of a problem by being prepared to handle the problem.

---

## Using EDT from the Console

When the system is booted with the console as the startup command procedure, you can use the EDT editor in line mode only.

Using an editor in line mode is difficult when you are accustomed to using a full-screen editor. Keep a sheet of line editing instruction handy for emergency situations or for using a hardcopy terminal.

To avoid having to use the line editor, create a DCL command procedure in every system account.

```
$ CREATE SYS$LOGIN:EDTINI.EDT
```

```
SET MODE CHANGE
```

```
[Ctrl/Z]
```

Add the following commands to the system LOGIN.COM file to define a symbol to use the EDTINI.EDT file:

```
$ SET TERM/DEV=VT100
```

```
$ EDIT :== EDIT/EDT/COMMAND=SYS$LOGIN:EDTINI.EDT
```



### Note

You can avoid this by having a SYSTARTUP\_VMS.WORKED!

---

## Handling a Missing System Account Password

If you do not have access to any privileged account on the system and do not have the system account password, you cannot log in to a privileged account to change the system account password.

1. Do a conversational boot and define the console as the startup procedure.
2. At the \$ prompt, do the following to install the images needed to modify the user authorization file:
 

```
$ SET NOON
$ SET NOVERIFY
$ INSTALL := $INSTALL/COMMAND_MODE
$ INSTALL
INSTALL> ADD/OPEN/HEADER_RESIDENT/SHARED  SYS$SHARE:SECURESHR.EXE
INSTALL> ADD/OPEN/HEADER_RESIDENT/PROTECTED/SHARED
          SYS$SHARE:SECURESHRP.EXE
INSTALL> ADD/OPEN/HEADER_RESIDENT/SHARED  SYS$SHARE:LIBRTL.EXE
INSTALL> EXIT
```
3. Use AUTHORIZE to modify the UAF.
 

```
$ SET DEFAULT SYS$SYSTEM  ! Set default to wherever the SYSUAF
                          ! is found
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> MODIFY SYSTEM/PASSWORD= new-password
UAF> EXIT
```
4. Restore the original startup procedure.
 

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET/STARTUP SYS$SYSTEM:STARTUP.COM
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```
5. Run the startup command procedure.
 

```
$ @SYS$SYSTEM:STARTUP.COM
```

You will see error messages regarding the images already installed.



---

**Note**

Discuss this problem resolution in detail to give the students a basic understanding of what you are doing when you define the console as the startup procedure. The purpose is not to just teach people to handle the forgotten system account password task, but to teach what must be done to solve problems using this technique. This is reinforced by a laboratory exercise.

Known file lists last only while the system is operating. If the system is shut down or fails for any reason, you must reinstall all known images after the system is rebooted. The site-independent startup command procedure includes a series of INSTALL commands that install certain system programs as known images. But remember we have not run the site-independent startup command procedure. The install commands here install the system files necessary to run AUTHORIZE. Students should be familiar with the INSTALL utility prior to this class.

Next we run AUTHORIZE to change the system account password. Students are expected to know how to run AUTHORIZE.

Next use SYSGEN to set the startup command procedure back to the original file.

Run the startup.com file to complete the system boot.

After discussing this problem, ask the students if they think they will remember these steps a year from now. The answer should be **NO**. Suggest that this is an appropriate procedure to document in a format they like and store as a handy reference. This is an example of minimizing the impact of problems by being prepared to handle them when they occur.

This procedure represents a security risk. Point out that this demonstrates the need for physically restricting access to the data center and to the console device.

When the system is booted to use OPA0:, it is very sensitive and terminates after more than one error. To prevent premature termination, enter SET NOON or SPAWN/NOWAIT.

---

## Using the Full Backup Utility at OPA0:

To use the full OpenVMS Backup utility when the system is initialized with the console as the startup command procedure, the images must be installed.

1. Perform a conversational boot:

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

```
SYSBOOT> CONTINUE
```

2. Configure the devices:

```
$ SET NOVERIFY
```

```
$ SET NOON
```

```
$ RUN SYSGEN
```

```
SYSGEN> AUTOCONFIGURE ALL
```

```
SYSGEN> EXIT
```

3. Install the images needed to run the Backup utility:

```
$ INSTALL :==$ INSTALL/COMMAND_MODE
```

```
$ INSTALL ADD SYS$SYSDEVICE:[SYS$SYSTEM]MOUNTSHR/OPEN/HEAD/PROT/SHARE
```

```
$ INSTALL ADD SYS$SYSDEVICE:[SYS$SYSTEM]DISMNTSHR/OPEN/HEAD/PROT/SHARE
```

```
$ INSTALL ADD SYS$SYSDEVICE:[SYS$SYSTEM]SECURESHRP/OPEN/HEAD/PROT/SHARE
```

4. Use the full Backup utility.

The primary advantage of using the full Backup utility is that you can selectively restore a single file or group of files instead of restoring the entire save set as must be done with standalone Backup.

### **Hint:**

If you do not remember which images to install, just issue the command to run the Backup utility. The error messages will list the images that could not be found.

## Using an Alternative Site-Independent Startup File

If the system does not boot because of problems with the site-independent startup file, use the SET/STARTUP command to boot the system with an alternative file.

Format:

```
SYSBOOT> SET/STARTUP file-spec
```

Example:

```
SYSBOOT> SET/STARTUP SYS$SYSTEM:XSTARTUP.COM
```

The file-spec must be the file specification of a startup command procedure on the system disk.



---

### Note

Compaq recommends that the system manager not modify the site-independent startup command procedure. The initial site-independent startup command procedure (as named in the software distribution kit) is SYS\$SYSTEM:STARTUP.COM.

However, if the system manager suspects that someone has modified the site-independent startup procedure and a previous version exists, this procedure can be used to boot the system.

How can the system and network manager determine the name of an alternate STARTUP.COM? One way is to examine the log files from the last backup of the system disk. If only the tape is available, the save set could be listed on another system. Another way is to determine if there is a standard for renaming files in the change management procedure. Change management is discussed at the end of this Module.

By default, there is no alternate site-independent STARTUP.COM.

---

## Displaying the Site-Independent Startup Command Procedure

To display the name of the current site-independent startup command procedure, use the SYSGEN command SHOW/STARTUP.

Format:

```
SYSBOOT> SHOW/STARTUP
```

Example:

```
SYSBOOT> SHOW/STARTUP
```

```
Startup command file = SYS$SYSTEM:STARTUP.COM
```

## Using Standalone Backup to Save and Restore the System Disk

### Standalone Backup

Standalone Backup is a subset of the Backup utility that is bootstrapped into main memory without the OpenVMS operating system.

You can use standalone Backup to resolve problems with system initialization caused by missing or corrupted system files.



---

**Note**

System managers should know how to do the following for every system they manage prior to taking this class:

- Boot standalone Backup from console media and from the system disk.
- Build a standalone Backup kit on the system disk.

This section discusses the following tasks:

- Making a backup copy of the system disk
- Restoring the system disk from tape
- Booting the system using an alternate system disk

If the system manager is responsible for several different systems, remembering the special instructions for each of the above tasks may be difficult. Recommend that for each system the system manager have a cookbook-style document that describes how to perform the above tasks on each system. When a system is down and cannot be booted, you do not want to be standing around trying to read the OpenVMS documentation set. To demonstrate the importance of such documentation, hand out a document for the systems that will be used for the laboratory exercises.

Use this as an example of preventing human errors by not expecting a person to remember a procedure that is used infrequently.

---

### Booting Standalone Backup

Use the menu system provided on the OpenVMS Alpha or VAX operating system distribution CD.

1. Run SHUTDOWN and halt the system.
2. Boot the system:
  - On OpenVMS Alpha systems, boot the distribution CD.
  - On OpenVMS VAX systems, boot the distribution CD from the SYS1 directory.
3. When standalone Backup finishes booting, it displays an identification message followed by the dollar sign prompt (\$).

## Backing Up the System Disk to Tape

To back up the system disk to tape:

1. Boot standalone Backup.
2. Enter the BACKUP command in the following format:

```
$ BACKUP/IMAGE/VERIFY input-specifier: output-  
specifier/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 and the name of the save set.
- *label* is the volume label of the tape or tape cartridge.

Example:

```
$ BACKUP/IMAGE/VERIFY DUA0: MUA0:SYSDISK.BCK/SAVE_SET
```



---

### Note

During the laboratory exercises and in the next section, we discuss using standalone Backup to restore the system disk when the system does not boot because a required file has been renamed or deleted.

Suppose, for example, that a file needed to initialize the system has been accidentally renamed to another directory. There is no way to boot the system to fix the problem. Also, the most recent backup of the system disk may have been made some time ago.

The system and network manager should proceed as follows:

- Boot standalone Backup.
- Save the current system disk.
- Boot standalone Backup.
- Restore the previous version of the system disk (before the system file was deleted).
- Boot the system.
- Use the Backup utility to restore system files that have been updated since the restored saveset was made.

See the section on Resolving System Boot Problems.

---



---

### Note

---

## Restoring the System Disk from Tape

To restore the system disk from tape, use the following procedure:

- Boot standalone Backup.
- Place the tape containing the backup copy of the system disk in the tape drive.
- Enter the BACKUP command in the following format:  
\$ BACKUP/IMAGE/VERIFY input-specifier/REWIND output-specifier  
where:
  - *input-specifier* is the name of the tape drive and the save set.
  - *output-specifier* is the name of the system disk.

Example:

```
$ BACKUP/VERIFY MUA0:SYSDISK.BCK DUA0:
```

## Handling System Disk Problems

### Handling System Disk Hardware Problems

If the problem is determined to be a system disk hardware problem, you can still initialize the system using the following steps:

1. Restore a backup copy of the system disk to a spare disk.
2. Boot the system using an alternate system disk.



---

**Note**

This is a good argument for having a spare disk. If you do not have a spare disk, boot standalone Backup. Make a backup copy of a user disk that contains less important files and then restore the system files to the user disk.

Each site should have a written procedure for handling system disk hardware problems. This procedure should include the identification of the user disk to be used in such an emergency.

You might want to point out that in cost justifying a spare disk, the cost of having to take a user disk off-line should be considered.

---

### Software Problems

If the initialization problem is caused by the accidental deletion or corruption of a system file, you may have to restore the entire system disk:

1. Use standalone Backup to make a copy of the current system disk on tape or on a spare disk.
2. Use standalone Backup to restore the previous backup of the system disk.
3. After the system is up and running, you can restore any system file that was changed from the system disk save set you made in step 1.

### Alternative Method for Handling System Disk Software Problems

If you have another system running OpenVMS, you can use the full Backup utility to handle system disk software problems.

1. Connect the problem system disk to a running system.
2. Mount the problem disk and use DCL commands to search the disk for missing files.
3. After you fix the problem, connect the system disk to the original system.

The advantage to this method is that you can use the full Backup utility to restore selected files.



---

**Note**

---

## Resolving System Booting Problems

### Overview

A system that does not boot is a serious problem. You must be able to isolate the source of the problem. If you determine the problem to be a hardware problem, you should notify Multivendor Customer Services.

You can use the methods previously discussed in this chapter to resolve software problems.

This section discusses isolating and resolving system initialization problems.

## Isolating and Resolving System Initialization Problems

### System Initialization Landmarks

Look for the following landmarks when the system is booting:

1. Hardware diagnostics
2. Boot prompt >>> (unless autostart)
3. OpenVMS banner
4. Cluster messages (if applicable)
5. Startup banner
6. SYSTARTUP\_VMS displays

### When a System Does Not Initialize

If a system does not initialize, take the following actions:

1. Check the console display.
  - a. If nothing appears on the console display, check the following:
    - ◆ Does the system have power?
    - ◆ Is the system disk on-line without fault lights?
  - b. If you see hardware initialization displays, but do not see the console >>> prompt:
    - ◆ Check the boot switches (LOCAL, REMOTE, ENABLE, DISABLE, and so on).
    - ◆ Try halting the console according to hardware specific instructions.
  - c. If you see the console >>> prompt, proceed to step 2.
  - d. If you do not see the console prompt, place a service call.
2. Check system manager and operator hardcopy site logs to see if any changes have been made to the software since the last reboot.
3. Reboot the system using a conversational boot.
  - a. Capture the error messages (if any) for review using a hardcopy terminal.
  - b. Read the error messages. Hardware errors are preceded by a question mark (?).
  - c. If you suspect hardware problems, call Multivendor Customer Services.



---

#### Note

How do you do 3a. on a xxxstation? See the hardware manual.

Set console serial ! on some

Connect terminal with printer to serial port 1.

---

4. If you see the SYSBOOT> prompt, go to step 6.
5. If you do not see the SYSBOOT> prompt, the problem may reside in the system files. Restore the system disk from a backup copy and go to step 3.
6. At the SYSBOOT> prompt:
  - Use the SHOW/SCS command to display system parameter settings.
  - Look for obvious problems with system parameters.
  - Set the system to boot OPA0: and enter CONTINUE.
7. At the DCL prompt:
  - Check the dates on the system startup files.
  - Remember to use specific directory file specifications.
  - If a file has been modified, run differences against the previous version.
  - Make sure all the files are present.
  - Turn on system verification so that each DCL command entered is displayed.  
\$ SET VERIFY
  - Execute the site-specific startup command procedure.
  - Watch the program execute, looking for problems.



---

**Note**

At the SYSBOOT> prompt, issue the command SHOW/SCS to display the parameters most likely to cause a boot problem.

On newer systems, you can capture the output of the console to a file by running a serial line (with RJ connectors on each end) from the alternate console port of the booting system to a spare serial port on a workstation. Typically, there is a small switch on the physical console to select between the local display and the alternate console device. On an AXP 3000, for example, the switch is located on the back panel.

---

## Using Standalone Backup in an Emergency

### Case Study: A Late Night Call

Late in the evening on a rainy night, the phone rings at the Smith home. John Smith is one of several system and network managers for a number of systems. The caller is the night shift operator at the plant. The plant just had a power outage and when power was restored, one system would not boot.

John lives some distance from the plant and would like to avoid a long trip over slick roads. Here are the questions he asks the operator and the operator's answers.

| John's Questions                                                       | Operator's Answers                                             |
|------------------------------------------------------------------------|----------------------------------------------------------------|
| What system is giving the problems?                                    | AXPA                                                           |
| Did you see anything on the console when you tried to boot the system? | Yes. I saw the numbers going across the screen.                |
| Did you get the boot prompt?                                           | Yes                                                            |
| Did you see the OpenVMS banner message?                                | No                                                             |
| Well let's try a conversational boot. Do you know how to do that?      | Yes... O.K... now booting conversationally. Nope failed again. |
| Did you get the SYSBOOT> prompt?                                       | No                                                             |

While the operator was trying to boot the system, John recalled a conversation he had with a new system manager regarding a system disk being short on space.

John now knows that he has to restore the system disk from a backup copy. However, the latest system disk backup is three weeks old.

1. What is the probable reason this system will not initialize?
2. How can John restore this system without losing any changes to system disk files that may have been made in the past three weeks?
3. Make a list of recommendations to avoid similar problems in the future.



---

#### Note

This case study is meant to be handled as a classroom discussion. No systems are needed.

It is recommended that you perform the laboratory exercises for Chapter 2 before this case study so the students will be familiar with the system initialization steps.

Suggested discussion questions:

- **How does John eliminate hardware as the cause of the problem?** He questions the operator regarding what he sees on the console display.
  - **What kind of display do you see as the system passes hardware diagnostics?** A series of numbers all on one line of the screen. The operator describes these numbers.
  - **What other landmarks are available from this conversations?** The most important is that the SYSBOOT> prompt is not displayed during a conversational boot.
-

**Case Study Solution: A Late Night Call**

1. What is the probable reason this system will not initialize?

It is likely that the new system manager accidentally deleted a file needed for system initialization when he was trying to free up some disk space on the system disk.

2. How can John restore this system without losing any changes to system disk files that may have been made in the past three weeks?

Follow these steps to correct the situation:

- a. Boot standalone Backup according to the processor-specific instructions.
  - b. Make a backup copy of the system disk to tape or to a spare disk.
  - c. Restore a previous backup copy of the system disk from tape.
  - d. Boot the system.
  - e. By comparing the files on the system disk to the files in the save set, you determine that SYSBOOT.EXE was deleted.
  - f. Compare the dates of the files on the system disk with the dates on the latest backup save set to determine if any files have been updated since the original backup was made.
  - g. Restore these files using the full Backup utility.
3. Make a list of recommendations to avoid similar problems in the future.

There is basically no way to prevent a privileged user from deleting a required system file. However, some recommendations include:

    - a. Review the list of all persons having access to the system account. Should any user be removed? Have all these people received sufficient training?
    - b. Conduct a training session for all persons having access to the system account.
    - c. Implement a change management policy that requires any such changes to be documented.
    - d. Eliminate unnecessary privileged accounts.



---

**Note**

# Avoiding System Initialization Problems

## Overview

You can avoid almost all the problems mentioned in this module if changes to the system files are handled in a controlled environment. You can improve the dependability of your system by designing and implementing change management procedures.

This section discusses change management procedures.



### Note

When should this material be presented?

The developer recommends that this material be presented right after you finish discussing the laboratory exercises and the case study for Chapter 2. At that time the students should be very motivated to accept such suggestions.

This material should be presented before the Chapter 3 laboratory exercises are started.

Because this material is **very important**, you might want to review it as the final section in the class.

All of the problems presented in this chapter could have been avoided by using a proper change management strategy.

---

## Change Management Procedures

### Questions to Ask Before You Schedule Changes

If the answers to the following questions are not satisfactory, do not schedule changes.

- Is the change necessary?
- Has the change been properly researched?
- What is the impact of the change?
- Is this the right time to make the change?
- If something unexpected happens, is there time to reverse the change?
- Are personnel available to help if something goes wrong?
- What procedures are available to return the system to its previous state?



---

#### Note

Emphasize that procedures and methods can vary from site to site, but all sites should use a standard procedure to control the change process.

Suggest that when a serious mistake is made, the first question should be to find out if the procedures were followed. If they were followed, then perhaps some changes to the procedures are required.

All changes should be recorded as they are made. If the change goes well, record that fact as well.

Some procedures that should be established are:

- A standard naming procedure, for example SYSTARTUP\_VMS.OLD
- A method for documenting changes to MODPARAMS.DAT, for example:

```
parameter-name = new-value ! old-value date initials
```

---

## Designing a Standard Procedure to Control Changes

The format of the change management procedure can vary from site to site, but should include the following rules:

- Always make a backup copy of all files *before* you make changes.
- When modifying files, save the previous version of the file under another name so you can use it if something goes wrong. Use a standard naming procedure.
- Document and date all file changes in comments included in the file.
- Record changes in the system manager log files and/or the operator shift reports. Proper notification of changes is required.
- Use the “buddy system.” Have someone check your work.
- Test changes to make sure they were made correctly and perform as expected.

## Training and Enforcement

To make sure that the change management procedure rules are followed:

- Document the change management procedures.
- Provide training for all personnel on the change management procedures.
- Check to see that these procedures are followed by all personnel all the time.

## Summary

### System Initialization Concepts

The OpenVMS operating system is initialized by a series of programs.

### Emergency Initialization Procedures

- When a system does not boot, you should reboot the system conversationally.
- Booting the system so that the startup procedure takes commands from the console is one way to research system initialization problems.
- Booting the system minimally runs all programs except the site-specific startup command procedure.
- To make a copy of the system disk, use standalone Backup.
- When the system files are deleted or corrupted, use standalone Backup to restore the system disk from the backup copy.

### Resolving System Booting Problems

- The first step in resolving a system booting problem is to eliminate hardware problems.
- Reboot the system using a conversational boot to isolate the problem.
- To resolve problems caused by missing or corrupted system files, restore the backup copy of the system disk using standalone Backup.
- You can resolve problems with the startup procedures at the DCL level.

### Avoiding System Initialization Problems

A change management procedure to control changes to the system software prevents errors that cause system downtime.

## Chapter Terms

**APB.EXE**

Primary bootstrap program on an OpenVMS Alpha system.

**Conversational boot**

Boot procedure that pauses after loading SYSBOOT.EXE, allowing changes to be made to system parameters.

**Standalone Backup**

A subset of the Backup utility used to save and restore files on a system that is running a reduced operating system.

**SYSBOOT.EXE**

Secondary bootstrap program on both OpenVMS Alpha and OpenVMS VAX systems.

**VMB.EXE**

Primary bootstrap program on an OpenVMS VAX system.



---

# ***Troubleshooting Printing Problems***

## ***Module 3***

## Introduction

Normal users put a heavy demand on computer systems for hardcopy output. Most systems support various types of printers and use a wide variety of formats and print types. Maintaining the printing environment is an important and time-consuming task of the system and network manager.

A print job is handled by several subsystems on its path from initiation to final completion of the task. Understanding the concepts of the queue management system, LAT queues, and printer formatting options enables you to handle and prevent printer problems.

## Objectives

To manage printing on OpenVMS systems, you should be able to:

- Describe the concepts of the OpenVMS print queue system.
- Describe the concepts of remote print queues.
- Set up a remote print queue.
- Apply the troubleshooting methodology to print queue problems.
- Describe how OpenVMS formats a print job.

## Resources

- *OpenVMS System Manager's Manual*
- *OpenVMS System Management Utilities Reference Manual*



### Note

Time

Materials: 3-3 ½ hours

Practice: 3 hours

---

## OpenVMS Batch and Print Queue System Concepts

### Overview

When a user submits a print job on an OpenVMS system, the job is handled by the OpenVMS batch and print queue system and the print symbiont. The device setting of the printer also influences the print job.

The system and network manager is responsible for setting up and maintaining the print queues and the system printers.

This section discusses:

- OpenVMS batch and print queue system
- Queue manager process
- Job controller process
- Queue database
- Obtaining information on queue system processes
- Print symbiont
- Customizing print queues
- Queue access

## OpenVMS Batch and Print Queuing System

### Print Queuing Process

The print queuing process follows these steps:

1. A user enters the DCL PRINT command.
2. The following information is sent to the OpenVMS software that manages batch and print queues:
  - Type of job
  - File name or names
  - Queue name
  - Special options
3. The information is stored in a queue database.
4. The print job is placed in the appropriate **queue** to await its turn for processing and the user is notified.
5. Print jobs are sent to an independent process, called a *symbiont* for formatting.
6. The formatted print files are sent to the printer.



---

#### Note

First we discuss printing without the complication of terminal servers or remote print queues. This is the OpenVMS part of the print process.

Print jobs are also initiated by system calls made by application programs, but for purposes of this discussion have the user issue the DCL command PRINT.

Note that, while it is possible to have multiple queue managers in a cluster since OpenVMS Version 6.0, the discussion in this chapter focuses on a single queue manager for simplicity.

Review the definition of the term *queue* as given in the chapter terms. A queue is basically a list of jobs waiting to be processed. Jobs are selected for printing based on:

- Priority
- Size
- Time of submission to the queue

The queuing system:

- Allows multiple jobs to get in line to use the same printer.
- Controls the options that are applied to the print job by associating the job with a print form.

Allows the system manager to control who can print to a certain printer.

---

## Software Components

The following are the software components of the OpenVMS batch and print queuing system:

- Queue manager processes running on one or more nodes in the cluster
- One or more queue databases
- Job controller process on each node of the cluster
- Print symbiont



### Note

The print symbionts are not technically part of the OpenVMS batch and print queue system. This is a list for discussion.

For the next section and figure:

The batch and print queuing system was changed dramatically in OpenVMS Version 5.5. Many students took their system management training before these changes were implemented, so treat this section as an update. Also be sensitive to the students whose systems have not been upgraded and be ready to discuss the way the queue manager worked prior to OpenVMS Version 5.5.

The big change is the queue database as a component of the queue management system. The presence of the database means that the queue and form definitions do not need to be included in the system startup files. As you go through the materials, it is very important to point out which commands must be issued each time the system is rebooted and which are stored in the database.

Since OpenVMS Version 6.0, multiple queue managers may run on a single cluster. In a way this is a return to the old method, where a queue manager ran on each node in the cluster, since you could conceivably have a single queue manager on each cluster node.

In a mixed-architecture cluster, the system manager will probably set up at least two queue managers and their associated queues, one for OpenVMS VAX systems and one for Alpha systems.

If the commands to initialize the queues are left in the startup file, the queue manager compares the queue definitions at startup time with the definitions in the queue master file. If there is no change, the only penalty is wasted CPU time. If the definition in the startup file is different, the queue master file is altered. This could cause problems. This action is an example of backward compatibility built into OpenVMS.

Notice that there is a discussion of the IPC coming up.

Points to make from this figure:

- In a cluster there are one or more queue manager processes.
- Each node in the cluster has a job controller process.
- The disk on which the queue databases are located must be available to the entire cluster.



---

**Note**

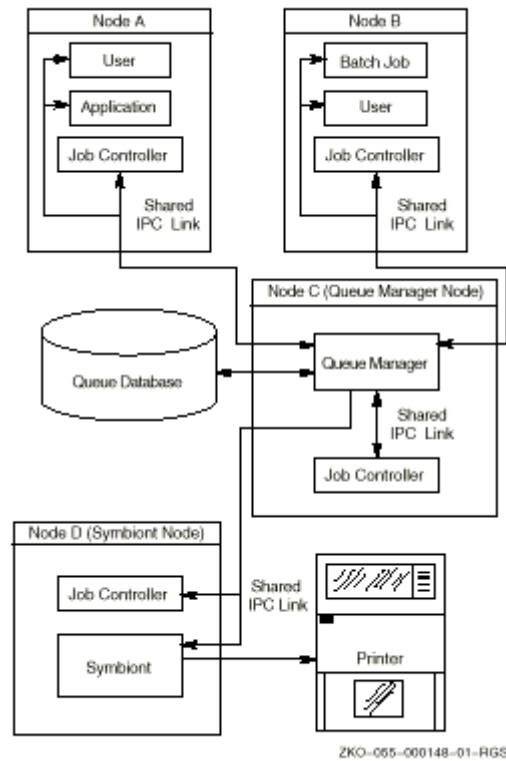
User processes, symbionts, and job controllers on each node communicate directly with the centralized queue manager through a shared interprocess communication (IPC) link.

IPC is an internal OpenVMS communication mechanism.

---

## Figure: OpenVMS Batch and Print Queuing System

The following figure shows the OpenVMS batch and print queuing system.



## Queue Manager Process

### Queue Manager Process in an OpenVMS Cluster System

An OpenVMS Cluster system has one or more queue manager processes. The system and network manager can designate a single node to run queue manager processes, or run several queue managers across several cluster nodes. In this section, the discussion focuses on a cluster with a single queue manager.

User processes, symbionts, and job controllers on each node communicate directly with the central queue manager through a shared *interprocess communication (IPC) link*.

Jobs submitted from one node can execute on a queue running on another node in the cluster.



---

**Note**

In OpenVMS Version 5.5, the queue manager process handled all queuing requests on a standalone node or in a cluster. Since OpenVMS Version 6.0, a cluster can have more than one queue manager. In the discussions that follow, we talk about a single queue manager, because each queue manager is separate from all others in a cluster.

A queue manager process accesses its associated queue database processes on a node or in a cluster that sends a job to a queue. User processes, symbionts, and job controllers on each node communicate directly with the queue manager through a shared *interprocess communication (IPC) link*. IPC is an internal OpenVMS communication mechanism and is used by DECdtm (not documented). Jobs can be submitted from one node and executed on a queue running on another node in the cluster.

Before you can enter any queue command, a queue manager process must be running.

As of Version 5.5, the START/QUEUE/MANAGER command does not need to be included in the startup command procedure, SYSTARTUP\_V5.COM (SYSTARTUP\_VMS.COM for OpenVMS Version 6.0 and later). However, leaving the command in the startup command procedure does not cause a failure during startup. Normally, a job controller handling such a command finds a queue manager already active and does not attempt to start another queue manager. If the queue manager is not running on the most preferred node, however, it is moved. This uses CPU cycles and makes system startup take longer.

By default, in an OpenVMS Cluster environment, the queue manager attempts to fail over to another node if the node on which the queue manager is running leaves the cluster. You can specify the order in which nodes in the cluster should claim the queue manager process or limit the nodes that can run the queue manager.

---

## Starting the Queue Manager and Creating the Queue Database

Enter the following command to start the queue manager and create the queue database in SYS\$COMMON:[SYSEXE] on an OpenVMS Cluster system,:

```
$ START/QUEUE/MANAGER/NEW_VERSION/ON=(node-list )
```

where node-list is a list of the nodes in the cluster.

The queue manager starts on the first node in the list. If the node that is running the queue manager fails, the process fails over to the next node in the list.

After you enter the command, the command is stored in the queue database. Thereafter the job controller process starts the queue manager during reboot unless a STOP/QUEUE/MANAGER/CLUSTER command is entered.



### Note

We have not yet discussed the queue database, but the queue database is created when the queue manager is started for the first time.

This procedure must be modified if the queue database is to be located on another disk other than the one indicated by the default. See the *OpenVMS System Manager's Manual*.

On a single node there is no option to have the queue manager process fail over.

What is really important here is that this command needs to be issued only once to define the queue database and the queue manager process. It is **not** issued each time the system is rebooted.

---

## Queue Manager Failover

If the node on which the queue manager process is running leaves the cluster, a system job controller starts the queue manager on another node, based on the priority specified by the system and network manager.

## Job Controller Process

In an OpenVMS Cluster system, there is one job controller process on each node.

- The job controller process is started by the startup command procedure.
- The job controller process name is JOB\_CONTROL.
- If the job controller process stops, restart it as follows:

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



---

### Note

The role of the job controller in the queue management process is changed as of OpenVMS Version 5.5. A good reference is the OpenVMS Version 5.5 Technical Update Seminar. Note that OpenVMS Version 6.0 and later versions allow multiple Version 5.5 style queue managers.

---

## Job Controller Queue Management Functions

Job controller queue management functions are to:

- Create and monitor batch, symbiont, and queue manager processes.
- Start the queue manager.
- Handle the failover of the queue manager to another OpenVMS Cluster node when the node on which the queue manager is running leaves the cluster.



---

### Note

This block answers the question “What is the role of the job controller process in the queue management business?”

---

## Events That Cause the Job Controller to Perform Queue Functions

The job controller logic that performs these functions runs when one of the following events occurs:

- The job controller process is started.
- The queue manager process terminates.
- A START/QUEUE/MANAGER command is entered.
- The preferred list of nodes on which the queue manager is to run is changed.



---

### Note

This block answers the question “When does the job controller get into the queue management business?”

When a system is initialized and the job controller is created, the job controller starts the queue manager process if:

- No queue manager process exists on any node in the cluster.
- A queue manager process was running when the system shut down.
- A new queue manager is not started automatically if:
  - The queue manager was stopped with a STOP/QUEUE/MANAGER/CLUSTER command.
  - The queue manager was stopped twice within 2 minutes with the STOP/ID command.
  - The queue manager failed because of problems caused by a trusted symbiont (usually third party). Error messages should indicate that such problems occurred. This can usually be fixed by stopping the symbiont.

When the job controller is involved in queue management, here are the steps it follows:

- Reads the queue database to identify how the queue manager process is set up.
  - Determines if a queue manager process needs to be started on the cluster. If one is already running the answer is no.
  - Asks “Am I the best job controller to start the queue manager process?”
-

## Queue Database

### Purpose

The queue database stores information used to keep the queuing system operating. This includes information about:

- Jobs
- Queues
- Queue manager

The queue database is stored on disk and is nonvolatile.



---

**Note**

The queue database is nonvolatile storage for queue information. Commands to initialize queues do not need to be executed each time the system is booted.

---

### Files

The queue database is made up of the following files:

- QMAN\$MASTER.DAT — the master file
- SYS\$QUEUE\_MANAGER.QMAN\$QUEUES — the queue file
- SYS\$QUEUE\_MANAGER.QMAN\$JOURNAL — the journal file



---

**Note**

The default location of the three files that make up the queue database is the directory SYS\$COMMON:[SYSEXE] on the system disk of the node that first creates the database.

---

### Queue Master File

The master file is shared by the job controller and queue manager.

The master file contains:

- Pointers to the queue and journal files
- Definitions of forms and characteristics
- List of queue names
- List of nodes allowed to run the queue manager



---

**Note**

The single master file is shared by all processes in the cluster. Because the information in the master file is not accessed often, sharing of the master file is not a queuing system performance factor.

---

## Queue File

The queue file contains the queue definitions you specify when you create, start, or modify queues using the commands:

- INITIALIZE/QUEUE
- START/QUEUE
- SET QUEUE

## Queue Journal File

The journal file contains the following information about jobs in the queues:

- Job definitions
- Updates to the queue and master files
- Information that allows the queue manager to return to the last known state if a system running the queue manager unexpectedly stops or leaves the OpenVMS Cluster system

The queue manager allocates space for the journal file.

- Each job submitted to a queue makes an entry in the journal file.
- When the journal file is filled, the queue manager allocates space for a new journal file.
- Newly created files are entered in the new journal file.
- Entries related to active jobs are transferred from the old journal to the new file.
- When the transfer is complete, the old journal file is deleted.



### Note

The new journal file is not necessarily the same size as the old journal file. An algorithm is used to allocate space which takes several factors into consideration including the past week's history.

If disk space becomes a problem, the queue manager sends warning messages to the OPCOM file. If this problem is not resolved, the queue manager fails.

If the disk on which the journal file is located must be restored, there is no need to restore the journal file because the backup would contain obsolete information anyway. In such a case, jobs submitted to queues would be lost.

---

## In-Memory Queue Manager Database

The queue manager maintains a database in memory that contains all the information needed for job processing.

The database in memory is built by:

- Reading all data from the queue file into memory
- Reading the required data from the journal file
- Verifying outstanding queuing system communications and job requests and resubmitting them as needed
- Verifying that all objects such as processes and queues that are referenced in the queue database are valid

Information required to process jobs and information requests is stored in the in-memory database and copied to the journal file to prevent loss of jobs and messages during a queue manager failover.



---

### Note

Students need to understand that the new queue system shows improved performance. You might think this is not true because the queue database is now maintained on a disk. The in-memory database is needed to improve the performance of the queue system.

---

## Default Queue Database Location

The three files that make up the queue database are located by default:

- On the system disk of the node that first creates the queue database
- In the directory SYS\$COMMON:[SYSEXE]

## Alternate Location for the Queue Database

When space on the system disk is limited, you can move the queue database to another disk.

- The queue file and the journal file must be located in the same directory.
- In an OpenVMS Cluster environment, the logical name QMAN\$MASTER must be defined identically on all nodes in the cluster.

## Performance Considerations

The performance of the queuing system is dependent on the performance of the disk that contains the queue journal file.

To obtain the best performance, place the journal file on:

- A disk dedicated to the queuing system
- A solid-state disk

If the journal file is located on a shared magnetic disk, use as fast a disk as possible.



### Note

Due to the sequential nature of the journal file write operations, head movement on a disk dedicated to the queuing database is kept to a minimum. Because all disks have about the same rotational speed, the head movement speed of the disk is not important if the disk is dedicated to the journal file.

Disk speed becomes important only when the disk is shared.

This is a good situation in which to use one of the smaller disks. The disk can be dedicated to the queue database without wasting so much disk space.

It was mentioned previously, but you might want to reiterate that the speed of the CPU that runs the queue manager process also has an effect on the performance of the queuing system.

The two performance factors are:

- CPU speed of the node running the queue manager
  - Access to the journal file
-

## Moving Queue and Journal Files

To move the queue file and the journal file, perform the following steps:

- Shut down the queue manager by entering the DCL command:  
`$ STOP/QUEUE/MANAGER/CLUSTER`
- Copy the files to their new location. The queue and journal files must reside in the same directory.
- Remove the original copies of the files.
- Restart the queue manager by entering the following DCL command, where *directory-name* is the directory specification for the directory containing the queue and journal files:  
`$ START/QUEUE/MANAGER]device:[directory-name]`

In an OpenVMS Cluster environment, if the string substituted for *directory-name* in the command is a concealed logical name, it must be identically defined on all nodes in the cluster.

After you restart the queue manager, the directory location is stored in the master file. If you restart the queue manager a second time, you do not have to specify the directory location of the queue and journal files.



### Note

There really is little reason to move the queue master file. It is not accessed frequently and does not occupy very much disk space. However, customers worried about disk space on the system disk may be tempted to move the queue master file.

Moving the queue and journal files to other disks may be important for disk space and performance considerations. Forgetting to use the `device:[directory-name]` will put everything back where it started, except empty.

If you restart the queue manager, do not specify the location of the queue and journal files to avoid making an error.

---

## Moving the Master File

To move the master file from the default location, perform the following steps:

1. Shut down the queue manager by entering the DCL command:  
\$ STOP/QUEUE/MANAGER/CLUSTER
2. Copy the file QMAN\$MASTER.DAT to its new location.
3. Edit the file SYSS\$COMMON:[SYSMGR]SYLOGICALS.COM to add the following line defining the logical name QMAN\$MASTER, where *directory-name* is the complete directory specification for the directory where the file is located:

```
$ DEFINE/SYSTEM/EXECUTIVE_MODE QMAN$MASTER device:directory-name
```

On an OpenVMS Cluster system, QMAN\$MASTER must be defined identically on all nodes in the cluster.

4. Restart the queue manager by entering the DCL command:  
\$ START/QUEUE/MANAGER

## Backup and Restore of the Queue Database

To save a record-by-record copy of the queue database *while the queuing system is functioning*, perform the following steps:

1. Use the Convert utility to save the master file and the queue file with different file names.

```
$ CONVERT/SHARE QMAN$MASTER.DAT QMASTER.KEEP
```

```
$ CONVERT/SHARE SYSS$QUEUE_MANAGER.QMAN$QUEUES QFILES.KEEP
```

2. Use the OpenVMS Backup utility to save the files created with the Convert Utility.

```
$ BACKUP/LOG QMASTER.KEEP, QFILE.KEEP saveset -name
```



### Note

This page is included because it is a good opportunity to demonstrate the importance of properly backing up database files.

The special instructions for saving the queue database while the queuing system is functioning are found in the *OpenVMS System Manager's Manual*. If the queue database is stored on the system disk and the system shuts down to back up the system disk, this is not a problem. Backup of the queue database becomes a problem when the queue database is stored on a disk that will be backed up while the system is running.

The instructor should, at this point, emphasize the importance of performing proper backup of databases. Point out that when the proper procedures are not followed, the problem is not discovered until the disk has to be restored. At that time, it is too late.

---

characteristics and reinitialize each queue if the disk needs to be restored because these data items are stored in the nonvolatile queue database. One way to handle the problem is to run the Convert utility for these two files before running the nightly backup of the disk. This requires special instructions for backing up this disk.

Also at this time, point out that other products have special backup requirements. Whenever a new layered product or application is added to the system, the system manager should check on the backup requirements for the product. This is another example of controls that should be built into the change management system. An example of a product with special backup requirements is ALL-IN-1 because it uses index files that are open whenever ALL-IN-1 is running.

The Convert utility copies records from one or more source data files to a second output data file, which can differ in file organization and format from the first.

You can also use CONVERT to reformat an indexed file in which you have deleted or inserted many records. The file specification of the indexed file is used as both the input and output file specification. In this case, the output file has a version number one greater than the highest existing version of the input file.

For more information see the *OpenVMS Record Management Utilities Reference Manual* or online help.

Another method of ensuring that the queue database is correctly saved when it is located on a disk other than the system disk is to follow a two-step procedure:

1. Use the CONVERT command to make two new copies of the files.
2. Back up the disk without the /IGNORE=INTERLOCK qualifier.

The open queue database files will not be copied by BACKUP. If the disk is ever restored, the new queue files must be renamed.

There is no need to back up the journal file because the information will be obsolete anyway by the time it is restored. Losing the journal file will cause the loss of all jobs in the queues.

---

## Obtaining Information on Queue System Processes

### Displaying Queuing System Processes

Use the SHOW SYSTEM command to look for the four processes used by the queuing subsystem.

