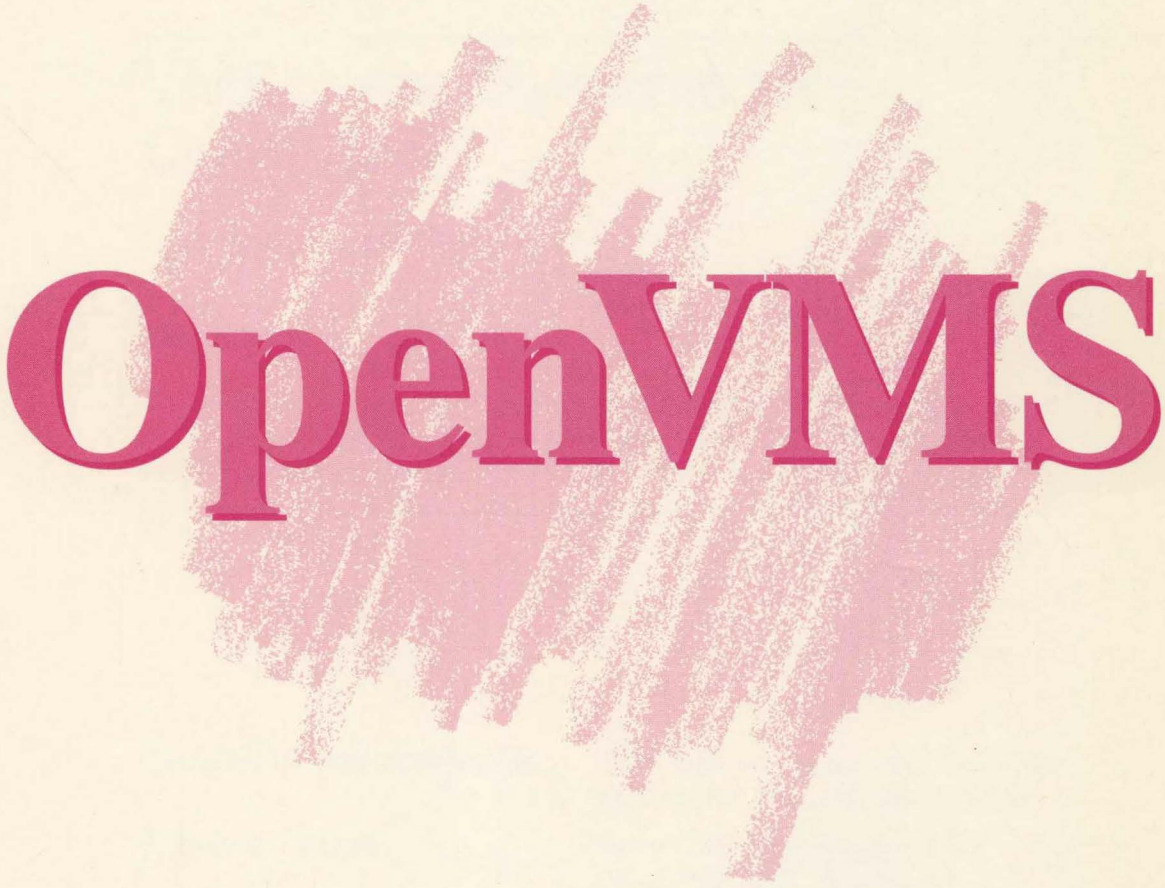
digital

VMScluster Systems for OpenVMS



Part Number: AA–PV5WA–TK

VMScluster Systems for OpenVMS

Order Number: AA–PV5WA–TK

May 1993

This manual describes procedures and guidelines for configuring and managing VAXcluster and VMScluster systems. The document also includes information for providing high availability, building-block growth, and unified system management across coupled systems.

Revision/Update Information:	This manual supersedes the *VMS VAXcluster Manual*, Version 5.5.
Software Version:	OpenVMS AXP Version 1.5 OpenVMS VAX Version 6.0

Digital Equipment Corporation/Maynard, Massachusetts

May 1993

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

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

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

© Digital Equipment Corporation 1993.