```
$ SHOW SYSTEM
```

```
OpenVMS V7.2 on node VRR00M 15-DEC-1998 10:01:33.55 Uptime 30 19:57:02
```

| Pid      | Process Name  | State | Pri | I/O   | CPU         | Page | flts | Pages |
|----------|---------------|-------|-----|-------|-------------|------|------|-------|
| 00000201 | SWAPPER       | HIB   | 16  | 0 0   | 00:00:07.57 | 0    | 0    |       |
| 00000207 | IPCACP        | HIB   | 10  | 7 0   | 00:00:00.08 | 95   | 183  |       |
| 0000020B | JOB_CONTROL   | HIB   | 10  | 649 0 | 00:00:08.61 | 292  | 426  |       |
| 0000020C | QUEUE_MANAGER | HIB   | 8   | 237 0 | 00:00:03.60 | 712  | 1065 |       |

```
.  
.
.
```



#### Note

Use this display to point out the processes that must be running for the queuing system.

### Using the Operator Log File

The job controller and queue manager write error information to the operator terminal and operator's log file.

To extract queuing system information, enter the following command:

```
$ SEARCH SYS$MANAGER:OPERATOR.LOG /WINDOW=n QUEUE_MANAGE, JOB_CONTROL -  
- /OUTPUT=file-name
```



#### Note

Searching the operator log file for messages regarding the queue manager and job controller was discussed in Module 1, Dependable Systems.

## Print Symbiont

### Function of the Print Symbiont

An output execution queue accepts jobs for processing by a symbiont.

- A symbiont is a process that transfers record-oriented data to or from a device.
- As the symbiont processes each file, it produces output for the device it controls, such as a printer or a terminal.
- The symbiont process is designated when the queue is established.



---

**Note**

For example, an output symbiont transfers data from a disk to a printer.

The symbiont is set by the INITIALIZE/QUEUE command qualifier /PROCESSOR.

This should be review.

```
$ INITIALIZE/QUEUE/PROCESSOR=file-name queue-name
```

If the qualifier is omitted, the system uses SYS\$SYSTEM:PRTSMB.EXE with an output execution queue.

Later we will see that when remote print queues are initialized, we will designate the symbiont as LATSYM.EXE.

---

### Standard OpenVMS Print Symbionts

The standard print symbiont provided by the OpenVMS operating system is named PRTSMB.EXE and is designed to print files on hardcopy devices.

Other symbionts include:

- LATSYM.EXE — Prints files on output devices connected to terminal servers
- CPS\$SMB.EXE — Prints files on PostScript printers on the LAT
- LPS\$SMB.EXE — Prints files on LPS printers

### System Resources for the Print Symbiont

The print symbiont is an OpenVMS process, and, like all processes, needs system resources.

The job controller creates the print symbiont as a detached process with the same WSDEFAULT, WSQUOTA, and WSEXTENT as the job controller when a START/QUEUE command is executed.

The job controller gets its quota values from the RUN/DETACH command that creates it.

Qualifiers to the INITIALIZE/QUEUE command can modify the resource quotas for the print symbiont. Some examples of resource quotas that can be modified are:

- BASE\_PRIORITY
- WSDEFAULT
- WSEXTENT
- WSQUOTA

## Displaying the Symbiont Process

Use the SHOW SYSTEM command to display the symbiont processes.

\$ SHOW SYSTEM

OpenVMS V7.2 on node BROWNY 15-DEC-1998 10:09:55.00 Uptime 30 20:05:23

| Pid      | Process Name    | State | Pri | I/O      | CPU         | Page  | flts | Pages |
|----------|-----------------|-------|-----|----------|-------------|-------|------|-------|
| 00000201 | SWAPPER         | HIB   | 16  | 0 0      | 00:00:07.64 | 0     | 0    |       |
| 00000205 | CONFIGURE       | HIB   | 8   | 5 0      | 00:00:00.08 | 114   | 179  |       |
| 00000207 | IPCACP          | HIB   | 10  | 7 0      | 00:00:00.08 | 95    | 183  |       |
| 00000208 | ERRFMT          | HIB   | 8   | 18775 0  | 00:01:11.96 | 151   | 239  |       |
| 00000209 | OPCOM           | HIB   | 9   | 256258 0 | 00:16:13.12 | 912   | 170  |       |
| 0000020A | AUDIT_SERVER    | HIB   | 10  | 267 0    | 00:00:01.43 | 495   | 867  |       |
| 0000020B | JOB_CONTROL     | HIB   | 10  | 649 0    | 00:00:08.61 | 292   | 426  |       |
| 0000020C | QUEUE_MANAGER   | HIB   | 8   | 237 0    | 00:00:03.60 | 712   | 1065 |       |
| 0000020D | SECURITY_SERVER | HIB   | 10  | 361 0    | 00:00:06.54 | 1489  | 1625 |       |
| 00000210 | NETACP          | HIB   | 10  | 122679 1 | 21:39:59.51 | 9008  | 9024 |       |
| 00000211 | EVL             | HIB   | 4   | 328901 0 | 00:08:10.23 | 316   | 593  | N     |
| 00000212 | REMACP          | HIB   | 9   | 201 0    | 00:00:00.40 | 107   | 87   |       |
| 00000213 | LATACP          | HIB   | 14  | 10 0     | 04:56:06.01 | 870   | 841  |       |
| 00000214 | NMAIL           | HIB   | 4   | 6 0      | 00:00:00.16 | 263   | 96   |       |
| 00000215 | INSPECT\$Exec   | HIB   | 8   | 2655 0   | 00:00:25.74 | 10997 | 393  |       |
| 00000218 | SYMBIONT_4      | HIB   | 6   | 184 0    | 00:01:34.90 | 2329  | 216  |       |
| 00000219 | SYMBIONT_5      | HIB   | 6   | 184 0    | 00:01:31.87 | 1848  | 219  |       |
| 0000021A | SYMBIONT_6      | HIB   | 6   | 31 0     | 00:01:30.81 | 1061  | 193  |       |
| 0000021B | SYMBIONT_7      | HIB   | 6   | 31 0     | 00:01:30.35 | 1060  | 193  |       |
| 0000021C | SYMBIONT_8      | HIB   | 6   | 31 0     | 00:01:31.80 | 1059  | 193  |       |
| .        |                 |       |     |          |             |       |      |       |
| .        |                 |       |     |          |             |       |      |       |
| .        |                 |       |     |          |             |       |      |       |



---

**Note**

As with any process, the system and network manager should monitor and adjust the resources for the print symbionts.

Defaults are:

- `BASE_PRIORITY` 4
  - `WSDEFAULT` = `WSQUOTA`/4
  - `WSEXTENT` = Systems `WSMAX`
  - `WSQUOTA` = `WSEXTENT`/4
  - The `RUN/DETACH` command is in  
`SYS$STARTUP:OpenVMS$CONFIG-050-JOBCTL.COM`
-

## Customizing Print Queues

### Queue Options

You can customize print queues by specifying certain options for each queue when it is defined. Options can:

- Set up the queue to print flag pages at the beginning of each job, or trailer pages at the end of each job, or both.
- Associate a print form with a queue to:
  - Set up the printer for certain formats using device control modules.
  - Associate a paper stock with a queue.
- Restrict access to the queue.
- Control the way jobs flow through queues.
- Control the resources available to the print symbiont process.
- Specify a queue as a generic queue.
- Control which jobs execute on a queue.
- Designate an execution queue as an autostart queue, and control the node on which it will run and the nodes to which it will fail over.



---

#### Note

The rest of this section is all review from the *OpenVMS System and Network Node Management*. If the class seems to be rather knowledgeable about this material you should be able to address it very lightly. However laboratory exercises will require that the students know this material in order to solve problems. Refer students to the *OpenVMS System Manager's Manual* for further information.

The student needs to understand how options become associated with:

- The print queue
- The print job

The student needs to understand what happens when conflicting options and characteristics become associated with the job.

Be careful of the word “characteristic;” it has a special meaning.

---

## How to Set Options on a Print Queue

You can customize a queue with qualifiers to the following commands:

- INITIALIZE/QUEUE
- START/QUEUE
- SET QUEUE

Queue definitions are stored in the queue database file. Queues do not have to be initialized each time the system is rebooted

## Displaying Print Queue Options

To resolve printer problems, you must obtain information about the generic queue, the execution queue, and the jobs in the queue.

```
$ SHOW QUEUE/FULL/ALL SYS$PRINT
```

```
Generic printer queue SYS$PRINT
```

```
  /GENERIC=(LPS40$SCDTST) /OWNER=[1,4] /PROTECTION=(S:E,0:D,G:R,W:W)
```

```
$ SHOW QUEUE/FULL/ALL LPS40$SCDTST
```

```
Printer queue LPS40$SCDTST, busy, on NODE01::SCDTST, mounted form
  DCPS$DEFAULT (stock=DEFAULT)
```

```
  /BASE_PRIORITY=4 /DEFAULT=(FORM=DCPS$DEFAULT (stock=DEFAULT))
```

```
  /NOENABLE_GENERIC /LIBRARY=DCPS_LIB /OWNER=[1,4] /PROCESSOR=DCPS$SMB
```

```
  /PROTECTION=(S:M,0:D,G:R,W:S) /SCHEDULE=(NOSIZE)
```

```
  /SEPARATE=(BURST,FLAG,TRAILER)
```

```
Entry Jobname  Username  Blocks Status
```

```
-----
```

```
691 LOGIN      SMITH      4    Printing
```

```
  Submitted 15-DEC-1998 12:18 LP$FORM (stock=DEFAULT)
```

```
  /PRIORITY=100
```

```
  File: _$1$DUA1:[SMITH]LOGIN.COM;25 (printing)
```



### Note



---

**Note**

Note the following:

```
/AUTOSTART_ON=(NODE01::SCDTST,NODE02::SCDTST)
/BASE_PRIORITY=4
/BLOCK_LIMIT=500
/DEFAULT=(FORM=LPS$$FORM (stock=DEFAULT))
/NOENABLE_GENERIC
  /LIBRARY=LPS$DEVCTL Lowercase
/OWNER=[1,4]
/PROCESSOR=LPS$$SMB
/PROTECTION=(S:E,O:D,G:R,W:W)
/SEPARATE=(FLAG,RESET=(LPS$$EOJ),TRAILER)
```

---

## Autostart Queues

The autostart feature simplifies queue startup and ensures high availability of execution queues in a cluster.

The autostart feature allows you to do the following:

- Start all autostart queues on a node at system startup with a single command.
- Specify a list of nodes (within an OpenVMS Cluster system environment) to which a queue can automatically attempt to fail over if a node leaves the cluster.

You can designate execution print queues as autostart queues. You cannot designate generic queues as autostart queues.

## Initializing Autostart Execution Queues

To initialize an autostart execution queue, use the /AUTOSTART\_ON qualifier.

```
$ INITIALIZE/QUEUE/AUTOSTART_ON=(node::device,,,)/START queue-name
```

To notify the queue manager to automatically start all active autostart queues on a node, use the following command in the system startup command procedure:

```
$ ENABLE AUTOSTART
```



### Note

An autostart queue is active if it was running when the cluster went down.

---

## Making an Autostart Queue Inactive

To make an autostart queue inactive, use the STOP/QUEUE command.

- An inactive autostart queue remains inactive until a START/QUEUE command is entered.
- You might want to stop a specific autostart print queue if the printer is being serviced.

## Disabling All Autostart Execution Queues

Use the DISABLE AUTOSTART/QUEUES command to notify the queue manager process to perform the following tasks on a node that is to be stopped:

- Mark all autostart queues as “stop pending” in preparation for a planned shutdown of the system.
- Prevent autostart queues from failing over to the node.
- Upon completion of any job executing on one of the node's autostart queues, force the queue to fail over to the next available node in a queue's failover list (if any) on which autostart is enabled.
- Autostart queues that do not have a failover list of nodes remain activated for autostart. The queue manager process restarts the stopped autostart queues when the ENABLE command is entered.



### Note

The best way to differentiate between STOP and DISABLE is to consider what you want to do.

If you want to take the node down, you need to disable the autostart queue so any job that is executing finishes and then the queue fails over. The printer itself continues to be available.

If however you need to service the printer, you need to use STOP because there is no printer to print to.

When you speak of the queue manager, use the term *queue manager process* to make sure that it is not misunderstood. You might think you were talking about the system manager — the person not the process.

---

## Queue Access

### Restricting Access to a Queue

As with files, you can use UIC-based or ACL-based protection to control access to a queue.

There is a laboratory exercise involving a problem with queue protection.

### Types of Queue Access

The following types of queue access are allowed:

|   | Access | Function                                                                                                          |
|---|--------|-------------------------------------------------------------------------------------------------------------------|
| W | SUBMIT | Submits jobs to the queue AND manages their jobs.                                                                 |
| D | DELETE | Modifies and deletes jobs in a queue.                                                                             |
| R | READ   | Displays attributes of jobs in a queue.                                                                           |
| E | MANAGE | Modifies any nonsecurity attribute of a queue and deletes the queue. Also allows SUBMIT, DELETE, and READ access. |

### Default Queue Protection

The system applies the default queue protection: (S:E,O:D,G:R,W:W)

- System MANAGE
- Owner DELETE
- Group READ
- World SUBMIT

### Displaying Queue Protection

When a user is unable to print a file to a queue, you may want to check the queue protection.

```
$ SHOW SECURITY /CLASS=QUEUE LPS20$CRIER
```

```
LPS20$CRIER object of class QUEUE
```

```
Owner: [SYSTEM]
```

```
Protection: (System: M, Owner: D, Group: R, World: S)
```

## Remote Print Queue Concepts

### Overview

A remote print queue is associated with a printer connected to a terminal server.

A *terminal server* is a communications server that provides an efficient way to connect terminals, modems, printers, and other devices to the Ethernet local area network (LAN).

To troubleshoot problems on remote print queues, the system manager should be familiar with LAT tools and concepts.

This section discusses:

- LAT network concepts
- Terminal server concepts
- LAT management tools

## LAT Network Concepts

### Overview

The primary goal of LAT is to provide users with access to computing resources connected to terminal servers.

Terminal servers use the LAT network *protocol* to communicate on an Ethernet LAN. LAT software and hardware implement the protocol on an OpenVMS system.

A LAT network consists of:

- Service nodes
- Terminal server nodes



#### Note

A protocol is a series of communications rules. A protocol is implemented with hardware and software.

The Local Area Transport (LAT) communications protocol is a protocol that the OpenVMS operating system uses within a local area network to communicate with terminal servers. The LAT software implements the protocol on an OpenVMS system. A LAT network consists of:

- **Service nodes** — Computers on the Ethernet LAN that offer services to users.
- **Terminal server nodes** — Communications servers that connect terminals and other devices to the Ethernet LAN.

LAT software is proprietary to Compaq and bundled with the OpenVMS operating system. (Bundled means the customer gets the product with the operating system license. Layered product licenses are purchased separately.)

### Service Nodes

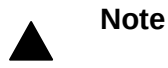
Each OpenVMS node on a LAT network can offer its resources as a *service*.

- Most nodes offer general processing services (interactive login).
- A node can also be set up to offer a special application processing service.

For example, an OpenVMS service node might offer three services:

- General processing
- Data entry application
- Stock quote application

Each service is distinguished by a name assigned by the system manager.

**Note**

---

**Terminal Server Node**

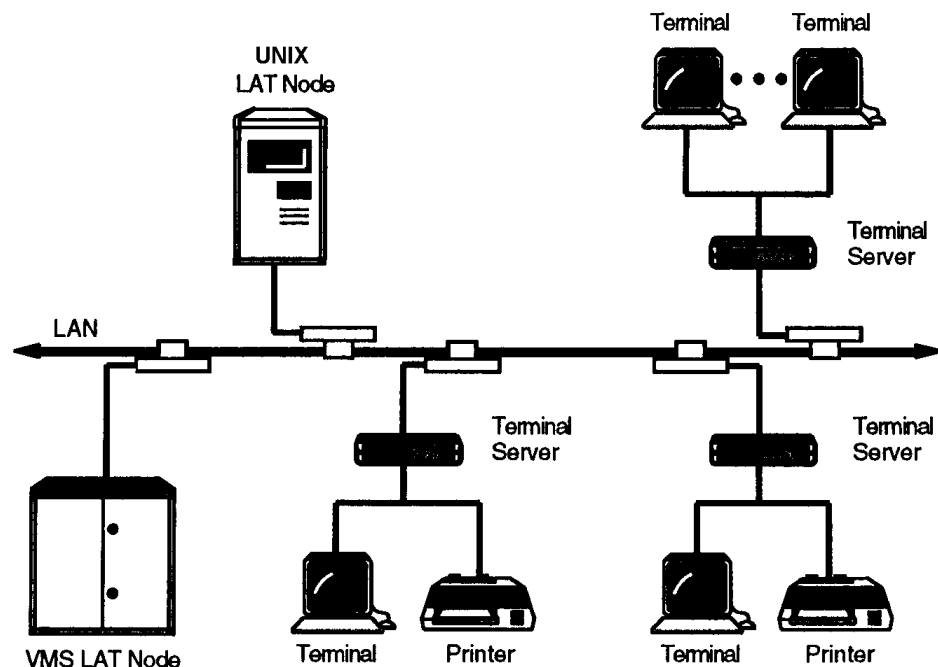
A terminal server is an Ethernet communications device that can establish and manage virtual circuit connections between serial devices and LAT hosts.

All Compaq terminal servers support:

- Terminals
- Printers

**Sample LAT Network**

The following figure shows a sample LAT network.



ZKO-055-000148-05-RGS

## Service Advertisements

An OpenVMS service node advertises its services over the network at regular intervals so that terminal servers know where network resources are available. The service announcement provides:

- Physical node name
- Service names
- Description of services
- Rating

Terminal servers listen to the Ethernet advertisement and record information.

When a user terminal requests service, the terminal server node requests connection to the appropriate OpenVMS service node.

## Types of Service Requests

There are two types of service request:

1. User-initiated request (local access)
  - Connection request from the user on a server port to a LAT service.
  - An example is a request from a user to connect to a service node.
2. Host-initiated request (remote access)
  - Connection from a LAT service node to a terminal server.
  - An example is a request from a service node to print a job on a printer attached to a terminal server.

## Terminal Server Concepts

A terminal server is a computer with:

- CPU
- Memory
- Operating system
- I/O controllers

Most Compaq terminal servers are nodes without disks.

When the terminal server is started up following a shutdown or power loss, the software image is downline loaded from another node on the network known as the load host.

Devices attach to the server at terminal server *ports*.



---

### Note

A terminal server is a component on a local area network that connects devices such as terminals, printers, modems, and computers to the network. It allows these devices to be located anywhere in the local area network.

Note: The terminal server DECserver 90Lplus does not need to downline load. The operating system is contained in firmware within the device.

---

## Terminal Server Operating Modes

In *local mode*, an interactive user enters server commands to communicate directly with the server.

In *service mode*, an interactive user communicates with a service. The server does not usually interpret data entered by the user.



### Note

In local mode, an interactive user enters server commands to communicate directly with the server. Local mode is distinguished by the prompt that appears on the port user's terminal. The default prompt is

```
Local>
```

At the local prompt, the user can enter commands such as the following which display the user's port characteristics on the terminal:

```
Local> SHOW PORT CHARACTERISTICS
```

In service mode, an interactive user communicates with a service in an active session. The server does not interpret data entered by the user. Data is passed to the service node.

To the user the connection appears to be direct to the service node. The user sees the prompt for the operating system or application program, not the server's prompt.

## Terminal Server Differences

Some features of terminal servers are device-specific.

- Operating system used
- Method of storing and loading the operating system
- Database storage and method of update

Refer to the documentation for information about specific devices.



---

**Note**

---

## Terminal Server Databases

There are two types of terminal server databases:

- Permanent database
  - Values set in the permanent database remain constant through initializations, power losses, and port logins.
  - Privileged users can modify server and service characteristics in the permanent database. These changes do not take effect until the server is reinitialized.
  - Privileged users can modify port characteristics. Changes to port characteristics become operational when the user logs out of the port and then logs back in or when the server is reinitialized.
- Operational or volatile database
  - After initialization, the operational database is a copy of the permanent database.
  - Privileged users can modify some server and service characteristics. Changes take effect immediately and remain in effect until the server is initialized.
- Both privileged and nonprivileged users can modify some port characteristics in the operational database. Changes take effect immediately and remain in effect until the server is initialized or the port is logged out.



---

**Note**

Details of the databases are server-specific. Students should refer to the manual for their particular terminal server.

---

## Troubleshooting Terminal Servers

Whenever you have a situation where a device works for a period of time, but does not work after the system is rebooted, investigate the following:

- Does this device use a volatile database?
- Were any definitions made to the volatile database that were not made permanent?



---

**Note**

This applies to more than just terminal servers. When using OpenVMS, you must understand how definitions are made permanent.

- Stored in a nonvolatile database usually on disk
- Defined each time the system is booted using a startup command procedure

When something works for a time and then does not work after the system is rebooted, port logged out, and so on, suspect a problem with making the definition permanent.

---

## Displaying Server Characteristics

A general user can enter commands in local mode at the terminal server prompt to display the information in the database.

```
Local> SHOW PORT CHARACTERISTICS
```

```
Local> SHOW SERVICE CHARACTERISTICS
```

## Testing a Terminal Server Port

To verify that a terminal server port is working correctly, a privileged user can:

- Log in to the server.
- Test the port using the command:

```
Local> TEST PORT # COUNT #
```

The server sends out a rotating character pattern to the port, which is displayed on the terminal.

If the pattern is not displayed or if the display is not consistent, there is something wrong with any of the following:

- Server
- Port
- Terminal
- Cabling out to the terminal

**Note**

When a remote print queue (printer attached to terminal server) is not working, you can isolate the problem by placing a terminal at the end of the printer cable and conducting the port test. If the test is good, then you can eliminate problems with:

- Server
- Port
- Cabling out to the terminal

This does not eliminate the printer itself as the cause of the problem.

The next step is to reconnect the printer and try the COPY command. If COPY fails, the problem is with the printer.

---

**Example: Terminal Server Port Test**

The following test indicates that the port is working correctly because the test pattern is consistent.

```
# [password]
```

```
Network Access SW V1.5 for DS700-16 BL95-33 ROM V3.4-9 Uptime: 35
18:54:58
```

```
Please type HELP if you need assistance
```

```
Enter username> SYSTEM
```

```
Local> TEST PORT COUNT 3
```

```
!"#$%&'()*+,-
./0123456789:;<literal>(<)=?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[^_`abc
!"#$%&'()*+,-
./0123456789:;<literal>(<)=?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[^_`abcd
!"#$%&'()*+,-
./0123456789:;<literal>(<)=?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[^_`abcde
```

```
Local>
```

## LAT Management Tools

### LAT Control Program

The LAT Control Program (LATCP):

- Is bundled with the OpenVMS operating system
- Is used to configure and control the LAT software from a LAT service node

### Uses for the LATCP

The LATCP allows you to do the following:

- Start and stop the LAT port driver (LTDRIVER) on an OpenVMS system.
- Specify operational characteristics for your service node and its services.
- Display the status for your LAT service node.
- Show and zero LAT counters.

### Invoking the LATCP

To invoke the LATCP, enter the command:

```
$ RUN SYS$SYSTEM: LATCP
```

You can enter LATCP commands at the prompt.

### Sample LATCP Commands

The following are some commands that you can enter using the LATCP:

|                           |                                              |
|---------------------------|----------------------------------------------|
| <b>SET NODE/STATE=ON</b>  | Starts the LAT port driver on the local node |
| <b>SET NODE/STATE=OFF</b> | Stops the LAT port driver on the local node  |
| <b>SHOW (entity)</b>      | Displays the LAT information specified       |
| <b>SET (entity)</b>       | Modifies the LAT information specified       |

### LATCP DCL Command Line Interface

You can define a DCL symbol to execute a single LATCP command in a command procedure.

```
$ LCP ::= $LATCP
```

Within a command procedure, you can enter a single command to the LATCP.

```
$ LCP START NODE
```

## Troubleshooting Printer Queue Problems

### Overview

There are many variables in managing a printer queue or remote printer queue. To troubleshoot problems with printer queues, it is most important to define and isolate the problem.

Remote print queue problems usually fall into one (or more) of the following categories:

- OpenVMS print queue problems
- LAT symbiont problems
- LAT device-mapping problems
- Hardware problems

This section discusses some printer queue troubleshooting tips.

## Some Troubleshooting Tips

### Questions to Ask

Obtain more information about the problem.

- Is this a new remote print queue installation? Has it ever worked?
- How many remote print queues are on this system? Are any queues working?
- If there is another print queue on this terminal, is it working?
- Did anything change immediately before the printer stopped working?
- Is there a queue on another system in the network pointing to this printer? Is it working?
- What error messages do you get when you print to the queue?
- What command was used to issue the print request?
- Is /RETAIN=ERROR set on the queue? If not, set it and try again.



#### Note

Problem Report: "My file won't print."

Question: "What error message are you getting?"

Answer: "Oh... Well let's see....Something about file not found."

```
$ PRINT MYFILE.DAT
%PRINT-E-OPENIN, error opening
WORK12:[SMITH]MYFILE.DAT; as input
-RMS-E-FNF, file not found
%PRINT-F-CREJOB, error creating job
-RMS-E-FNF, file not found
```

**Always get the user to write down the error message.**

### Starting the Investigation

If the problem is with a new remote print queue that has never worked, the problem is most likely to be found in:

- LAT device mapping
- Hardware

If the problem is with an existing queue that has had no recent changes made to it, but just stopped working, start by investigating:

- Device.... Is it on?
- OpenVMS queuing system
- LAT symbiont

Of course, if you are aware that a change has been made, start by investigating the change.

**Note**

What happens when two execution queues are mapped to the same printer? This is not relevant to the text because we always show one execution queue mapped to a printer.

**What happens when a print command is issued?**

```
$ PRINT/QUEUE=LN03 MYFILE.TXT
```

In this example, LN03 is a remote print queue on a terminal server ZK01123, PORT\_2.

NODEA on which the command was issued sends a multicast message onto the Ethernet, and that message basically says “Hey all you servers! Whoever has a name of ZK1123 and a port name PORT\_2, would you please initiate a connection back to me?” As you can see, all NODEA knows about LTA8000: is what we’ve told it: the server name and the port name. This is a multicast message to all the servers, and ZK1123 creates a session between his port 2 and the requesting NODEA. NODEA pumps the job out to the port, and then disconnects from the port, logging the port out, and freeing the port for further requests.

**What happens when two execution queues are mapped to the same remote printer?**

While the printer is busy printing NODEA's job, NODEB sends print jobs to its LN03 queue which is mapped to the same server port. NODEB sends out a multicasce connection request message and the server acknowledges the request, but port 2 is currently busy with NODEA's file. The server puts NODEB's connection request into the NODEB connection queue and sends a response to NODEB that this request has been queued. if NODEA's job is long enough, NODEB's queue may eventually become “stalled,” waiting for port 2 to become available.

When NODEA's job finishes, the session between NODEA and ZK1123 port 2 is terminated and port 2 is logged out. ZK1123 looks in the connection queue and sees that NODEB is next in the queue for port 2. The connection is made and NODEB's print queue goes from “stalled” to “busy.”

---

**Isolating the Point of Failure**

Replace the printer with a terminal connected to the printer port.

Can you reach the node that hosts the printer queue?

## Using the COPY Command to Test a Printer

A powerful tool to isolate remote printer queue problems is the COPY command. However, if a device is set spooled, even a privileged user cannot use the COPY command to test the printer. To use this tool on a spooled printer:

- Determine the LTA# of the device in question using the command:  
`$ SHOW QUEUE/FULL/ALL queue-name`
- Stop the queue using the command:  
`$ STOP/QUEUE/RESET queue-name`
- Set the device to NOSPOOL.  
`$ SET DEVICE/NOSPOOL LTA####`
- Copy a file to the device. Make sure the file is at least one page long.  
`$ COPY/LOG file-name LTA####:`
- Repeat the copy command six or more times.
- If the copy command works, check the print queue or the way the device is spooled.
- If the copy command fails, check:
  - Mapping to the port on the server
  - Cable
  - Printer
- To restore the original condition, use the SET DEVICE command and start the queue that was stopped.



### Note

If a printer is set spooled and a privileged user tries to copy a file to the printer, the file is written to the designated disk and then submitted to the print queue for the printer. This does not test the printer.

You can only test the printer when it is set to nospool. To test the printer, you want the file to go directly to the printer instead of through the queue.

---

## Obtaining Information About the Queue

Check the status of the queue using the SHOW QUEUE/FULL/ALL command. This provides the following information:

- Type of queue: printer, generic, logical
- Queue state
- Job state

```
$ SHOW QUEUE/FULL/ALL LPS20$PRTA
```

```
Printer queue LPS20$PRTA, busy, on NODE01::LTA1, mounted form
DCPS$DEFAULT
```

```
(stock=DEFAULT)
```

```
/BASE_PRIORITY=4 /DEFAULT=(FORM=DCPS$DEFAULT (stock=DEFAULT))
```

```
/NOENABLE_GENERIC /LIBRARY=DCPS_LIB /OWNER=[1,4] /PROCESSOR=DCPS$SMB
```

```
/PROTECTION=(S:M,O:D,G:R,W:S) /SCHEDULE=(NOSIZE)
```

```
/SEPARATE=(BURST,FLAG,
```

```
TRAILER)
```

```
Entry Jobname Username Blocks Status
```

```
-----
```

```
78 LOGIN SMITH 4 Printing
```

```
Submitted 16-DEC-1998 12:24 /FORM=DCPS$DEFAULT (stock=DEFAULT)
```

```
/PRIORITY=100
```

```
File: _$1$DUA1:[SMITH]LOGIN.COM;25 (printing)
```

```
$
```

Can other nodes print to this printer?

## Checking Mapping to the Server Port

Use the LATCP to examine the state of the LAT ports on a node.

\$ MC LATCP

LATCP> SHOW PORT

| Port Name | Port Type | Status | Remote Target (Node/Port/Service) |
|-----------|-----------|--------|-----------------------------------|
|-----------|-----------|--------|-----------------------------------|

-----

|        |             |          |               |
|--------|-------------|----------|---------------|
| _LTA1: | Application | Inactive | ZK1115/3062A/ |
|--------|-------------|----------|---------------|

|        |             |          |               |
|--------|-------------|----------|---------------|
| _LTA2: | Application | Inactive | ZK1115/3062B/ |
|--------|-------------|----------|---------------|

|          |             |          |             |
|----------|-------------|----------|-------------|
| _LTA101: | Application | Inactive | ZK1123/12D/ |
|----------|-------------|----------|-------------|

|          |             |          |             |
|----------|-------------|----------|-------------|
| _LTA102: | Application | Inactive | ZK1122/12B/ |
|----------|-------------|----------|-------------|

|           |         |          |    |
|-----------|---------|----------|----|
| _LTA5016: | Forward | Inactive | // |
|-----------|---------|----------|----|

|           |             |        |    |
|-----------|-------------|--------|----|
| _LTA5192: | Interactive | Active | // |
|-----------|-------------|--------|----|

|           |             |        |    |
|-----------|-------------|--------|----|
| _LTA5195: | Interactive | Active | // |
|-----------|-------------|--------|----|

|           |             |        |    |
|-----------|-------------|--------|----|
| _LTA5210: | Interactive | Active | // |
|-----------|-------------|--------|----|

LATCP> SHOW PORT LTA1

Local Port Name: \_LTA1: Local Port Type: Application (Queued)

Local Port State: Inactive

Connected Link:

Target Port Name: 3062A Actual Port Name:

Target Node Name: ZK1115 Actual Node Name:

Target Service Name: Actual Service Name:

LATCP>

## Getting a Test Sheet from the Printer

Refer to the printer instructions to determine the appropriate printer settings.

## Checking the Hardware

Install a terminal at the end of the printer cable and repeat the COPY command.

- If the COPY command fails, replace the cable and try again.
- If the COPY command works, the problem is with the printer.
  - Check the power source.
  - Call the appropriate printer repair service.



---

**Note**

Enter the following to obtain more information regarding a LAT error message such as %LAT-F-NOMSG, message number 01749FBC:

```
$ SET MESSAGE SYS$MESSAGE:NETWRKMSG
$ EXIT %X01769FBC
%LAT-F-CONTIMEOUT, connection timed out, server not available,
or incorrect
server name specified
```

---

## Print Job in Pending or Holding State

A print job goes into a PENDING or HOLDING state whenever the job is not eligible to print. Possible causes include:

- The queue is currently printing another job.
- The queue is stopped or stalled.
- The print job was submitted with an /AFTER= time qualifier.
- The print job is waiting for a paper stock to be mounted on the queue.
- The print job was submitted with /CHARACTERISTICS that are not identified with the queue.
- There is a BLOCK\_LIMIT on the queue, and the print job does not fall within the specified block range.

## Example: Print Job Submitted with /AFTER Qualifier

The following print job was submitted with the /AFTER qualifier to the PRINT command.

```
$ PRINT/QUE=LPS20$PRT1/AFTER=14:30 LOGIN.COM
```

Job LOGIN (queue LPS20\$PRT1, entry 91) holding until 16-DEC-1998 14:30

```
$ SHOW QUEUE/FULL/ALL LPS20$PRT1
```

Printer queue LPS20\$PRT1, idle, on *NODE01::PRT1*, mounted form  
DCPS\$DEFAULT

```
/BASE_PRIORITY=4 /DEFAULT=(FORM=DCPS$DEFAULT (stock=DEFAULT))
```

```
/NOENABLE_GENERIC /LIBRARY=DCPS_LIB /OWNER=[1,4] /PROCESSOR=DCPS$SMB
```

```
/PROTECTION=(S:M,O:D,G:R,W:S) /SCHEDULE=(NOSIZE)
```

```
/SEPARATE=(BURST,FLAG,
```

```
TRAILER)
```

| Entry | Jobname | Username | Blocks | Status                          |
|-------|---------|----------|--------|---------------------------------|
| 91    | LOGIN   | SMITH    | 4      | Holding until 16-DEC-1998 14:30 |

-----

Submitted 16-DEC-1998 14:06 /FORM=DCPS\$DEFAULT (stock=DEFAULT)

/PRIORITY=100

File: \_\$1\$DUA1:[SMITH]LOGIN.COM;25

```
$ SET ENTRY 91/AFTER=14:05
```

```
$ SHOW ENTRY
```

| Entry | Jobname | Username | Blocks | Status   |
|-------|---------|----------|--------|----------|
| 91    | LOGIN   | SMITH    | 4      | Printing |

-----

On busy printer queue LPS20\$PRT1

## How OpenVMS Formats a Print Job

### Overview

Three components contribute to formatting a print job:

- Print symbiont
- Driver
- Printer

You should understand the contribution of each of these components to troubleshoot problems with print job formats.

This section discusses:

- How a print file is formatted
- Troubleshooting file output problems

## How a Print File Is Formatted

### Print Symbiont

The print symbiont formats the data in a file according to the form assigned:

- To the print queue
- By the DCL command  
\$ PRINT/FORM=form\_name file-name

Example form:

```
$ DEFINE/FORM/WIDTH=80/WRAP/MARGIN=(TOP=3,BOTTOM=6)/LENGTH=66 DEFAULT 0
```

With the example form, the print symbiont counts lines as they are printed. The symbiont “knows” that it must leave a bottom margin of six blank lines after it prints 60 lines as follows:

- Three for the top margin
- Fifty-seven lines of text

The symbiont inserts a form-feed character (<FF>) into the data stream to position the next page.

### Driver Reformats the File

The print symbiont sends the file to the device driver, which reformats the file according to its own settings.

The device driver settings are determined by one of the following DCL commands:

- SET TERMINAL (for terminal printers)
- SET PRINTER (for line printers)

For example, a port is defined:

```
WIDTH=60
/WRAP
```

If the driver receives 80 characters of data from the print symbiont, it inserts a line-feed character (<LF>) after column 60.

The print symbiont sets the page length of the port to that specified on the form.

The device driver also counts the lines as they are printed and looks for the form-feed character (<FF>) in the data stream FF. If it finds an FF, it acts according to the device setting.

---

| Setting | Action                                                                                                                                                  |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| /FORM   | The driver passes the FF character to the printer.                                                                                                      |
| /NOFORM | The driver converts the FF character to the number of LF characters needed to get the printer to advance to the top of the next physical page of paper. |

---



---

**Note**

The print symbiont in this example sends the file:

- 3 blank lines
- 57 lines of text
- FF

The print symbiont sets the device page length to 66 from the form definition.

---

## Printer

The driver sends the formatted file to the printer.

Printer characteristics supersede those of the symbiont and the driver. Usually these setup values are for margins and form length.

Printer settings are device specific. They are set by DIP switches on the printer or escape sequences in:

- The file
- A setup module specified in the form

For example, if both the symbiont and the driver format a file with a line 80 characters wide, but the printer is set for 60 characters, the printer allows 60 characters on the line.

In another example, the printer is set up with a length of 40 lines and a bottom margin of five lines.

- When 35 lines of text are printed, the printer advances five lines to the next page.
- If the paper is 66 lines long, the new page starts on the lower half of the 66-line page instead of advancing to the physical top of the next page, 31 lines away.

## Managing Printer Settings

Because the printer settings override those of the symbiont and driver:

- Set the printer length and width to be greater than or equal to the longest and widest setting on your forms.
- Set the port to /NOFORM so the page length is controlled by the print form by having the driver issue multiple LF characters to get to the top of the form.

## Troubleshooting File Output Problems

### Problem Definition

The output from a print job is not formatted according to the user's requirements.  
Find out as closely as possible how the user wants the file to be formatted.

### Getting More Information

Get more information by:

- Determining the command that was used to print the file
- Displaying the default form for the queue
- Displaying the form that was specified in the print command (if any)
- Determining the device characteristics

## Summary

### OpenVMS Batch and Print Queue System Concepts

- The job controller and queue manager processes play important roles in the OpenVMS batch and print queue system.
- The queue database provides nonvolatile storage for queue definitions and forms.
- A print symbiont processes print jobs.
- The system and network manager customizes print queues.
- Use print forms to customize print queues and to set options for print jobs.
- Device control modules are associated with print forms.
- The system manager can control access to queues using UIC and ACL protection.

### Remote Print Queue Concepts

- Terminal servers use the LAT network protocol to communicate on an Ethernet network.
- A terminal server is a communications device used to connect terminals and printers to the network.
- Use the LAT Control Program (LATCP) to configure and control the LAT software.

### Troubleshooting Print Queue Problems

Isolate the problem to the:

- Print queue system
- Symbiont
- LAT network
- Hardware

### How OpenVMS Formats a Print Job

- Three components contribute to the formatting of a print job:
  - Print symbiont
  - Device driver
  - Printer
- To troubleshoot print format problems, obtain a clear definition of the problem.

## Chapter Terms

### **local mode**

A user communicating with terminal server software directly is using local mode.

### **port**

A hardware connector on the terminal server where a device plugs in, and software associated with it (port characteristics).

### **protocol**

A set of rules for communicating.

### **queue**

A sequence of jobs to be processed, for example, a batch job queue or a printer job queue. Processing occurs primarily in first-in/first-out (FIFO) order, but does not reflect the priority of the process that submitted the job.

### **service**

A software application (interactive login, MAIL, PHONE) or hardware device (printer, dial-in lines) available on the network.

### **service mode**

A user communicating with an available service through a terminal connected to a terminal server is using service mode.

### **service node**

A computer system on the Ethernet LAN that offers service.

### **symbiont**

A process that transfers record-oriented data to or from a device. For example, an input symbiont transfers data from card readers to disks. An output symbiont transfers data from disks to line printers.

### **terminal server**

A type of communications server that connects terminals and other devices to the Ethernet LAN.

---

# ***Troubleshooting OpenVMS Cluster Systems***

## ***Module 4***

## Introduction

An OpenVMS Cluster system is a highly integrated organization of OpenVMS VAX and Alpha computers that share processing resources, data storage, and queues under a single OpenVMS security and management domain. OpenVMS Cluster systems are used to improve system availability.

This module discusses special tools and techniques used to perform troubleshooting in an OpenVMS Cluster system environment.

## Objectives

To troubleshoot in an OpenVMS Cluster system environment, you should be able to:

- Describe OpenVMS Cluster system concepts.
- Apply the troubleshooting methodology to solve problems when an OpenVMS Cluster system node does not initialize or join the cluster system.
- Apply the troubleshooting methodology to solve problems when an OpenVMS Cluster system hangs.

## Resources

- *Guidelines for OpenVMS Cluster Configurations*
- *OpenVMS Cluster Systems*
- *OpenVMS System Management Utilities Reference Manual*
- *OpenVMS System Manager's Manual*
- *DECnet-Plus for OpenVMS Introduction and User's Guide*
- *DECnet-Plus for OpenVMS Network Management*
- *OpenVMS System Dump Analyzer Utility Manual*



---

**Note**

Study Time

Materials: 3 hours

Practice: 2 1/2 hours

---

# OpenVMS Cluster System Concepts

## Overview

To troubleshoot OpenVMS Cluster system problems, you should be able to describe the diagnostic messages produced during both cluster initialization and normal operations. There are some special tools used to troubleshoot problems in OpenVMS Cluster systems.

This section discusses:

- OpenVMS Cluster system software components
- Types of OpenVMS Cluster system nodes
- Cluster communication
- Cluster formation
- OpenVMS Cluster system diagnostic messages
- Using NCP in an OpenVMS Cluster system
- OpenVMS Cluster system bugchecks

**Note**

This material should be a review from the *OpenVMS System and Network Node Management* course.

You may also give the class a diagram of the OpenVMS Cluster system that is being used for the laboratory examples and discuss the importance of having such a diagram available for all clusters they manage.

---

## OpenVMS Cluster System Software Components

The software components responsible for the OpenVMS Cluster system include the following:

| Component                                        | Description                                                                                                                                                                                                                                                                                      |
|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System communications services (SCS)             | Implements intercomputer communication according to the Compaq systems communications architecture (SCA)                                                                                                                                                                                         |
| Connection manager                               | Uses SCS to maintain the integrity of the cluster by coordinating participation of cluster members; prevents cluster partitioning through the use of quorum                                                                                                                                      |
| Distributed lock manager                         | Synchronizes access to resources                                                                                                                                                                                                                                                                 |
| Distributed job controller and the queue manager | Make batch and print queues available across the cluster                                                                                                                                                                                                                                         |
| VAXport drivers                                  | Control the communication paths between local and remote ports. Examples include: <ul style="list-style-type: none"><li>■ PADRIVER for the CI</li><li>■ PEDRIVER for the LAN</li><li>■ PIDRIVER for the DSSI</li><li>■ PEDRIVER for the FDDI</li><li>■ XXDRIVER for the Memory Channel</li></ul> |
| Distributed file system                          | Allows all computers to share mass storage devices and works with VMS Record Management Services (VMS RMS) to provide file sharing to the record level across the cluster                                                                                                                        |
| MSCP server                                      | Makes locally connected disks available to all nodes on the cluster                                                                                                                                                                                                                              |
| TMSCP server                                     | Makes locally connected tape drives available to all nodes on the cluster                                                                                                                                                                                                                        |



---

### Note

System communications services (SCS) is a set of system routines that implement device-independent communications for OpenVMS systems. The system communications architecture simply lays out a set of rules that all communications should follow.

The *connection manager*:

- Determines and coordinates membership in the cluster
  - Creates a cluster when the first active computer is booted
  - Reconfigures the cluster when computers join or leave it
-



**Note**

The value of the system parameter VAXCLUSTER is checked at boot time, and it determines whether or not the image SYS\$CLUSTER is loaded:

| VAXCLUSTER | Action                                                                                                                                                                                                                   |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0          | SYS\$CLUSTER is not loaded; the system boots standalone.                                                                                                                                                                 |
| 1          | SYS\$CLUSTER is loaded if one or more of the following is true:<br>There is a CI, Ethernet, or SHAC adapter on the system.<br>The system is booting from a local DSA disk.<br>The system parameter NISCS_LOAD_PEA0 is 1. |
| 2          | SYS\$CLUSTER will be loaded.                                                                                                                                                                                             |

Another way of presenting this is:

| VAXCLUSTER | Action                             |
|------------|------------------------------------|
| 0          | Do not join or form a cluster      |
| 1          | Join a cluster if one is available |
| 2          | Always join or form a cluster      |

Reconfigures the cluster when computers join or leave it.

OpenVMS Cluster members can share various data and system resources, such as disk volumes. To achieve the coordination that is necessary to maintain resource integrity, the members must share a clear sense of cluster membership, which is maintained by the connection manager.

Within a single OpenVMS Cluster system, the OpenVMS operating system guarantees the integrity of shared resources by carefully coordinating their use. However, because use of shared resources is not coordinated between computers in separate clusters (a condition known as *cluster partitioning*) the connection manager prevents this condition using a scheme called *quorum*.

## **Types of OpenVMS Cluster System Nodes**

### **Satellite Node**

A satellite node does not have a system disk of its own. The OpenVMS operating system is downline loaded to the satellite from a boot server over a network.

### **Boot Node**

Boot nodes are able to downline load the OpenVMS operating system.

A local disk on the boot node contains the cluster startup and authorization and the directory roots from which satellite nodes are booted. This disk can be the boot node's system disk or another local disk on the boot node.

The MSCP server is always loaded on a boot node.

### **Participating Node**

A node can boot from its own system disk, but also participate in cluster activity. In a mixed architecture cluster, for example, there are separate boot servers for OpenVMS VAX and Alpha systems.

## Cluster Communication

### Overview

There can be many clusters on a given network. The connection manager uses two key pieces of information to determine which systems should be part of any given cluster:

- Cluster group number
- Cluster password

### Cluster Group Number

You establish the cluster group number with the CLUSTER\_CONFIG utility or the SYSMAN utility. It can be any number between 1 and 4095 or 61440 to 65535. Each cluster on the local network must have a different group number.

You can display the group number from the SYSMAN utility:

```
SYSMAN> CONFIG SHOW CLUSTER_AUTHORIZATION
```

```
Node BROWNY: Cluster group number: 65084
```

```
Multicast address: AB-00-04-01-3C-FF
```

### Cluster Password

You establish the cluster password using the CLUSTER\_CONFIG utility or the SYSMAN utility. You cannot display the cluster password.

## Cluster Formation

### Overview

Cluster formation involves a series of interconnected steps as outlined below. In this discussion, a mixed interconnect cluster is assumed: A boot server connects to satellite nodes over an Ethernet.

1. **Check VAXCLUSTER Parameter.**  
When an OpenVMS system is booted, it checks the value of the system parameter VAXCLUSTER to determine whether to join or form a cluster.
2. **Boot server sends HELLO packets.**  
In a mixed interconnect cluster, the boot server broadcasts a message over the local network that is addressed to all systems sharing the same cluster group number. The message indicates that the sender is available to downline load an operating system.  
The message is made up of the multicast address AA-00-04-01-xx-xx, where xx-xx is the cluster group number plus one.
3. **Satellite requests boot assistance.**  
A satellite that is booting recognizes the HELLO message as coming from a member of its group, and sends a request to the local network for boot assistance. All systems on the network that have boot service enabled check their databases to see if they are configured to boot the satellite.
4. **Boot server establishes connection.**  
All boot servers configured to boot the satellite send a verification message to the satellite indicating that boot service is available. The satellite acknowledges the first such message received, and ignores all others.
5. **Boot server sends secondary bootstrap.**  
Upon receipt of the acknowledgement from the satellite, the boot server begins to send portions of the secondary bootstrap image. Each portion is acknowledged by the satellite and loaded into its memory. After the final packet has been sent, the boot server sends the address to which the satellite should jump to execute the secondary bootstrap program.

6. **Satellite loads operating system**

The secondary bootstrap program, through NISCS\_LOAD, sends a request for service that contains the cluster group number and password, as well as the device name of the system disk. A connection is made to any MSCP server that shares the group number and password, and is serving the specified disk.

The satellite loads SYSBOOT through the MSCP server and transfers control to it. The boot process then continues much as it would if the system were booting standalone.

## OpenVMS Cluster System Diagnostic Messages

### Software Sources of Diagnostic Messages

Several types of diagnostic messages are displayed by an OpenVMS Cluster system, including:

- Connection manager messages
- VAXport error messages
- Error log entries for cluster-available devices
- HSC error messages and fault codes

### Connection Manager Messages

Connection manager (%CNXMAN) messages:

- Provide information about OpenVMS Cluster system transitions
- Are displayed at the console terminal
- Are generated by events such as:
  - When a node joins or leaves the cluster
  - When the state of the cluster changes



---

**Note**

---

## Sample Connection Manager Messages

The following are some examples of connection manager messages:

```
%CNXMAN, Lost connection to system NODEB
%CNXMAN, Waiting to form or join OpenVMS Cluster
%CNXMAN, Sending OpenVMS Cluster membership request to system NODEB
%CNXMAN, Completing OpenVMS Cluster state transition
%CNXMAN, Now a NODEcluster member -- system NODEA
%CNXMAN, Established "connection" to quorum disk
%CNXMAN, Proposing modification of quorum or quorum disk membership
%CNXMAN, Quorum lost, blocking activity
%CNXMAN, Error reading quorum disk
%CNXMAN, Lost "connection" to quorum disk
%CNXMAN, Proposing modification of quorum or quorum disk membership
%CNXMAN, Timed-out lost connection to system NODEB
%CNXMAN, Aborting VAXcluster state transition
%CNXMAN, Proposing reconfiguration of the VAXcluster
%CNXMAN, Removed from VAXcluster system NODEB
%CNXMAN, Completing VAXcluster state transition
%CNXMAN, Discovered system NODEB
%CNXMAN, Established connection to system NODEB
```

Note: Even though the official name is OpenVMS Cluster, the messages still say VAXcluster.

## VAXport Error Messages

| Messages from the... | Are prefixed by... |
|----------------------|--------------------|
| CI port              | %PA                |
| Ethernet port        | %PE                |
| DSSI port            | %PI                |
| FDDI port            | %PE                |
| Memory Channel       | %xx                |
| SCSI port            | %PK                |

Messages appear in the error log and at the console.

- Use DECEvent software or the SHOW ERROR command to see errors.
- Use DIAGNOSE (OpenVMS Alpha) or ANALYZE/ERROR\_LOG (OpenVMS VAX) to examine the error log.



### Note

VAXport errors are not seen in OPCOM.

## Port Polling

The CI port driver (PADRIVER) performs configuration polling to find other active ports.

- Normally polling occurs every 5 seconds (controlled by system parameter PAPOLLINT).
- It probes cable A, cable B, cable A, and so on.
- It sends and receives REQID packets.
- A handshake starts when packets have exchanged successfully.
- Port drivers then exchange datagrams about the systems.
- If successful, a virtual circuit is opened.

## Ethernet Communications

- Every 3 seconds, the port emulator driver (PEDRIVER) sends HELLO messages to an address derived from the group number.
- Nodes send the HELLO back.
- When the driver receives a HELLO from a node that does not currently have an open circuit, it attempts to create one.
- A standard three-message handshake creates the virtual circuit.

## System Communications Services (SCS)

- SCS handles communications for
  - Disk class driver
  - Connection manager
  - MSCP server
- SCS is implemented in the VAXport driver.
- As system services discover each other on the system, SCS connections are established.

### Example: VAXport Error Messages

The following are examples of VAXport error messages:

%PAA0, Software is Closing Virtual Circuit - REMOTE PORT 4

%PAA0, Software is Closing Virtual Circuit - REMOTE PORT 2

%PAA0, Path #1. Loopback has gone from GOOD to BAD

%PAA0, Path #0. Loopback has gone from GOOD to BAD

%PAA0, Path #0. Loopback has gone from BAD to GOOD

%PAA0, Software is Closing Virtual Circuit - REMOTE PORT 3

### Error Log Entries for Cluster-Available Devices

Many devices, like port drivers, generate errors.

- Errors for nonshared hardware (memory) are logged to the error log file associated with that CPU.
- Errors for shared devices, such as disks or tapes, are directed to only one system.
- Error log entries for HSC or DSSI device errors are:
  - Logged by the system that requested the associated I/O operation
  - Logged as a port error if not associated with I/O operation (HSC ERROR LOGGING DATAGRAM)
- Error log entries for MSCP- or TMSCP- served device errors are:
  - Logged by the system on which the device is local
  - If dual-ported, logged by the system that served the associated I/O operation

## HSC Diagnostic Messages and Fault Codes

Diagnostic messages appear at the HSC console terminal.

- If the FAULT light is on, push and release the FAULT switch: The lights will display a pattern indicating a certain fault. Record this pattern.
- If the fault is TU58 drive FAILURE — HSC50 — or MISSING REQUIRED FILES:
  - Make sure the HSC operating system tape is in TU58 drive 0.
  - Try the backup HSC operating system tape.
- For other faults:
  - Power down and reboot the HSC unit.
  - Call your Multivendor Customer Services representative.

If your HSC disks are dual ported, an HSC unit can fail without affecting the rest of the cluster.

- Disks automatically fail over if they are not mounted /FOREIGN, so you may not notice that one of the HSC units has failed.
- Diagnostics are especially important. Look at the indicator lights and console printout regularly.



### Note

These diagnostics may just be status displays or error messages. Monitor the console output regularly to learn what is normal for your system.

See the *Hardware Operations Guide* for information about the HSC switches. The SECURE/ENABLE switch inside the HSC has two positions: secure and enable. In the secure position the switch prevents:

- Use of the SET utility
- Use of the INIT switch (used to reboot the HSC)

In the enable position, it permits use of the FAULT and ONLINE switches.

## Using NCP in an OpenVMS Cluster System

The Network Control Program (NCP) displays information about:

- Nodes
- Circuits
- Lines
- The network

### NCP SHOW Commands

The following table summarizes common NCP SHOW commands.

| This command...                     | Displays information about...         |
|-------------------------------------|---------------------------------------|
| SHOW NODE node-name CHARACTERISTICS | A single node                         |
| SHOW KNOWN CIRCUIT CHARACTERISTICS  | All currently defined DECnet circuits |
| SHOW KNOWN LINE CHARACTERISTICS     | All currently defined DECnet lines    |
| SHOW EXECUTOR CHARACTERISTICS       | The current system                    |

### Example: Displaying Node Characteristics

The following example uses NCP to display the characteristics of a node.

```
NCP> SHOW NODE DBLDOG CHAR
```

```
Node Volatile Characteristics as of 15-DEC-1998 14:40:16
```

```
Remote node = 62.642 (DBLDOG)
```

```
❶ Hardware address      = 08-00-2B-30-A0-D2
Tertiary loader        = SYS$SYSTEM:TERTIARY_VMB.EXE
Load Assist Agent      = SYS$SHARE:NISCS_LAA.EXE
Load Assist Parameter  = SYS$SYSDEVICE:<SYS1E.>
```

### Example Note: Displaying Node Characteristics

- ❶ These parameters, defined in the DECnet database, allow BROWNY to initiate the cluster system boot process for the node DBLDOG.

### Example: Displaying Circuit Characteristics

The following example uses NCP to display circuit characteristics.

```
VRROOM> MC NCP SHOW KNOWN CIRCUITS CHAR
```

Known Circuit Volatile Characteristics as of 15-DEC-1998 14:42:41

```
❶ Circuit = ISA-0

❷ State          = on
❸ Service        = enabled
Designated router = 62.1022
Cost              = 4
Maximum routers allowed = 33
Router priority   = 64
Hello timer       = 15
Type              = Ethernet
Adjacent node     = 62.1022
Listen timer      = 45
```

### Example Notes: Displaying Circuit Characteristics

- ❶ The Circuit field displays the name of the circuit as defined in the DECnet database.
- ❷ The State field is on. This circuit is available to process network traffic.
- ❸ The Service field must be set to enabled on the boot node for downline loading to function correctly.

### Example: Displaying Line Characteristics

Use NCP to display line characteristics.

```
VRROOM> MC NCP SHOW KNOWN LINES CHAR
```

Known Line Volatile Characteristics as of 15-DEC-1998  
14:44:10

```
Line = ISA-0
Receive buffers    = 10
Controller         = normal
Protocol           = Ethernet
Service timer      = 4000
Hardware address   = 08-00-2B-2F-F2-B3
Device buffer size = 1498
```

**Example: Displaying Executor Characteristics**

Use NCP to display information about the DECnet executive.

```
$ MCR NCP SHOW EXECUTOR CHAR
```

```
Node Volatile Characteristics as of 15-DEC-1998 14:44:58
```

```
Executor node = 62.905 (VRR00M)
```

```
Identification      = DECnet for OpenVMS VAX V7.0
```

```
Management version  = V4.0.0
```

```
Incoming timer      = 45
```

```
Outgoing timer      = 60
```

```
Incoming Proxy      = Enabled
```

```
Outgoing Proxy      = Enabled
```

```
NSP version         = V4.1.0
```

```
Maximum links       = 32
```

```
Delay factor        = 80
```

```
Delay weight        = 5
```

```
Inactivity timer    = 60
```

```
Retransmit factor   = 10
```

```
Routing version     = V2.0.0
```

```
Type               = nonrouting IV
```

```
Routing timer       = 600
```

```
Broadcast routing timer = 180
```

```
Maximum address     = 1023
```

```
Maximum circuits    = 16
```

```
Maximum cost        = 1022
```

```
Maximum hops        = 30
```

```
Maximum visits      = 63
```

```
Maximum area        = 63
```

```
Max broadcast nonrouters = 64
```

```
Max broadcast routers  = 32
```

```
Maximum path splits   = 1
```

```
Area maximum cost     = 1022
```

```
Area maximum hops     = 30
```

```
Maximum buffers       = 100
```

```
Buffer size           = 576
```

```
Default access        = incoming and outgoing
```

```
Pipeline quota        = 4032
```

```
Alias maximum links      = 32
Path split policy  = Normal
Maximum Declared Objects      = 31
```

## OpenVMS Cluster System Bugchecks

The OpenVMS operating system performs bugcheck operations when it detects conditions that could compromise the integrity of system activity or endanger data integrity.

### CLUEXIT Bugchecks

CLUEXIT bugchecks initiated by the connection manager are triggered by conditions resulting from:

- Configuration errors
- System management errors
- Hardware failures
- Operating system failures

### Common CLUEXIT Bugcheck Conditions

Some of the most common conditions that result in CLUEXIT bugchecks include:

- Cluster connection between nodes is broken for a longer interval than RECNXINTERVAL seconds.
- The partitioning of the cluster.
- The value of system parameter SCSMAXMSG for a node is too small.



### Note

The conditions that most commonly result in CLUEXIT bugchecks are as follows:

- The cluster connection between two computers is broken for longer than RECNXINTERVAL seconds. Thereafter, the connection is declared irrevocably broken. If the connection is later reestablished, either or both of the computers shut down with a CLUEXIT bugcheck.

The condition can occur upon power failure recovery with battery backup, after the repair of an SCS communication link, or after the computer was halted for a period longer than RECNXINTERVAL seconds, and was restarted with a CONTINUE command entered at the operator console.

You must determine the cause of the interrupted connection and correct the problem. For example, if recovery from a power failure is longer than RECNXINTERVAL seconds, you may want to increase the value of the RECNXINTERVAL parameter on all computers.

- Cluster partitioning occurs. A member of a cluster discovers or establishes connection to a member of another cluster, or a foreign cluster is detected in the quorum file. In this case, you must review the setting of EXPECTED\_VOTES on all computers.

The value specified for the SYSGEN parameter SCSMAXMSG on a computer is too small. Verify that the value of SCSMAXMSG on all OpenVMS Cluster computers is set to a value that is at the least the default value.

---

## Failure of Nodes to Boot or Join the Cluster

### Overview

To perform diagnostic and recovery procedures effectively, you should understand the events that occur when a cluster node boots and attempts to join the cluster.

This section discusses:

- OpenVMS Cluster system initialization concepts
- CI node boot problems
- Satellite node initialization problems

# OpenVMS Cluster System Initialization Concepts

## Normal Cluster Node Initialization

The normal sequence of events when a cluster CI node or satellite node boots is as follows:

1. The node boots.
2. The node attempts to form or join the cluster.
3. After the booting computer joins the cluster, normal startup procedures execute.



### Note

The computer boots.

If the computer is a satellite, a message like the following shows the name and LAN address of the MOP server that has downline loaded the satellite. At this point, the satellite has completed communication with the MOP server and further communication continues with the system disk server using OpenVMS Cluster system communications.