All Rights Reserved.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation: Alpha AXP, AXP, BI, Bookreader, CI, DEC, DECdtm, DECmcc, DECnet, DECram, DECwindows, DELNI, DEMPR, DEQNA, DESTA, Digital, HSC, KDA, LAN Bridge 200, LAT, MASSBUS, MicroVAX, MicroVAX II, MSCP, OpenVMS, Q–bus, RA, SDI, STI, ThinWire, TK, TMSCP, TU, UDA, UNIBUS, VAX VAX DOCUMENT, VAX 11/750, VAX 11/780, VAX 6000, VAX 8200, VAX 8250, VAX 8300, VAX 8350, VAX 8600, VAX 9000, VAX RMS, VAXcluster, VAXft, VAXstation, VAXstation 4000 VLC, VMS, VMScluster, VT, VT200, VT300, VT320, VT330, VT340, VT420, XMI, XUI, the AXP logo, and the DIGITAL logo.

The following are third-party trademarks:

Hewlett-Packard, HP, and HP 4927A LAN Protocol Analyzer are registered trademarks of the Hewlett-Packard Company.

Open Software Foundation is a trademark of the Open Software Foundation, Inc.

PostScript is a registered trademark of Adobe Systems Incorporated.

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

ZK4477

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

4 Preparing the Cluster Operating Environment

5 Setting Up and Managing Cluster Disks and Tapes

6 Setting Up and Managing Cluster Queues

7 Building and Maintaining the Cluster

D Local Area VMScluster Network Connections

E Local Area VMScluster Sample Programs

F Local Area VMScluster Subroutine Package

G Troubleshooting the NISCA Protocol

Figures

Tables

Preface

VMScluster Systems for OpenVMS describes system management for both VAXcluster systems and VMScluster systems. Although the two products are separately purchased, licensed, and installed, the difference between the two architectures lies mainly in the hardware used. Essentially, system management for VAXcluster systems and VMScluster systems is identical. Exceptions are pointed out in the manual titled *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX.*

Intended Audience

This document addresses persons responsible for setting up and managing VAXcluster and VMScluster systems. To use the document as a guide to cluster management, you must have a thorough understanding of system management concepts and procedures, as described in the *OpenVMS System Manager's Manual.*

Document Structure

VMScluster Systems for OpenVMS contains seven chapters and nine appendixes.

Chapter 1 introduces VAXcluster and VMScluster systems.

Chapter 2 describes various VAXcluster and VMScluster configurations and the ways they are interconnected.

Chapter 3 presents the software concepts integral to maintaining cluster integrity and security.

Chapter 4 explains how to prepare the cluster operating environment.

Chapter 5 discusses disk and tape management concepts and procedures and how to use Volume Shadowing for OpenVMS on VAX nodes to prevent data unavailabity.

Chapter 6 discusses queue management concepts and procedures.

Chapter 7 explains how to build a VMScluster system once the necessary preparations are made, and how to reconfigure and maintain the cluster.

Appendix A lists and defines cluster system parameters.

Appendix B provides guidelines for building a cluster common user authorization file.

Appendix C provides troubleshooting information.

Appendix D provides information on local area VMScluster network connections.

Appendix E presents three sample programs for local area VMSclusters and explains how to use the Local Area VMScluster Network Failure Analysis Program.

Appendix F describes the subroutine package used with local area VMScluster sample programs.

Appendix G provides techniques for troubleshooting network problems related to the NISCA transport protocol.

Appendix H describes how the interactions of workload distribution and network topology affect VMScluster system performance.

Appendix I discusses transmit channel selection by PEDRIVER.

Associated Documents

This document is not a one-volume reference manual. The utilities and commands are described in detail in the *OpenVMS System Manager's Manual*, the *OpenVMS System Management Utilities Reference Manual*, and the *OpenVMS DCL Dictionary*.

For additional information on the topics covered in this manual, refer to the following documents:

- *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX*
- *Building Dependable Systems: The OpenVMS VAX Approach*
- *DECnet for OpenVMS Network Management Utilities*
- *DECnet for OpenVMS Networking Manual*
- *Guide to OpenVMS File Applications*
- *Guidelines for VAXcluster System Configurations*
- *OpenVMS AXP Guide to System Security*
- *OpenVMS VAX Guide to System Security*
- *OpenVMS AXP System Dump Analyzer Utility Manual*
- *OpenVMS VAX System Dump Analyzer Utility Manual*
- *OpenVMS I/O User's Reference Manual*
- *OpenVMS License Management Utility Manual*
- *OpenVMS System Management Utilities Reference Manual*
- *OpenVMS System Manager's Manual*
- *OpenVMS System Services Reference Manual*
- *Volume Shadowing for OpenVMS* (an optional OpenVMS manual)
- VAXcluster Software for OpenVMS VAX *Software Product Description* (SPD 29.78.*xx*)
- VMScluster Software for OpenVMS AXP *Software Product Description* (SPD 42.18.*xx*)

Conventions

In this manual, every use of OpenVMS AXP means the OpenVMS AXP operating system, every use of OpenVMS VAX means the OpenVMS VAX operating system, and every use of OpenVMS means both the OpenVMS AXP operating system and the OpenVMS VAX operating system.