```
%OpenVMS Cluster-I-SYSLOAD, system loaded from Node x... (xx-xx-xx-xx-xx)
```

For any booting computer, the OpenVMS “banner message” is displayed.

The SYSINIT process:

- Initializes the cluster connection management.
- Creates the standalone configure process, STACONFIG, which autoconfigures disks and SCS communication ports.
- If the system parameter DISK\_QUORUM indicates that there is to be a quorum disk, STACONFIG polls to discover the disk.

System parameter DISK\_QUORUM specifies the cluster quorum disk name. SYSINIT prints the following message on the console terminal:

```
waiting to form or join an OpenVMS Cluster system
```

If the computer is a member of a local area or mixed-interconnect cluster, the cluster security database is loaded. Optionally, the MSCP server and TMSCP server may be loaded:

```
%OpenVMS Cluster-I-LOADSECDB, loading the cluster security database
%MSCPLOAD-I-LOADMSCP, loading the MSCP disk server
%TMSCPLLOAD-I-LOADTMSCP, loading the TMSCP tape server
```

If the computer discovers a cluster, the computer attempts to join. If a cluster is found, the connection manager displays one or more messages in the following format:

```
%CNXMAN, Sending OpenVMS Cluster membership request to system:
```

Otherwise, the connection manager forms the cluster when it has enough votes to establish quorum (that is, when enough voting computers have booted).

As the booting computer joins the cluster, the connection manager displays a message in the following format:

```
%CNXMAN, now an OpenVMS Cluster member -- system X...
```

Note that if quorum is lost while the computer is booting, or if a computer is unable to join the cluster within 2 minutes of booting, the connection manager displays a message like the following:

```
%CNXMAN, Discovered system X....  
%CNXMAN, Deleting CSB for system X....  
%CNXMAN, Established "connection" to quorum disk  
%CNXMAN, Have connection to system X...  
%CNXMAN, Have "connection" to quorum disk
```

The last two messages show any connections that have already been formed. If the cluster includes a quorum disk, you may also see a message like the following:

```
%CNXMAN, Using remote access method for quorum disk  
%CNXMAN, Using local access method for quorum disk
```

The first message indicates that the connection manager is unable to access the quorum disk directly, either because the disk is unavailable or because it is accessed through the MSCP server. Another computer in the cluster that can access the disk directly must verify that a reliable connection to the disk exists.

The second message indicates that the connection manager can access the quorum disk directly and can supply information about the status of the disk to computers that cannot access the disk directly.

Note that the connection manager may not see the quorum disk initially, because the disk may not yet be configured. In that case, the connection manager first uses remote access, then switches to local access.

After the computer has joined the cluster, normal startup procedures execute. One of the first functions is to start the OPCOM process.

---

## Node Boots

When a node boots:

- A satellite node displays a message that identifies the boot node.
- Both CI nodes and satellite nodes display the OpenVMS banner.
- If the computer is a satellite node, a message such as the following shows the name and LAN address of the MOP server that has downline loaded the satellite.  

```
%OpenVMS Cluster-I-SYSLOAD, system loaded from Node x... (xx-xx-xx-xx-xx-xx)
```
- At this point, the satellite has completed communication with the MOP server and further communication continues with the system disk server using OpenVMS Cluster system communications.
- For any booting computer, the OpenVMS "banner message" is displayed.

## Node Attempts to Form or Join Cluster

When a node attempts to form or join a cluster, the connection manager displays messages such as the following:

waiting to form or join an OpenVMS Cluster system

- If the computer is a member of a local area or mixed interconnect cluster, the cluster security database is loaded.
- Optionally, the MSCP and TMSCP servers are loaded.

If the node finds a cluster, the node attempts to join. The connection manager displays messages such as the following:

%CNXMAN, Sending OpenVMS Cluster membership request to system.

If no cluster is found, the connection manager forms a cluster when enough nodes have booted to establish a quorum. The connection manager displays messages such as the following:

%CNXMAN, Discovered system X...

%CNXMAN, Deleting CSB for system X...

%CNXMAN, Established "connection" to quorum disk

%CNXMAN, Have connection to system X...

%CNXMAN, Have "connection" to quorum disk



---

**Note**

When the cluster security database is loaded, the following message is displayed:

```
%OpenVMS Cluster-I-LOADSECDB, loading the cluster security database
```

When the MSCP server and TMSCP server are loaded, the following messages are displayed:

```
%MSCPLOAD-I-LOADMSCP, loading the MSCP disk server
```

```
%TMSCPLOAD-I-LOADTMSCP, loading the TMSCP tape server
```

If quorum is lost while the computer is booting, or if a computer is unable to join the cluster within 2 minutes of booting, the connection manager displays the same kind of message as it does when no cluster is found.

---

## Node Joins Cluster

As the booting computer joins the cluster, the connection manager displays a message such as the following:

```
%CNXMAN, now an OpenVMS Cluster member -- system name
```



### Note

After the computer has joined the cluster, normal startup procedures execute. One of the first functions is to start the OPCOM process.

The SYSINIT process:

- Initializes the cluster connection management.
- Creates the standalone configure process, STACONFIG, which autoconfigures disks and SCS communication ports.

If the system parameter DISK\_QUORUM indicates that there is to be a quorum disk, STACONFIG polls to discover the disk.

System parameter DISK\_QUORUM specifies the cluster quorum disk name.

SYSINIT prints the following message on the console terminal:

Waiting to form or join an OpenVMS Cluster system

If the cluster includes a quorum disk, you may also see message such as the following:

%CNXMAN, Using remote access method for quorum disk

%CNXMAN, Using local access method for quorum disk

- The first message indicates that the connection manager is unable to access the quorum disk directly, either because the disk is unavailable or because it is accessed through the MSCL server. Another computer in the cluster that can access the disk directly must verify that a reliable connection to the disk exists.
- The second message indicates that the connection manager can access the quorum disk directly and can supply information about the status of the disk to computers that cannot access the disk directly.

Note that the connection manager may not see the quorum disk initially, because the disk may not yet be configured. In that case, the connection manager first uses remote access, then switches to local access.

## CI Node Boot Problems

### Cluster-Related System Initialization Problems

If a CI node initialization fails to achieve the OpenVMS banner:

1. Check to see that the boot startup command procedure specifies:
  - a. The contents of register 0 (R0) to be 20 (indicates booting from CI cable).
  - b. R1 should be the internal bus computer number (if applicable) Multivendor Customer Services supplies this number.
  - c. R2 should be the port number on the Star coupler where the HSC is located, or the HSC node number.
  - d. R3 should contain a hexadecimal value, which is the HSC disk unit number from which the node is to boot.
  - e. R5 should contain the correct root number.
2. Use the HSC SHOW ALL command to check the status of the HSC unit.
  - a. Is it on line and operational?
  - b. Verify that all expected equipment is present and available.
  - c. If the node appears in the display as DISABLED, use the SETSHO utility to set the node to the ENABLED state.

### CI Node Fails to Join the Cluster

If the OpenVMS banner is displayed, but the node fails to join the cluster:

1. Check that OpenVMS Cluster system software has been loaded.
2. Check that the system parameter OpenVMS Cluster is 2.
3. Check for the existence of the cluster security database file,  
SYS\$COMMON:[SYSEXE]CLUSTER\_AUTHORIZE.DAT.  
  
If all nodes do not use the same security database, the contents of all copies of the database must be the same.
7. Check that the node booted from the correct disk and system root.
8. Check that SCSNODE and SCSSYSTEMID are unique.
9. Check voting or quorum issues.

## Satellite Node Initialization Problems

### Satellite Initialization Landmarks

When initializing a satellite, look for a load host message in addition to the other initialization landmarks, such as:

- Hardware diagnostics
- Boot prompt >>> (unless autostart)
- Load from host message
- OpenVMS banner
- Startup banner
- SYSTARTUP display



#### Note

To boot, a satellite node must be able to communicate with a boot server over the Ethernet.

- DECnet event logging can be used to verify this communication.
- Use NCP HELP DEFINE/SET/SHOW/CLEAR LOGGING for more information.

If you do not see the load host message, the satellite node is out there saying “Boot Me! Boot Me!,” but nobody is responding.

### Message from Load Host Not Displayed

If a satellite node fails to display the load host message or banner, check the following:

- Is the DECnet software on the host up and running? Enter the SHOW NETWORK command to check.
- Is the Ethernet boot request being sent and received?
- Is circuit service on the host enabled?
- Is the Ethernet hardware address on the host correct?
- Is the boot device available to the host?

### Is DECnet Software Running?

Check to make sure that the DECnet software is running on the load host by entering the DCL command SHOW NETWORK.

If you receive a message similar to the following on the host in response, the network software is running.

```
$ SHOW NETWORK
```

```
Product: DECNET   Node: VMSAFF   Address(es): 61.104
```

If the network is not running, you must start the network before you can initialize the satellite.

### Enabling Event Logging

If a satellite node fails to boot, log in as SYSTEM on the boot server and enter the following NCP commands to enable event logging:

```
NCP> SET LOGGING MONITOR EVENT 0.*
```

```
NCP> SET LOGGING MONITOR STATE ON
```

### Is the Ethernet Boot Request Received and Sent?

To check that Ethernet boot requests are received and sent:

1. Enter the following DCL command:

```
$ REPLY/ENABLE=NETWORK
```

2. Reboot the satellite node and look at the messages.

- a. If you get an appropriate MOP service request message similar to the following example, DECnet is not the problem.

```
DECnet event 0.3, automatic line service
From node 3.1 (WHYNOT), 15-DEC-1998 10:34:24.01
Circuit SVA-0, Load, Requested, Node = 3.12 (BECAUS)
File = SYS$SYSDEVICE:<SYS10.>, Operating system
Ethernet address = 08-00-2B-17-AC-04
```

- d. A message similar to the following indicates that the Ethernet address for the satellite node was entered incorrectly or not at all.

```
DECnet event 0.7, aborted service request
From node 3.1 (WHYNOT), 15-DEC-1998 11:07:21.18
Circuit SVA-0, Line open error,
Ethernet address = 08-00-2B-12-AC-04
```

Verify the load host network database.

- e. If no messages appear, there may be problems with physical cable connections or adapter service, or the satellite node is not generating boot requests.

## Verifying the Satellite Information in the Load Host DECnet Database

On the load host, enter the following command to display the satellite node's hardware address:

```
NCP> SHOW NODE satellite-name CHARACTERISTICS
```

If the satellite data is not present in the DECnet database:

1. Examine SYSS\$MANAGER:NETNODE\_UPDATE.COM for the satellite node's data.

If found, execute this procedure.

2. Run CLUSTER\_CONFIG and add the satellite node again.

If the data is in the DECnet database, compare the address displayed with the hardware address of the satellite. To obtain this information, follow the specific instructions for the satellite hardware.

If the Ethernet address is not correct in the load host database, correct it using the CLUSTER\_CONFIG procedure.



### Note

To get a satellite Ethernet address on the satellite:

Newer equipment >>> SHOW ETHERNET

>>> SHOW DEVICE

Older equipment >>> T 50

In a networked environment, you can also try booting with a tertiary boot file called READ\_ADDR in the following manner:

>>>BOOT/100

Enter file name: READ\_ADDR

Any machine with boot service enabled that also has the file READ\_ADDR.SYS in SYSS\$COMMON:[MOM\$SYSTEM] will return the Ethernet address of the satellite.

## Verifying the Circuit Service on the Load Host

Use the NCP SHOW CIRCUIT command to verify that the circuit service is enabled on the load host.

```
NCP> SHOW CIRCUIT CHARACTERISTICS circuit-id
```

To enable circuit service:

```
NCP> SET CIRCUIT circuit-id STATE OFF
```

```
NCP> DEFINE CIRCUIT circuit-id SERVICE ENABLED
```

```
NCP> SET CIRCUIT circuit-id SERVICE ENABLED STATE ON
```

## Handling OpenVMS Cluster System Hangs

### Overview

Certain conditions can cause an OpenVMS Cluster system to hang.

- Cluster quorum is lost.
- A shared cluster resource is inaccessible.

You must be able to identify an OpenVMS Cluster system hang and to take action to restore the cluster.

This section discusses recognizing and handling OpenVMS Cluster system hangs.

## Recognizing and Handling OpenVMS Cluster System Hangs

### Be Sure the OpenVMS Cluster System Is Hung

Heavy network traffic or a long cluster transition can make the OpenVMS Cluster system appear stopped.

Be patient; the cluster may not be hung at all.

### Cluster Quorum Is Lost

When cluster quorum is lost because a voting member exited:

- Process creation and I/O activity on all nodes ceases.
- Information about events that cause loss of quorum is sent to OPCOM and the console terminal at each node.
  - Quorum may be lost before OPCOM has a chance to display the message.
  - OPA0 is the most reliable source of information.

### Handling Loss of Quorum

There are two options for restoring cluster quorum.

- Lower quorum and reconfigure the cluster.
- Be sure you know why you are lowering quorum, if you choose to do so, and make sure that you do not set up a situation where the cluster can become partitioned.
- Have a node rejoin the cluster.



#### Note

The following is a trick that can help to restore quorum.

```
[Ctrl] /P
>>>H
>>>D/1 14 C
>>>CONTINUE
IPC> Q
IPC>[Ctrl] /z
```

Present this to the class with the following cautions:

- It does not always work.
- It does not work on certain processors such as the VAX 9000.
- What it does is to say “Trust the human” so the human better do it right.

## **Shared Resource Inaccessible**

When a process is granted a lock on a resource, other processes must wait to access it.

If the lock is held for an extended period of time, the cluster will appear to hang.

## **Conditions That Cause Shared Resource Problems**

The following conditions can cause the cluster to appear to hang because of shared resource problems:

- Volume rebuilding causes the volume being rebuilt to be locked.
- Locks on parts of SYS\$SYSTEM:SYSUAF.DAT for write access will cause processes attempting to log in to appear to hang.
  - Typically, locks such as this do not endure long enough to cause major problems.
  - If the process with the lock is unable to proceed, the backlog of blocked processes could proliferate rapidly.
  - The lock manager is functioning correctly in this case, so no error messages are generated.

## Case Studies

### Overview

This section contains two case studies that supplement the laboratory exercises. The case studies do not require you to use the laboratory systems.

The case studies are:

- OpenVMS Cluster system booting problem
- OpenVMS Cluster system communication problem



#### **Note**

The developer recommends that you discuss these case studies after the laboratory exercises for this module are finished.

You may want to take a course in conducting case studies. The level of student participation is a good measure of your success with this teaching technique. If the students are quietly listening, it is a lecture not a case study.

Business schools have made good use of the case study discussion technique. If you know of a professor who is proficient in leading case studies, try to sit in on a class. (Not all business school professors are good at presenting case studies.)

Questions are presented in the instructor notes to help you with the case study technique. It is very important to allow the students time to read each section before beginning the discussion.

---

## Case Study Problem: OpenVMS Cluster System Booting Problem

### Problem Statement

You are the system manager of an OpenVMS Cluster system consisting of two CI nodes (NODEA and NODEB), one satellite node (NODEC), and one HSC controller. After a power failure, serious problems occur when you try to reboot the cluster.

You begin to investigate the problem by gathering information from the individual systems.



---

#### Note

Start the discussion of this case study by drawing a diagram of the cluster on the board. Label each node. Do not tell the class how the system parameters or votes are set. Allow the students time to study each bullet.

- NODEA and NODEB are CI nodes.
  - NODEC is a satellite node.
  - NODEA has system parameter "OpenVMS Cluster=0".
  - NODEC will try and fail to boot from NODEA.
  - NODEA and NODEB have "VOTES=1" and the system disk is the "QUORUM\_DISK".
  - NODEB has system parameter "OpenVMS Cluster=2." NODEB will boot.
  - NODEB boots the satellite, if it is configured as a boot server.
  - NODEA cannot boot the satellite because it is not a cluster member.
- 

### Example: Console Errors

When you boot NODEA, the following errors are displayed on the console:

```
%MSCPLOAD-E-SERVE_ERR, error serving device $4$DUA6: (HSC001), R0  
status = 00000084
```

```
%MSCPLOAD-E-SERVE_ERR, error serving device $4$MUA1: (HSC001), R0  
status = 00000084
```



### Note

Suggested questions:

- What do the error messages displayed after NODEA is booted indicate?
- The disk and tape device cannot be MSCP served.
- How can you get more information on this error message?