Note

Discussions that refer to VMScluster environments apply to both VAXcluster systems that include only VAX nodes and VMScluster systems that include at least one AXP node, unless indicated otherwise. When the behavior differs significantly between a VAXcluster and VMScluster system, that behavior is described in text and is marked with the AXP or VAX icon, as appropriate.

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

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

The following conventions are also used in this manual:

Ctrl/*x*	A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
GOLD *x*	A sequence such as GOLD *x* indicates that you must first press and release the key defined GOLD, then press and release another key. GOLD key sequences can also have a slash (/), hyphen (-), or underscore (_) as a delimiter in EVE commands.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
. . .	A horizontal ellipsis in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.

. . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.
[ ]	In format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in an OpenVMS file specification, or in the syntax of a substring specification in an assignment statement.)
{ }	In format descriptions, braces surround a required choice of options; you must choose one of the options listed.
boldface text	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason. Boldface text is also used to show user input in Bookreader versions of the manual.
italic text	Italic text emphasizes important information, indicates variables, and indicates complete titles of manuals. Italic text also represents information that can vary in system messages (for example, Internal error *number*), command lines (for example, /PRODUCER=*name*), and command parameters in text.
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
-	A hyphen in code examples indicates that additional arguments to the request are provided on the line that follows.
numbers	All numbers in text are assumed to be decimal, unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

Note

All references to DECnet in this document mean the DECnet for OpenVMS software.

1

Introduction to VMScluster Environments

A VMScluster system is a highly integrated organization of AXP computers or a combination of VAX and AXP computers. A VAXcluster system consists of VAX computers only.

Because the system management of these systems is essentially the same, all discussions in this manual that refer to VMScluster environments apply equally to VAX systems and AXP systems when configured together in a single VMScluster system. Exceptions are clearly noted.

As members of a VMScluster system, AXP and VAX computers can share processing resources, data storage, and queues under a single security and management domain, and they can boot or fail independently. A system disk is one of the few resources that cannot be shared between the two systems. However, an AXP system can mount a VAX system disk as a data disk, and a VAX system can mount an AXP system disk as a data disk.

Using procedures described in Chapter 4, system managers can tailor the cluster operating environment to create a **common-environment** or a **multiple-environment** VMScluster system.

- In a common-environment VMScluster system, the same resources are available on all computers. User accounts are identical, the same known images are installed, the same logical names are defined, and mass storage devices and queues are shared.
- In a multiple-environment VMScluster system, a group of computers shares one set of resources, while another group shares a different set. Alternatively, an individual computer can perform a specialized function using restricted resources, while other computers perform general timesharing work.

Although most cluster resources can be shared, user processes and memory are computer specific. When a process is created on a VMScluster computer, the process must complete on that computer, using local memory. If the computer shuts down before the process completes, the process is terminated. However, users can recover from such a failure more quickly than on a standalone computer, because they need not wait until the computer is rebooted. Typically, they can log in on another VMScluster computer to create a new process and continue working, provided that the resources required by the process (such as images and disk files) are available on that computer.

This chapter describes VMScluster operating features and components, including the following:

- Shared resources
- Interconnect devices
- Software components

- Configuration planning

Be sure you understand these topics before you attempt to perform any cluster setup operations.

1.1 Shared Resources

In any VMScluster system, users can share computing, disk storage, and batch and print processing resources. The ability to share resources facilitates workload balancing, because work can be distributed across the cluster. To keep pace with user demand, resources can be added without disrupting normal cluster operation.

1.1.1 Disk and Tape Storage

A major advantage of VMScluster systems is the ability to make disk and tape storage resources accessible to all VMScluster computers. Storage devices such as Digital Storage Architecture (DSA) disks and tapes, RF series **integrated storage elements (ISEs)**, Small Computer Systems Interface devices (SCSI), and ESE20 solid state disks can be configured for local or clusterwide access. A **cluster-accessible** storage device can be used directly by multiple computers in the cluster. By means of the OpenVMS MSCP and TMSCP server software, disks and tapes can be made accessible to nodes that are not directly connected.

Cluster-accessible disks offer the following advantages:

- More efficient use of mass storage, because more than one computer can use the same disk.
- Access by users to their default work disks when logging in to any computer on which the disks are accessible.
- Clusterwide file sharing. Because computers can share common versions of files, updates to a file are made only once to a single copy of the file.
- Implementation of clusterwide job-controller queues. Batch and print processing can be done on any computer that has access to the disks.

Some VMScluster systems include HSC subsystems. These are self-contained, intelligent mass storage subsystems that enable VMScluster computers to share DSA disks and, in the VMScluster configurations described in Section 2.2, DSA tapes. Because the HSC subsystem is an intelligent controller, it optimizes physical disk and tape operations and supports many combinations of standard disk interfaces (SDIs) and standard tape interfaces (STIs), that connect disks and tapes. HSC disk configurations provide flexibility, expansion potential, maintenance, and backup capability.

Disk data can be replicated within VMScluster configurations for higher availability using volume shadowing, as described in Section 5.6. For detailed information about volume shadowing, see *Volume Shadowing for OpenVMS*.

Procedures for setting up and managing cluster disks and tapes are described in Chapter 5.

1.1.2 Batch and Print Job Processing

System managers control how jobs share batch processing and printer resources by setting up and maintaining clusterwide generic queues. The strategy for setting up and managing these queues determines how well work loads are matched to available resources.

On AXP systems running Version 1.5 and on VAX systems running Version 5.5–2 queues are controlled by a clusterwide queue manager process that accesses the clusterwide queue database for all processes in a cluster. Job controllers, user processes, and symbionts all communicate directly with the centralized queue manager through a shared interprocess communication (IPC) link. The centralized queue manager makes queues available across the cluster and enables jobs to execute on any queue from any AXP or VAX computer, provided that the necessary mass storage volumes can be accessed by the computer on which the job executes.

Procedures for setting up and managing cluster queues are described in Chapter 6.

(Note that a generic queue holds a job that will execute on an execution queue when it is available to process jobs. When the job is sent to an execution queue on a specific node, it is executed on that queue.)

1.2 Interconnect Devices

Interconnect devices used to configure a VMScluster system include the following:

- **CI.** This high-speed, dual-pathed interface links computers and HSC subsystems in a computer room environment. A computer interconnect consists of several components, such as CI port controllers (adapters), the star coupler, star coupler expander (CISCE), and the high-bandwidth CI cables themselves. All nodes on a common CI must be part of the same VMScluster.
- **CI port controllers.** Port controllers like the CI780, CIBCA, CIBCI, and CIXCD (CI to XMI adapter) are microcoded, intelligent adapters that connect computers to CI cables. Each interface connects to one pair of transmitter cables and one pair of receiver cables.

  Under normal operating conditions, both pairs of cables are available to meet traffic demands. If one path fails, all traffic uses the remaining path. The operating system software periodically tests a failed path. As soon as a failed path is restored, it is automatically used for normal traffic.
- **Star couplers and star coupler expanders (CISCE).** Star couplers and star coupler expanders provide common connection points for CI connected computers and HSC subsystems. Both coupler devices connect CI cables from computers and HSC subsystems, creating a radial or "star" arrangement that has a maximum radius of 45 meters (147 feet). These devices support the physical connection and disconnection of individual computers and HSC subsystems without affecting other computers or HSC subsystems.

  The star coupler and CISCE are dual-pathed devices that contain separate components for each path. The star coupler is a passive device; the CISCE consists of redundant amplifiers. Both devices are designed so that all CI cables are transformer coupled and independent of earth ground reference. These attributes help to ensure signal integrity.

- **Digital Storage Systems Interconnect (DSSI)**. The DSSI bus permits multiple computers to communicate directly with storage devices. The DSSI bus connects as many as eight nodes that can be ISEs or host CPU interfaces.
- **DSSI port controllers**. Port controllers like the KFMSA, KFQSA, and EDA-type adapters are intelligent adapters that connect computers to DSSI buses.
- **Local area network (LAN) adapters**. LAN adapters include:
  - Ethernet—The Ethernet is a bus that uses digital baseband signaling.
  - Fiber Distributed Data Interface (FDDI)—FDDI is an ANSI-standard LAN interconnect based on a fiber-optic transmission medium.

  The LAN is used both for DECnet for OpenVMS transmissions and, in some VMScluster systems, for cluster communications. The LAN must be configured according to requirements specified in the VMScluster Software for OpenVMS AXP *Software Product Description* (SPD 42.18.*xx*) or the VAXcluster Software for OpenVMS VAX *Software Product Description* (SPD 29.78.*xx*).