```
$ HELP ERRORS SERVE_ERR
ERRORS
SERVE_ERR
error serving device DDcnn: R0 status = nnnnnnnn
```

**Facility:** STACONFIG, Standalone Configure Process

**Explanation:** A newly discovered device that fits all the serving parameters could not be served. If the value in the R0 status is 00000084, the device was not served because the server is not loaded.

**User Action:** Check to see that the MSCP server has been loaded properly. Use the SHOW DEV/SERVED command or examine SCS\$GL\_MSCP.

This global address should be a valid system space address. If the R0 status is other than 84, get the corresponding message (using EXIT nn) and find out what is wrong with the device.

## Example: SHOW CLUSTER /CONTINUOUS Output

The SHOW CLUSTER/CONTINUOUS command produces the following output on NODEA:

```
$ SHOW CLUSTER/CONTINUOUS
```

View of Cluster from system ID 123456 node: NODEA 15-DEC-1998 12:59:48

```
+-----+-----+
| SYSTEMS | MEMBERS |
+-----+-----+
|  NODE  | SOFTWARE | STATUS |
+-----+-----+
| NODEA  | VMS V7.2 |      |
| JUST   | HSC V400 |      |
| LEGAL  | HSC V400 |      |
+-----+-----+
```

**Note**

Suggested questions:

- What is the status of NODEA? What status could be displayed here?  
Member
  - Is NODEA a cluster member?  
No.
- 

**Example: SHOW SYSTEM Output**

The SHOW SYSTEM command produces the following output on NODEA:

\$ SHOW SYSTEM

OpenVMS V7.0 on node DBLDOG 15-DEC-1998 13:08:26.15 Uptime 34 23:03:54

| Pid      | Process Name | State | Pri | I/O      | CPU         | Page | flts  | Pages |
|----------|--------------|-------|-----|----------|-------------|------|-------|-------|
| 00000201 | SWAPPER      | HIB   | 16  | 0 0      | 00:00:08.99 |      | 0     | 0     |
| 00000205 | CONFIGURE    | HIB   | 8   | 5 0      | 00:00:00.08 |      | 114   | 179   |
| 00000207 | IPCACP       | HIB   | 10  | 7 0      | 00:00:00.08 |      | 95    | 183   |
| 00000208 | ERRFMT       | HIB   | 8   | 21283 0  | 00:01:21.38 |      | 151   | 239   |
| 00000209 | OPCOM        | HIB   | 8   | 259856 0 | 00:16:25.40 |      | 925   | 183   |
| 0000020A | AUDIT_SERVER | HIB   | 10  | 284 0    | 00:00:01.50 |      | 495   | 867   |
| 0000020B | JOB_CONTROL  | HIB   | 10  | 752 0    | 00:00:09.73 |      | 293   | 427   |
| 00000210 | NETACP       | HIB   | 10  | 125553 1 | 22:13:38.64 |      | 9008  | 9024  |
| 00000211 | EVL          | HIB   | 6   | 332255 0 | 00:08:15.22 |      | 316   | 593 N |
| 00000212 | REMACP       | HIB   | 8   | 270 0    | 00:00:00.49 |      | 107   | 87    |
| 00000B47 | SYSTEM       | CUR   | 4   | 1969 0   | 00:00:42.75 |      | 12753 | 343   |

**Note**

Suggested questions:

- How far has the initialization of NODEA gone? What processes are running?
  - What processes are missing from NODEA?
-

## Example: Satellite Boot Errors

When you boot the satellite node, NODEC, this error is displayed:

2. .

?54 Retry

At the same time, on NODEA, an error similar to the following is displayed at the console:

```
%%%%%%%%%% OPCOM 15-DEC-1998 09:48:33.3 %%%%%%%%%%
```

```
Message from user DECNET on NODEA
```

```
DECnet event 0-.7, aborted service request
```

```
From node 1.9 (NODEA), 15-DEC-1998 09:48:33.3
```

```
Circuit BNA-0, Line open error, File open error, Load file
```

```
%LAA-F-LCLNOTCLS, local node is not a VAXcluster member
```

```
Node = 1.15 (NODEC), Ethernet address = 08-00-2B-15-23-A9
```



### Note

Suggested question:

- What do these error messages indicate?  
NODEC cannot be loaded from NODEA because NODEA is not a cluster member.

**Example: SHOW CLUSTER /CONTINUOUS Output**

After booting NODEB, compare the display from SHOW CLUSTER/CONTINUOUS on NODEB to that on NODEA.

```
$ SHOW CLUSTER/CONTINUOUS
```

```
View of Cluster from system ID 123457 node: NODEB 15-DEC-1998 09:52:26
```

```
+-----+-----+
|  SYSTEMS      | MEMBERS |
+-----+-----+
|  NODE | SOFTWARE | STATUS |
+-----+-----+
| NODEB | VMS V7.0 | MEMBER |
| JUST  | HSC V400 |        |
| NODEA | VMS V7.0 |        |
| LEGAL | HSC V400 |        |
+-----+-----+
```

```
$ SHOW CLUSTER/CONTINUOUS
```

```
View of Cluster from system ID 123456 node: NODEA 15-DEC-1998 09:53:40
```

```
+-----+-----+
|  SYSTEMS      | MEMBERS |
+-----+-----+
|  NODE | SOFTWARE | STATUS |
+-----+-----+
| NODEA | VMS V7.0 |        |
| JUST  | HSC V400 |        |
| LEGAL | HSC V400 |        |
+-----+-----+
```



---

**Note**

Suggested questions:

- What is the status of NODEA?  
NODEA is not a member of the cluster.
  - What is the status of NODEB?  
NODEB is a member of the cluster.
-

## Example: SHOW SYSTEM Output

Issue the SHOW SYSTEM command on both NODEA and NODEB. Compare the display on each node.

SHOW SYSTEM on NODEB:

\$ SHOW SYSTEM

OpenVMS V7.2 on node NODEB 15-DEC-1998 14:01:02.03 Uptime 0 00:02:45

| Pid      | Process Name   | State | Pri | I/O   | CPU         | Page | flts | Ph.Mem |
|----------|----------------|-------|-----|-------|-------------|------|------|--------|
| 22600301 | SWAPPER        | HIB   | 16  | 0 0   | 00:00:00.30 | 0    | 0    |        |
| 22600306 | CONFIGURE      | HIB   | 10  | 23 0  | 00:00:00.12 | 112  | 195  |        |
| 22600307 | IPCACP         | HIB   | 10  | 7 0   | 00:00:00.08 | 102  | 169  |        |
| 22600309 | ERRFMT         | HIB   | 8   | 16 0  | 00:00:00.12 | 84   | 144  |        |
| 2260030A | CACHE_SERVER   | HIB   | 16  | 6 0   | 00:00:00.05 | 63   | 118  |        |
| 2260030B | CLUSTER_SERVER | HIB   | 10  | 10 0  | 00:00:00.11 | 127  | 284  |        |
| 2260030C | OPCOM          | HIB   | 8   | 39 0  | 00:00:00.26 | 185  | 223  |        |
| 2260030D | AUDIT_SERVER   | HIB   | 10  | 33 0  | 00:00:00.42 | 1347 | 430  |        |
| 2260030E | JOB_CONTROL    | HIB   | 10  | 35 0  | 00:00:00.17 | 152  | 279  |        |
| 2260030F | SMISERVER      | HIB   | 9   | 32 0  | 00:00:00.34 | 280  | 510  |        |
| 22600311 | NETACP         | HIB   | 10  | 39 0  | 00:00:00.40 | 185  | 399  |        |
| 22600312 | EVL            | HIB   | 5   | 54 0  | 00:00:00.33 | 278  | 63   | N      |
| 22600313 | REMACP         | HIB   | 9   | 8 0   | 00:00:00.06 | 65   | 58   |        |
| 22600315 | SYSTEM         | CUR   | 14  | 228 0 | 00:00:02.91 | 898  | 319  |        |

SHOW SYSTEM on NODEA:

\$ SHOW SYSTEM

OpenVMS V7.2 on node NODEA 15-DEC-1998 14:28:08.49 Uptime 0 01:05:15

| Pid      | Process Name | State | Pri | I/O   | CPU         | Page | flts | Ph.Mem |
|----------|--------------|-------|-----|-------|-------------|------|------|--------|
| 22400201 | SWAPPER      | HIB   | 16  | 0 0   | 00:00:00.26 | 0    | 0    |        |
| 22400206 | CONFIGURE    | HIB   | 10  | 32 0  | 00:00:00.10 | 112  | 195  |        |
| 22400207 | IPCACP       | HIB   | 10  | 7 0   | 00:00:00.06 | 102  | 169  |        |
| 22400209 | ERRFMT       | HIB   | 8   | 46 0  | 00:00:00.10 | 84   | 144  |        |
| 2240020C | OPCOM        | HIB   | 8   | 61 0  | 00:00:00.25 | 303  | 177  |        |
| 2240020D | AUDIT_SERVER | HIB   | 10  | 42 0  | 00:00:00.48 | 1355 | 438  |        |
| 2240020E | JOB_CONTROL  | HIB   | 10  | 45 0  | 00:00:00.26 | 153  | 280  |        |
| 22400211 | NETACP       | HIB   | 10  | 94 0  | 00:03:16.13 | 8486 | 8374 |        |
| 22400212 | EVL          | HIB   | 6   | 218 0 | 00:00:00.75 | 2700 | 75   | N      |
| 22400213 | REMACP       | HIB   | 8   | 8 0   | 00:00:00.04 | 65   | 58   |        |
| 22400218 | SYSTEM       | CUR   | 14  | 246 0 | 00:00:03.04 | 1225 | 293  |        |



---

**Note**

Suggested question:

- What processes do you see on NODEB that you do not see on NODEA?
-

### Example: Displaying VAXCLUSTER Parameter

Use the System Generation utility (SYSGEN) to check the system parameter VAXCLUSTER on NODEA.

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

```
SYSGEN> SHOW VAXCLUSTER
```

| Parameter Name | Current | Default | Min. | Max. | Unit      | Dynamic |
|----------------|---------|---------|------|------|-----------|---------|
| VAXCLUSTER     | 0       | 1       | 0    | 2    | Coded-val |         |

```
SYSGEN>
```



#### Note

Suggested question:

- Why didn't NODEA join the cluster?  
VAXCLUSTER = 0

## Case Study Solution: OpenVMS Cluster System Booting Problem

### Solution

NODEA is unable to boot the satellite node NODEC because NODEA is not a cluster member.

There are two ways you can correct this situation:

- If NODEA is not to be restored as a cluster member, configure NODEB as a boot server.
- Restore NODEA as a cluster member.

### Configuring a Boot Server

Use the command procedure SYS\$MANAGER:CLUSTER\_CONFIG.COM to enable NODEA as a boot server.

### Restoring a Cluster Member

To restore NODEA as a cluster member:

1. On NODEA, use SYS\$MANAGER:CLUSTER\_CONFIG.COM to reset the following:
  - NODENAME:
  - NETWORK ADDRESS:
  - ENABLE ETHERNET COMMUNICATIONS?
  - CLUSTER GROUP NUMBER:
  - QUORUM DISK (Y/N)?
  - ALLOCLASS:
10. The system will execute AUTOGEN GETDATA REBOOT. When it has rebooted, you can check your success.

11. On NODEA, enter SHOW CLUSTER and check the member status.

```
$ SHOW CLUSTER
```

```
View of Cluster from system ID 123456 node: NODEA 15-DEC-1998
09:53:44
```

```
+-----+-----+
|  SYSTEMS      | MEMBERS |
+-----+-----+
|  NODE   | SOFTWARE | STATUS |
+-----+-----+
|  NODEA  | VMS V7.0 | MEMBER |
|  JUST   | HSC V400 |        |
|  NODEC  | VMS V7.0 | MEMBER |
|  LEGAL  | HSC V400 |        |
|  NODEB  | VMS V7.0 | MEMBER |
+-----+-----+
```

12. On NODEA, run SYSGEN, and enter SHOW VAXCLUSTER.

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

```
SYSGEN>SHOW VAXCLUSTER
```

| Parameter Name | Current | Default | Min. | Max. | Unit       | Dynamic |
|----------------|---------|---------|------|------|------------|---------|
| VAXCLUSTER     | 2       | 1       | 0    | 2    | Coded-valu |         |

```
SYSGEN>EXIT
```

## Case Study Problem: OpenVMS Cluster System Communication Problem

### Problem Statement

You are the system and network manager for an OpenVMS Cluster system with two CI nodes.

You observe a series of error messages on the operator console and must determine their cause.



---

**Note**

This exercise is meant to be done with only the text material provided. No lab equipment is required.

Start this discussion by drawing a diagram of the cluster on the board. Fill in the information you know. As you learn more, add the information to the diagram.

---

### Example: Console Errors

The following errors are displayed on the operator console:

```
%CNXMAN, Lost connection to system NODEB
```

```
%PAA0, Software is Closing Virtual Circuit - REMOTE NODE NODEB
```

```
%PAA0, Path #0. Has gone from GOOD to BAD - REMOTE NODE HSC000
```

```
%PAA0, Path #0. Has gone from GOOD to BAD - REMOTE NODE HSC000
```

```
%CNXMAN, Re-established connection to system NODEB
```

```
$4$dua20: (HSC000) OnLine
```

```
%%%%%%%%%% OPCOM 15-DEC-1998 15:56:29.86 %%%%%%%%%%%
```

```
15:56:27:68 Node NODEA (CSID 00010006) lost connection to node NODEB
```

```
$4$dua26: (HSC000) OnLine
```

```
%%%%%%%%%% OPCOM 15-DEC-1998 15:56:29.87 %%%%%%%%%%%
```

```
15:56:28:27 Node NODEA (CSID 00010006) re-established connection to  
node NODEB
```

```
$4$dua27: (HSC000) OnLine
```

```
%%%%%%%%%% OPCOM 15-DEC-1998 15:56:29.99 %%%%%%%%%%% from node NODEB  
at
```

15-DEC-1998 15:56:28.95)

15:56:28:94 Node NODEB (CSID 00010001) lost connection to node NODEA

\$4\$dua101: (HSC000) OnLine

%%%%%%%% OPCOM 15-DEC-1998 15:56:29.99 %%%%%%%%% from node NODEB  
at

15-DEC-1998 15:56:29.95)

15:56:28:88 Node NODEB (CSID 00010001) Re-established connection to  
node NODEA

%PAA0, Path #1. Loopback has gone from GOOD to BAD

%PAA0, Path #0. Loopback has gone from GOOD to BAD

## Example: SHOW CLUSTER /CONTINUOUS Output

The following is a view of the cluster from the SHOW CLUSTER/CONTINUOUS command:

```
$ SHOW CLUSTER/CONTINUOUS
```

```
View of Cluster from system ID 123456 node: NODEA 15-DEC-1998 16:23:44
```

```
+-----+-----+-----+
| SYSTEMS | MEMBERS | CIRCUIT |
+-----+-----+-----+
| NODE   | SOFTWARE | STATUS | CABLE |
+-----+-----+-----+
| NODEA | VMS V7.0 | MEMBER |      |
| JUST  | HSC V400 |      |      |
| LEGAL | HSC V400 |      |      |
| NODEB | VMS V7.0 | MEMBER |      |
+-----+-----+-----+
|
```

## Case Study Solution: OpenVMS Cluster System Communication Problem

### Solution

The errors occurred because the CI cable to the HSC controller was disconnected.

After the CI cable is reconnected, the following messages appear:

```
%PAA0, Path #0. Loopback has gone from BAD to GOOD
```

```
%CNXMAN, RE-established connection to system NODEB
```

```
%%%%%%%%%% OPCOM 15-DEC-1998 15:57:27.79 %%%%%%%%%%
```

```
15:57:27:74 Node NODEA (csid 00010006) re-established connection to
node NODEB
```