Depending on how a VMScluster system is configured, VMScluster software can use multiple interconnects for cluster communications.

1.3 Software Components

The software components used to implement VMScluster communication and resource-sharing functions always run on each computer in the cluster. Thus, if one computer fails, the VMScluster system continues operating, because the components still run on the remaining computers. These software components are as follows:

- **System Communications Services (SCS)** software implements intercomputer communication, according to the System Communications Architecture (SCA).

  All cluster **system applications (SYSAPs)** (for example, the distributed lock manager, disk and tape class drivers, and MSCP server) use SCS software for interprocessor communication.
- **Port drivers** control the communication paths between local and remote ports. (Examples are PADRIVER for the CI, PEDRIVER for the LAN, PIDRIVER for the DSSI, and PNDRIVER for the CI on AXP computers.)
- The **connection manager** dynamically defines the VMScluster system and coordinates participation of computers in the cluster. The connection manager uses SCS to provide an acknowledged message delivery service for higher software layers. The connection manager also maintains cluster integrity when computers join or leave the cluster—that is, when cluster **state transitions** occur. (State transitions are discussed in Section 3.1.3.)
- The VMScluster **distributed file system** allows all computers to share mass storage, whether the storage device is connnected to an HSC, DSSI, or SCSI subsystem. Any disk can be made available to the entire cluster. All cluster-accessible disks appear as if they are connected to every computer.

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

- The **distributed lock manager** is used for synchronization functions by the distributed file system, job controller, device allocation, and other cluster facilities. It is available to users for developing cluster applications. The distributed lock manager implements the $ENQ and $DEQ system services to provide clusterwide synchronization of access to resources by allowing the locking and unlocking of resource names. (For detailed information on system services, refer to the *OpenVMS System Services Reference Manual.*) The distributed lock manager also provides a queuing mechanism so that processes can be put into a wait state until a particular resource is available. As a result, cooperating processes can synchronize their access to shared objects, such as files and records.

  If a VMScluster computer fails, all locks that it holds are released. This mechanism allows processing to continue on the remaining computers. The distributed lock manager also supports clusterwide deadlock detection.

- The **distributed job controller** makes queues available across the cluster. VMScluster computers can share batch and print queues. Users can submit jobs to any queue in the cluster, provided that the necessary mass storage volumes and peripheral devices are accessible to the computer on which the job executes. System managers can also set up generic batch and print queues that distribute processing work loads among computers. For detailed information about VMScluster queues, see Chapter 6.

- The **MSCP** server implements that mass storage control protocol which is used to communicate with a controller for DSA disks, such as RA disks. In conjunction with one or both of the disk class drivers (DUDRIVER, DSDRIVER), the MSCP server implements this protocol on a computer, allowing the computer to function as a storage controller. The computer submits I/O requests to locally accessed disks, such as locally connected RA disks or Small Computer Systems Interface (SCSI) disks, and accepts the I/O requests from any computer in the cluster. In this way, the MSCP server makes locally connected disks available across the cluster. In the local area and mixed-interconnect VMScluster systems described in Section 2.4 through Section 2.6, the MSCP server can also make HSC disks and DSSI ISE disks accessible over the LAN.

- The **TMSCP** server implements the tape mass storage control protocol, which is used to communicate with a controller for local MSCP tapes, such as TU series tapes. In conjunction with the tape class driver (TUDRIVER), the TMSCP protocol is implemented on a processor, allowing the processor to function as a storage controller.

  The processor submits I/O requests to locally accessed tapes, such as Q–bus tapes, and accepts the I/O requests from any node in the cluster. In this way, the TMSCP server makes locally connected tapes available to all nodes in the cluster. The TMSCP server can also make HSC tapes and DSSI ISE tapes accessible to VMScluster satellites.

In addition to these components, all VMScluster systems require **DECnet for OpenVMS software**, which ensures that system managers can access all VMScluster computers from a single terminal, even if terminal switching facilities are unavailable.

In VMScluster systems, DECnet for OpenVMS software is required both for system management functions and for remote booting operations.

In these systems, DECnet and SCS software coexist on the same extended LAN. They share the same data link and physical link protocols, which are implemented by the LAN data link drivers, the LAN adapters, and the LAN itself.

1.4 Configuration Planning

The process of setting up a VMScluster system requires careful preparation. In planning your configuration, you must determine the following:

- **Interconnect type.** If you want to include 3000-class computers or workstations in your VMScluster system, you must set up a local area or mixed-interconnect configuration. These configurations are described in Chapter 2.
- **Integrity and security.** Chapter 3 describes the rules for maintaining VMScluster integrity and security.
- **Operating environment (common or multiple).** These environments are described at the beginning of this chapter and in Chapter 4, which provides information about configuring the DECnet network and on preparing the startup, user authorization, and other files that define the operating environment.
- **Disk and tape storage configuration.** Chapter 5 provides information about disk and tape storage configurations, including descriptions of disk types, rules for specifying disk and tape names in VMScluster systems, sample disk and tape configurations, and the Volume Shadowing for OpenVMS product that provides for data availability.
- **Queue configuration.** Chapter 6 provides information about VMScluster queues and includes a sample queue setup command procedure.
- **Computer configuration.** Procedures for configuring VMScluster computers are described in Chapter 7. That chapter also includes a detailed discussion of the cluster configuration command procedure, SYS$MANAGER:CLUSTER_CONFIG.COM.

Once you have planned your configuration, installed the necessary hardware, and checked hardware devices for proper operation, you can set up the cluster using various system software facilities. Setup procedures are typically as follows:

- Installing or upgrading the operating system on the first VMScluster computer. Follow the instructions in the installation and operations guide for your computer.
- Installing required software licenses. Follow the instructions in the *OpenVMS License Management Utility Manual.*
- Configuring and starting the DECnet network. Follow the instructions in Chapter 4. For more detailed information about network operations, refer to the *DECnet for OpenVMS Networking Manual.*
- Preparing files that define the cluster operating environment and that control disk and queue operations. Follow the instructions in Chapters 4, 5, and 6.
- Adding computers to the cluster. Follow the instructions in Chapter 7.

Depending on various factors, the order in which these operations are performed can vary from site to site, as well as from cluster to cluster at the same site.

2

VMScluster Interconnect Configurations

This chapter discusses the various types of VMScluster configurations and the ways they are interconnected.

2.1 Configuration Types

While processing needs and available hardware resources must determine how individual VMScluster systems are configured, sites can choose from the following:

- CI VMScluster systems
- DSSI VMScluster systems
- Local area VMScluster systems
- Mixed-interconnect VMScluster systems

These configuration types are described in the following sections. Configurations that include FDDI connections are described in Section 2.6. For complete information about currently supported configurations, refer to the VAXcluster Software for OpenVMS VAX *Software Product Description* (SPD 29.78.*xx*) or the VMScluster Software for OpenVMS AXP *Software Product Description* (SPD 42.18.*xx*), as appropriate.

2.2 VMScluster Systems Based on CI

A VMScluster system based on the CI for cluster communications uses star couplers as common connection points for computers and HSC subsystems.

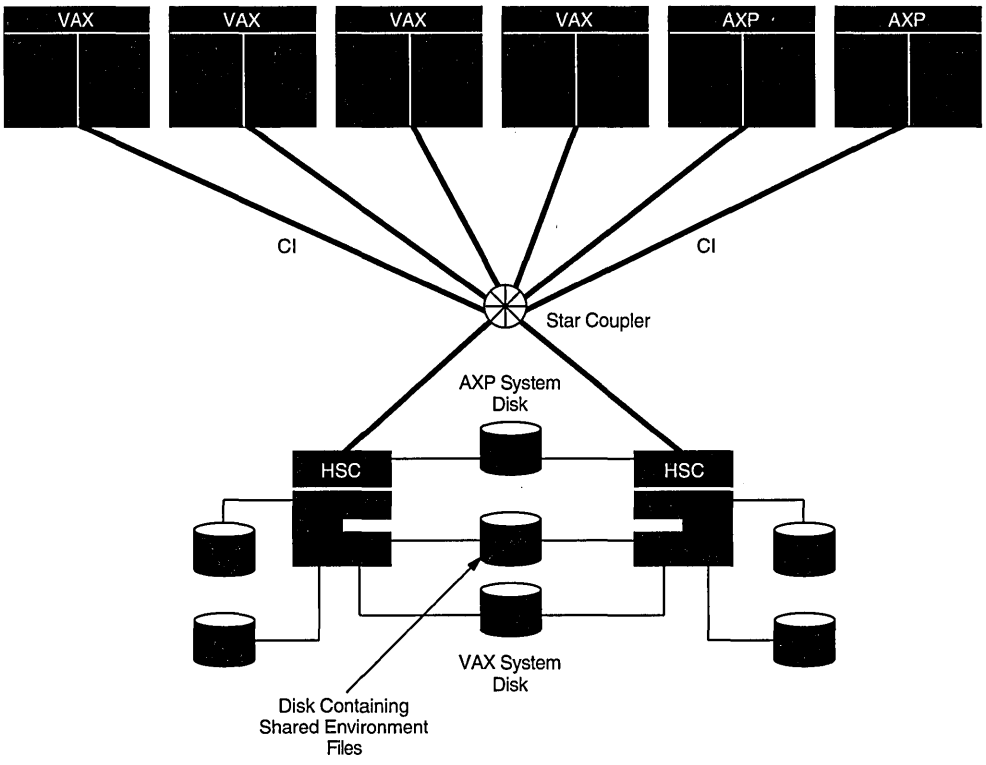
The CI is a high-performance, fault-tolerant way to connect AXP and VAX nodes to disk and tape storage devices and to each other. The star topology inherent in a CI VMScluster system utilizes redundant coaxial cables in which two transmit and two receive cables form a single CI physical connection; if one cable fails, the others continue to provide service. Star couplers enable all nodes and the HSC disk and tape controllers to communicate directly. The star couplers operate at full cable bandwidth and are also dual redundant.