```
$
```

## Summary

### Concepts

#### OpenVMS Cluster System Concepts

- OpenVMS Cluster system software components include system communications services, VAXport drivers, connection manager, distributed file system, distributed lock manager, distributed job controller, MSCP server, and the TMSCP server.
- OpenVMS Cluster system diagnostic messages important to troubleshooting OpenVMS Cluster system problems include connection manager messages, VAXport error messages, HSC error messages and fault conditions, error log entries, and DECnet event messages.
- Use the HSC SETSHO utility to diagnose cluster problems and to set parameters.
- Use the Network Control Program (NCP) to troubleshoot OpenVMS Cluster system problems.
- The connection manager initiates CLUEXIT bugchecks.

#### Failure of Nodes to Boot or Join the Cluster

- The connection manager on each cluster node is responsible for forming or joining the node to the OpenVMS Cluster system.
- When a CI node fails to boot or join the cluster, the problem may be with the node or with cluster components.
- When a satellite node fails to boot or join the cluster, you can use the boot node to research the problem.

#### Handling OpenVMS Cluster System Hangs

- Cluster state transition and inaccessible resources can cause the OpenVMS Cluster system to appear to hang.

## Chapter Terms

**bugcheck**

A system crash caused by a severe internal software error.

**quorum**

Voting system that prevents a cluster from partitioning.

**VAXCLUSTER**

System parameter used to determine whether or not a booting node will attempt to form an OpenVMS Cluster system.



---

## ***Learning Check***

## Questions

1. In the space provided, number each of the following system initialization events or landmarks to indicate the order in which they occur during a conversational boot.
  - a. VMB or APB runs .....
  - b. SYSTARTUP\_VMS.COM runs .....
  - c. EXE\$INT is called .....
  - d. STARTUP.COM runs .....
  - e. SYSBOOT prompt appears .....
  - f. SYSINIT runs .....
  - g. EXE\$SWAPINIT runs .....

In the space provided, write the letter corresponding with the best answer to each of the following multiple-choice questions.

2. Testing a printer by despooling and using the COPY command to print a file on the printer is an attempt to:
  - a. Isolate the problem
  - b. Verify the solution
  - c. Propose a solution
  - d. Document the problem
3. If a user reports that his file did not print, which question will \_\_\_\_\_ give you the most useful information regarding the problem?
  - a. What kind of file were you printing?
  - b. When did you print the file?
  - c. What command did you use to print the file?
  - d. What privileges do you have?
4. Which of the following *cannot* be done during a conversational boot at the SYSBOOT> prompt?
  - a. Specify another parameter file be used.
  - b. Change system parameters.
  - c. Edit the site-independent startup command file.
  - d. Display system parameters.

5. A minimum system initialization:
  - a. Ignores all errors in the startup command procedure
  - b. Does not invoke the site-specific startup command procedure
  - c. Sets the number of interactive users to 1
  - d. All of the above
6. Which processes are active after a minimum system initialization?
  - a. SWAPPER
  - b. IPCACP
  - c. ERRFMT
  - d. All of the above
7. Before you can use the full Backup utility when the system is initialized with the console as the startup command procedure, you must:
  - a. Install the required images.
  - b. Set the system parameter STARTUP to 1.
  - c. Delete the swapper process.
  - d. Define a symbol.
8. Which procedures can you use to boot the system when the site-independent startup file has been corrupted?
  - a. Run AUTOGEN.
  - b. Use an alternate site-independent startup file on the system disk.
  - c. Boot minimum.
  - d. Run SYSGEN.
9. If the system disk is broken, you can initialize the system by:
  - a. Running the Convert utility to make another disk the system disk
  - b. Using standalone Backup to restore a copy of the system disk to a spare disk
  - c. Using the console disk
  - d. None of the above
10. If the file SYSDUMP.DMP has been deleted from the system disk before the system crashes, the system dump file will be:
  - a. Lost
  - b. Written to a new file on another disk
  - c. Written to the system paging file
  - d. Written to a newly created SYSDUMP.DMP file on the system disk

11. Which of the following are written to the system dump file when the operating system fails?
  - a. Processor registers
  - b. System parameters
  - c. Operator log file
  - d. Cause of the crash
12. A nonfatal bugcheck:
  - a. Copies the contents of memory to the system dump file
  - b. Causes the system to crash
  - c. Results in an error log entry
  - d. Aborts normal system operation
13. The required size of the system dump file is determined by:
  - a. Size of memory
  - b. System parameter ERRORLOGBUFFERS
  - c. System parameter ERLBUFFERPAGES
  - d. All of the above
14. Which of the following is *not* a function of the ERRFMT process?
  - a. Reports the contents of the error log file
  - b. Formats standard error messages
  - c. Writes error messages regarding device errors to the error log buffers
  - d. Writes the error log buffers to a file on disk
15. If the node on which the queue manager process is running leaves the cluster:
  - a. All batch queues stop, but output queues continue if they are AUTOSTART.
  - b. The queue manager process fails over to another node.
  - c. All AUTOSTART queues stop.
  - d. All batch and print queues stop.
16. Which statement regarding the queue database files is true?
  - a. The queue file and the journal file must be located on the same disk.
  - b. All three files must be located on the system disk.
  - c. The three files can be located on three separate disks as long as the master file is on the system disk.
  - d. All three files must be located on the same disk.

17. Which statement regarding the connection manager in an OpenVMS cluster system is true?
  - a. The connection manager prevents cluster partitioning using the quorum scheme.
  - b. The connection manager controls the integrity of the cluster.
  - c. The connection manager uses SCS to provide an acknowledged message delivery service for the MSCP server.
  - d. All of the above.
18. If connection manager messages do not appear when the cluster systems are booting, you should suspect:
  - a. The console is turned off.
  - b. A cluster quorum does not exist.
  - c. The OpenVMS cluster system software was not loaded correctly.
  - d. All of the above.
19. Which NCP command can you use to display information about the state of a circuit?
  - a. SHOW KNOWN CIRCUIT CHARACTERISTICS
  - b. SHOW KNOWN LINE CHARACTERISTICS
  - c. SHOW KNOWN NODE
  - d. None of the above
20. Bugchecks initiated by the connection manager are called:
  - a. Nonfatal bugchecks
  - b. Fatal bugchecks
  - c. Cluster bugchecks
  - d. CLUEXIT bugchecks



---

## ***Learning Check Answers***

## Answers

1. In the space provided, number each of the following system initialization events or landmarks to indicate the order in which they occur during a conversational boot

- |    |                        |   |
|----|------------------------|---|
| a. | VMB or APB runs        | 1 |
| b. | SYSTARTUP_VMS.COM runs | 7 |
| c. | EXE\$INT is called     | 3 |
| d. | STARTUP.COM runs       | 6 |
| e. | SYSBOOT prompt appears | 2 |
| f. | SYSINIT runs           | 5 |
| g. | EXE\$SWAPINIT runs     | 4 |

In the space provided, write the letter corresponding with the best answer to each of the following multiple-choice questions.

2. Testing a printer by despooling and using the COPY command to print a file on the printer is an attempt to:
- Isolate the problem
  - Verify the solution
  - Propose a solution
  - Document the problem
3. If a user reports that his file did not print, which question will \_\_\_\_\_ give you the most useful information regarding the problem?
- What kind of file were you printing?
  - When did you print the file?
  - What command did you use to print the file?
  - What privileges do you have?
4. Which of the following *cannot* be done during a conversational boot at the SYSBOOT> prompt?
- Specify another parameter file be used.
  - Change system parameters.
  - Edit the site-independent startup command file.
  - Display system parameters.

5. A minimum system initialization:
  - a. Ignores all errors in the startup command procedure
  - b. Does not invoke the site-specific startup command procedure
  - c. Sets the number of interactive users to 1
  - d. All of the above
6. Which processes are active after a minimum system initialization?
  - a. SWAPPER
  - b. IPCACP
  - c. ERRFMT
  - d. All of the above
7. Before you can use the full Backup utility when the system is initialized with the console as the startup command procedure, you must:
  - a. Install the required images.
  - b. Set the system parameter STARTUP to 1.
  - c. Delete the swapper process.
  - d. Define a symbol.
8. Which procedures can you use to boot the system when the site-independent startup file has been corrupted?
  - a. Run AUTOGEN.
  - b. Use an alternate site-independent startup file on the system disk.
  - c. Boot minimum.
  - d. Run SYSGEN.
9. If the system disk is broken, you can initialize the system by:
  - a. Running the Convert utility to make another disk the system disk
  - b. Using standalone Backup to restore a copy of the system disk to a spare disk
  - c. Using the console disk
  - d. None of the above
10. If the file SYSDUMP.DMP has been deleted from the system disk before the system crashes, the system dump file will be:
  - a. Lost
  - b. Written to a new file on another disk
  - c. Written to the system paging file
  - d. Written to a newly created SYSDUMP.DMP file on the system disk

11. Which of the following are written to the system dump file when the operating system fails?
  - a. Processor registers
  - b. System parameters
  - c. Operator log file
  - d. Cause of the crash
12. A nonfatal bugcheck:
  - a. Copies the contents of memory to the system dump file
  - b. Causes the system to crash
  - c. Results in an error log entry
  - d. Aborts normal system operation
13. The required size of the system dump file is determined by:
  - a. Size of memory
  - b. System parameter ERRORLOGBUFFERS
  - c. System parameter ERLBUFFERPAGES
  - d. All of the above
14. Which of the following is *not* a function of the ERRFMT process?
  - a. Reports the contents of the error log file
  - b. Formats standard error messages
  - c. Writes error messages regarding device errors to the error log buffers
  - d. Writes the error log buffers to a file on disk
15. If the node on which the queue manager process is running leaves the cluster:
  - a. All batch queues stop, but output queues continue if they are AUTOSTART.
  - b. The queue manager process fails over to another node.
  - c. All AUTOSTART queues stop.
  - d. All batch and print queues stop.
16. Which statement regarding the queue database files is true?
  - a. The queue file and the journal file must be located on the same disk.
  - b. All three files must be located on the system disk.
  - c. The three files can be located on three separate disks as long as the master file is on the system disk.
  - d. All three files must be located on the same disk.

17. Which statement regarding the connection manager in an OpenVMS cluster system is true?
  - a. The connection manager prevents cluster partitioning using the quorum scheme.
  - b. The connection manager controls the integrity of the cluster.
  - c. The connection manager uses SCS to provide an acknowledged message delivery service for the MSCP server.
  - d. All of the above.
18. If connection manager messages do not appear when the cluster systems are booting, you should suspect:
  - a. The console is turned off.
  - b. A cluster quorum does not exist.
  - c. The OpenVMS cluster system software was not loaded correctly.
  - d. All of the above.
19. Which NCP command can you use to display information about the state of a circuit?
  - a. SHOW KNOWN CIRCUIT CHARACTERISTICS
  - b. SHOW KNOWN LINE CHARACTERISTICS
  - c. SHOW KNOWN NODE
  - d. None of the above
20. Bugchecks initiated by the connection manager are called:
  - a. Nonfatal bugchecks
  - b. Fatal bugchecks
  - c. Cluster bugchecks
  - d. CLUEXIT bugchecks



---

# ***Setting Up a Remote Print Queue on a Terminal Server***

## ***Appendix A***

## Overview

The correct way to create a remote print queue on a terminal server is to follow a four-step process.

1. Set up the terminal server.
2. Set up the printer port on the server.
3. Set up the application port on the OpenVMS system.
4. Set up the print queue.



---

### Note

Setting up the terminal server is covered in the *OpenVMS System and Network Node Management* course, however, you might want to hand out a review sheet for the particular terminal server you are using in the laboratory exercises.

It is very important for every step to discuss how the data is made permanent on the system or how it survives system reboot.

- Does it go in the queue database?
  - Is it specified in a system startup file?
  - Does it go into a terminal server permanent database?
-

## Obtaining Printer Information

### Printer Test Sheet

Obtain a test sheet from the printer and verify the device setting for:

- Character size
- Parity
- Flow control
- Baud rate

## Step 1: Set Up the Terminal Server

### Specify a Terminal Server Name

Follow the specific instructions for installing and initializing the terminal server.

Assign a name to the terminal server.

- The name must be unique on the Ethernet.
- The name can be up to six characters long.
- The name must be written to both the permanent and volatile server database.

To set the server name, use local mode.

Format:

```
Local> SET SERVER NAME name
```

For example, to set the server name to ZK1123:

```
Local> SET PRIV
```

```
Local> SET SERVER NAME ZK1123
```

Verify the server name.

```
Local> SHOW SERVER
```



---

#### Note

Because each terminal server has a different operating system, the commands to set up the terminal server vary from one product to another. You should see that the documentation for the terminal server to be used in the laboratory exercises is available to the students.

On low-end servers the server name command is DEFINE after which the server must be initialized or the SET command used to write the name to the volatile database.

```
Local> DEFINE SERVER NAME name  
Local> SET SERVER NAME name
```

The DECserver 500 has three databases, not two.

The Terminal Server Configurator (TSC) utility is a software tool that resides on the load host and is used to modify the DECserver 500 permanent database. It gives a TSC> prompt. Because this is specific to the DECserver 500, it is not discussed here.

---

---

## Step 2: Set Up the Printer Port on the Server

### Display Port Characteristics

Use the terminal server in local mode to display the characteristics of the port to which the printer is to be connected.

```
Local> SHOW PORT 2 CHARACTERISTICS
```

Port 2:

```
Character Size:      8   Input Speed:    4800
Flow Control:      XON   Output Speed:  4800
Parity:            None   Modem Control: Disabled
```

```
Access:           Remote   Local Switch:  None
Backwards Switch: None   Name:         LN03_4
Break:           Disabled   Session Limit: 1
Forwards Switch:  None     Type:         Hard
```

Preferred Service: None

Authorized Groups: 0  
(Current) Groups: 0

Enabled Characteristics:

Autoconnect, Autoprompt, Inactivity Logout, Internet Connections,  
Line Editor, Password



#### Note

How is this made permanent?

It is stored in the permanent database of the server.

In this example, terminal servers are named with the first two letters of the building code and a number. Terminal server ZK1223. You might want to discuss how a database could be built to tie the terminal server number to a pole location.

---

## Set the Printer Port

In local mode on the terminal server, use the commands appropriate to the specific server to set up the printer port.

1. Give the port an appropriate name. For example, to assign the name LN03\_4 to port number 2:
2. Modify the port characteristics. Disable all enabled characteristics on the high-end servers and all enabled characteristics on low-end servers except for INPUT FLOW CONTROL and OUTPUT FLOW CONTROL.

```
Local> SET|DEFINE PORT 2 NAME LN03_4
```

```
Local> SET|DEFINE PORT 2 SPEED 4800
```

```
Local> SET|DEFINE PORT 2 AUTOBAUD DISABLE
```

```
Local> SET|DEFINE PORT 2 BROADCAST DISABLE
```

3. Make the changes permanent using the terminal server specific method.

4. Verify the port setting and the server name.

```
Local> SET PRIV
```

```
Local> SHOW SERVER
```

```
Local> SHOW PORT 2
```

5. Test the port. You should see a consistent test pattern.

```
Local> TEST PORT 2 COUNT 20 WIDTH 80
```



---

### Note

The expression SET|DEFINE indicates that on some servers the command is SET and on others DEFINE. For example, the following specifies the port access characteristic as REMOTE and to disable the characteristic AUTOBAUD where n= port-number on the server.

```
Local> SET|DEFINE PORT n ACCESS REMOTE AUTOBAUD DISABLE
```

How are these definitions made permanent?

They are stored in the server permanent database. On the low-end server, this happens when you log out of the port. On the high-end server it happens with the SAVE PORT command. SAVE PORT will only work on the DECserver 500 servers.

To continue to follow this example, you need to keep the following in mind. You might want to write these on the board.

Server name: ZK1123

Port name: N03\_4

---

## Example: Low-End Server

The following example sets up the server port 2 for a 4800 baud printer on a low-end server such as the DECserver 200.

```
Local> DEFINE PORT 2 NAME LN03_4
Local> DEFINE PORT 2 ACCESS REMOTE
Local> DEFINE PORT 2 AUTOBAUD DISABLED
Local> DEFINE PORT 2 SPEED 4800
Local> DEFINE PORT 2 PREFERRED NONE DEDICATED NONE
Local> LOGOUT PORT
```

## Example: High-End Server

The following example sets up the server port 2 for a 4800 baud printer on a high-end server such as the DECserver 500.

```
Local> SET PORT 2 NAME LN03_4
Local> SET PORT 2 ACCESS REMOTE
Local> SET PORT 2 AUTOBAUD DISABLED
Local> SET PORT 2 SPEED 4800
Local> SET PORT 2 PREFERRED NONE DEDICATED NONE
Local> SAVE PORT 2
Local> LOGOUT PORT 2
```



---

**Note**

Use the Terminal Server Configurator on the DECserver 500 system:

```
TSC>DEFINE PORT 2 SPEED 4800
```

---

## Step 3: Set Up the Application Port on the OpenVMS System

### Application Port

You must create and map an application port on each node that will host the remote print queue every time the system is initialized.

The computers that will access the printer must all be located on the same OpenVMS Cluster system.

Application ports are called LTA####, where #### is a number from 1 to 9999.

Each application port name must be unique.

- Design a numbering system to keep the port names unique.
- Before assigning a port name, check to see if it already exists.



---

**Note**

You can make this permanent through the STARTUP.COM file.

---

### Setting Up the Application Port

Perform the following steps on the node that will run the print queue:

1. Run LATCP.
2. Create the port:  

```
LATCP> CREATE PORT LTA####
```
3. Associate (or map) the application port with the terminal server port.  

```
LATCP> SET PORT LTA####:/APPLICATION/NODE=terminal-server-name -  
_LATCP> /PORT=port-name
```
4. Make the changes permanent by adding both commands to the LAT startup command procedure.
  - The LAT startup command procedure should be called from SYSTARTUP\_VMS.COM.
  - The call to invoke the LAT startup command procedure should be after the DECnet network is up and running.

**Note****LAT Control Program command: CREATE PORT**

Creates a logical port on your local OpenVMS node that connects with a remote device on a terminal server. Alternatively, this command creates a logical port on your local OpenVMS node that connects with a specific service. The service can be offered by a terminal server or associated with one or more dedicated ports on a remote LAT service node.

**Format: LATCP> CREATE PORT [port-name]**

Port-name specifies the port name in the form LTAn:, where n is a unique number from 1 to 9999. If the port you specify already exists, the LATCP returns the following error message:

```
%LAT-W-CMDERROR, error reported by command executor -SYSTEM-F-
DUPLNAM, duplicate name
```

The port name must be unique on the system. Two systems on the same cluster can have the same application port name and you can map the same port (identified by server name + port name ) to application ports with the different names on different nodes.

**/APPLICATION –**

Specifies that a logical port on your OpenVMS node is to be an application port. It can be used to connect to a remote printer on a terminal server. If you do not specify /APPLICATION or /DEDICATED, /APPLICATION is the default.

**LAT Control Program command: SET PORT**

Associates a logical port on the local OpenVMS node with a remote port on a terminal server that supports a device. Alternatively, it associates a logical port on the local OpenVMS node with a specific service. The service can be offered by a terminal server or associated with one or more dedicated ports on a remote LAT service node.

Format: LATCP> SET PORT port-name

Qualifiers:

/APPLICATION (same as above)

/NODE= remote-node-name

Specifies the name of a terminal server to be logically associated with the specified application port on your OpenVMS node. The server supports a remote device. In our example, the server name was ZK1123.

/PORT=remote-port-name

Specifies the name of the remote port on a terminal server that supports a remote device.

## Example: Create and Map an Application Port

The following example creates an application port name LTA8000 and maps the application port to the port LN03\_4 on terminal server ZK1123.

```
LATCP> CREATE PORT LTA8000: /APPLICATION
```

```
LATCP> SET PORT LTA8000:/APPLICATION/NODE=ZK1123/PORT=LN03_4
```

## Verify the Port Mapping

To verify the mapping of the application port to the terminal server port, enter the following command on the node that is to host the remote print queue:

```
$ COPY/LOG filename LTA####:
```

```
$ COPY/LOG filename LTA8000:
```



---

### Note

You can use the COPY command to test the printer because the printer has not been set spooled yet. After the device is set spooled, this command will not work.

To test an LN03 printer, you must use a file that contains at least 66 lines.

---

## Step 4: Set Up the Print Queue

### Special Steps for Remote Print Queues

Initialize the remote print queue using the same procedures as for regular print queues, with some important differences.

1. Set the printer characteristics.
2. Set protection on the device to allow only the system to access the printer.
3. Set the device spooled.
4. Define forms.
5. Initialize the queue.
6. Start the queue.
7. Test the queue.



#### Note

#### DCL command INITIALIZE/QUEUE

Qualifiers:

`/AUTOSTART_ON =(node::[device][,...])`

Designates an execution queue as an autostart queue and specifies the node, or node and device, on which the queue can be located. Both node and device must be specified for output queues.

When you enter the INITIALIZE/QUEUE command with the `/AUTOSTART_ON` qualifier, you must initially activate the queue for autostart, either by specifying the `/START` qualifier with the INITIALIZE/QUEUE command or by entering a START/QUEUE command. However, the queue will not begin processing jobs until the ENABLE AUTOSTART/QUEUES command is entered for a node on which the queue can run. If the ENABLE command has already executed on the node, the queue will start immediately.

This qualifier cannot be used in conjunction with the `/ON` or `/GENERIC` qualifier. However, if you are reinitializing an existing queue, you can specify the `/AUTOSTART_ON` qualifier for a queue previously created or started with the `/ON` qualifier. Doing so overrides the `/ON` qualifier and makes the queue an autostart queue.

`/ON=[node::]device[:]`

Specifies the node or device, or both, on which this execution queue is located. For output execution queues, you can specify both the node name and the device name. By default, a queue executes on the same node from which you start the queue. The default device parameter is the same as the queue name.

## Set the Printer Characteristics

A command procedure to be executed at system initialization sets the printer characteristics on each node that will use the queue.



---

**Note**

How is this made permanent? In the system startup command procedure.

---

## SET TERMINAL Command

Format:

`$ SET TERMINAL application-port-name`

Qualifiers:

|                                     |                                                                                                                                                                                                                                                                             |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>/PERMANENT</code>             | Sets characteristics to survive terminal sessions. Characteristics revert to original values if the system is halted and restarted.                                                                                                                                         |
| <code>/DEVICE=terminal- type</code> | Specifies the type of terminal and sets characteristics according to the terminal type.                                                                                                                                                                                     |
| <code>/WIDTH=x</code>               | Specifies the maximum characters line. This value must be an integer in the range of 1 to 511 and should be set to the maximum number of characters for all print forms.                                                                                                    |
| <code>/PAGE=y</code>                | Specifies the number of print lines perforations for hardcopy terminals. (When the terminal reads a form feed, it advances the paper to the next perforation.) The value of the parameter can be from 0 to 255 and defaults to 0 (which treats a form feed as a line feed). |
| <code>/NOBROADCAST</code>           | Controls whether reception of OpenVMS broadcast messages is enabled. When used with a print queue, stops messages from interrupting the printing of the output file.                                                                                                        |

Example:

`$ SET TERMINAL/PERMANENT/DEVICE=LN03/NOBROADCAST/PAGE=60 LTA8000:`



---

**Note**

The DCL command SET TERMINAL used without specifying the `/PAGE=lines` defaults to 0. This means the printer will recognize a form feed to advance the page.

The qualifiers `/PERMANENT` and `/NOBROADCAST` must be specified for application ports.

`/DEVICE=keyword`

Specifies the type of output execution queue that you want to initialize or start. The keywords are as follows:

- PRINTER (default)
  - TERMINAL
  - SERVER -- Specifies a server queue (use `/PROCESSOR=filename` to specify the user-written symbiont).
-

## Set the Device Protection

The application port should only be accessed by the system and not by owner, group, or world.

```
$ SET SECURITY/CLASS=DEVICE/PROTECTION=(S:RWLP,O,G,W) LTA####:
```

The access categories:

|      |                                               |
|------|-----------------------------------------------|
| R    | Can issue read requests to the device         |
| W    | Can issue write requests to the device        |
| L    | Can issue logical I/O requests to the device  |
| P    | Can issue physical I/O requests to the device |
| Null | No access                                     |



### Note

```
$ SET SECURITY/CLASS=DEVICE/PROTECTION
```

This establishes the protection to be applied to a specific non-structured device. The protection for a device limits the type of access available to users. The /DEVICE qualifier is required.

## Set the Device Spooled

The remote print queue device should be set spooled to prevent applications from sending output directly to the printer.

Format:

```
$ SET DEVICE LTA####:/SPOOLED
```



### Note

Setting the device spooled prevents a privileged user from sending a direct COPY command to the printer.

## Initialize the Queue

Create or initialize the queue.

Format:

```
$ INITIALIZE/QUEUE queue-name
```

Special qualifiers to remote print queues:

|                                |                                                                                                                    |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------|
| /PROCESSOR = LATSYM            | Specifies that the print symbiont be the one designed specifically for use with the LAT protocol                   |
| /AUTOSTART_ON=(node::device,.) | Associates the queue with the application port, which was mapped previously to a particular server and server port |
| /START                         | Makes the queue active                                                                                             |

You can also use other INITIALIZE qualifiers appropriate for print queues.



---

**Note**

As of OpenVMS Version 5.5, the print queues do not need to be initialized each time the system is rebooted. This is also of the print form's definitions since both are stored in the queue database.

See the *OpenVMS Cluster Manual* for a new command procedure to replace REMOTE\_PRINT.COM.

On each node the ENABLE command must be issued.

---

## Test the Printer

Test the printer setup by printing a file on the printer.

```
$ PRINT/QUEUE=laser myfile.txt
```