The HSC intelligent disk and tape controllers also connect to star couplers via the CI cables. Because the Digital Storage Architecture (DSA) disk and tape drives have dual access ports, you can configure HSC controllers in a redundant fashion. By having two HSC controllers connected to each disk or tape, if one of the HSC controllers fails, the other continues to provide service. Failover support in the OpenVMS operating system makes this capability transparent to application programs.

In the CI configuration shown in Figure 2–1, the AXP nodes boot from the common AXP system disk, and VAX nodes boot from a common VAX system disk. This restriction ensures that VAX systems do not try to downline load AXP systems. Once booted, the AXP node can access data on the VAX system disk

and the VAX nodes can access data on the AXP system disk. You can set up the VMScluster environment so that other environment files can be shared between the two architectures. In addition, quorum access can be provided by all nodes. Figure 2–1 shows how the CI components are typically configured.

Figure 2–1 VMScluster Configuration Based on CI



What appears to be a single point of failure is the star coupler that connects all the CI lines in the figure. The star coupler is not a single point of failure because there are actually two star couplers in every box: one for the path A cables and one for the path B cables. Star couplers are also immune to power failures because they contain no powered components but are constructed as sets of high-frequency pulse transformers. Because they do no processing or buffering, star couplers also are not I/O throughput bottlenecks. They operate at the full-rated speed of the CI cables. However, in very heavy I/O situations, it is possible to overload the CI cable/coupler system and thus to require multiple star couplers.

All AXP and VAX nodes in any type of VMScluster must have direct connections to all other nodes. The important detail about the CI that is not shown in Figure 2–1 is that the computer interconnect is a fully redundant communications medium. Each CI adapter connects to four coaxial cables, two for transmitting and two for receiving. The full CI bandwidth is available over either path A or path B.

Note that if you want to add workstations to a CI VMScluster system, you must utilize another type of interconnect such as Ethernet or DSSI in the configuration. For descriptions of configurations that include multiple interconnects (sometimes

referred to as mixed-interconnect systems), refer to Section 2.5. For instructions on adding satellites to an existing CI VMScluster system, refer to Section 7.5.4.

2.3 VMScluster Configurations Based on DSSI

The DSSI is a high-bandwidth interconnect that AXP and VAX nodes can use to access disk and tape peripherals. Each peripheral is an ISE that contains its own controller and its own MSCP server that works in parallel with the other ISEs on the DSSI. Satellites (and users connected through terminal servers) can access any disk through either boot server. If one of the boot servers fails, applications on satellites continue to run because disk access fails over to the other server. The combination of high throughput and common data access by both AXP and VAX nodes provides a way to configure VMScluster systems at much lower cost than using the CI and HSC controllers, but with much greater capacity than using only the Ethernet interconnect (described in Section 2.4).

VAX

Because VAX nodes have direct access to the VAX system disks, DSSI VAXcluster systems can support rolling upgrades. This capability allows the version of the operating system to be upgraded on one node and that node to be rebooted while the other node is still providing service.♦

AXP

On AXP systems, AXP nodes in DSSI VMScluster systems do not support rolling upgrades.♦

Generic guidelines for DSSI VMScluster systems are as follows:

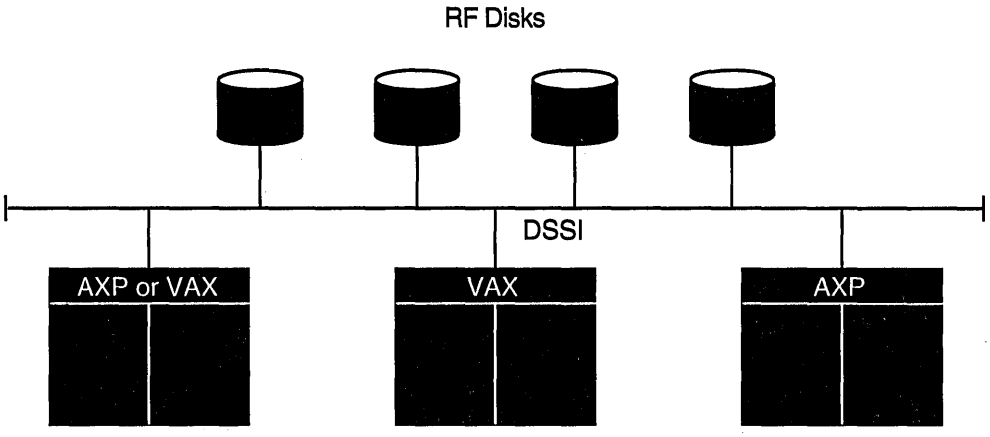
- Currently, a total of four AXP and/or VAX nodes can be connected to a common DSSI.
- Multiple DSSI buses can operate in a VMScluster configuration. The maximum number of nodes in a VMScluster configuration is not increased by having multiple DSSI buses. However, the maximum number of ISE controllers is dramatically increased, allowing much more storage to be configured into the system.

Some restrictions apply to the type of CPUs and DSSI I/O adapters that can reside on the same DSSI bus. Consult your Digital Services group or see the Software Product Description (SPD) for your VMScluster or VAXcluster configuration for complete and up-to-date configuration details about DSSI VMScluster systems. See also the DSSI installation manual for specific CPU or adapter types.

In the DSSI configuration shown in Figure 2–2, the AXP boot server and the AXP satellites boot from a common AXP system disk. Similarly, the VAX boot server and the VAX satellites boot from a common VAX system disk. This restriction ensures that VAX systems do not try to downline load AXP systems.

Once booted, the AXP node can access VAX data and system disks and VAX nodes can access AXP data and system disks. Moreover, using the MSCP server, the AXP node can serve the VAX disks to both VAX and AXP satellites, and the VAX node can serve both AXP and VAX satellites. This flexibility in dual-architecture configurations provides full data access regardless of a system failure on either the VAX or AXP node.

Figure 2–2 DSSI VMScluster Configuration



In addition to the boot servers, a DSSI VMScluster configuration typically includes several DSSI-connected ISEs and either a TK70 tape drive or DSSI TF tape drives serving the configuration. AXP and VAX satellite nodes can access the tape drives through the systems on the DSSI that are running the TMSCP server.

Using additional DSSI cables, you can connect either one or two **storage expansion boxes** containing additional ISEs. If you decide to use a storage expansion box, it is a good idea to place a common system disk and critical data disks in the expansion box, which has a dedicated power supply. Thus, if one boot server fails, the other servers and satellites can still access the disks.

Note that there are specific rules for various DSSI configurations; for example, the setup for a DSSI MicroVAX 3800 system is different from that for a DSSI VAX 4000 system or for a DEC 4000 system. These rules are provided in the appropriate system installation and user manuals.

2.4 Local Area VMScluster Systems

In local area VMScluster systems, cluster communications are carried out over the LAN by a port driver that emulates certain CI port functions.

A single LAN can support multiple local area VMScluster systems. Each system is identified and secured by a unique cluster **group number** and a **cluster password**. (For information about maintaining the integrity of cluster membership, see Section 3.1.3.1.)

Computers in a local area VMScluster are generally configured as either servers or satellites. The following list describes the types of servers:

- **MOP servers** downline load the boot driver to satellites by means of the DECnet Maintenance Operations Protocol (MOP). When a satellite requests an operating system load, a MOP server for the appropriate OpenVMS AXP or OpenVMS VAX operating system sends an image to the satellite that allows the satellite to load the operating system from a disk server and join the cluster. VMScluster satellites can be configured as additional MOP servers.

- **Disk servers** use MSCP server software to make their locally connected disks and any CI or DSSI connected disks available to satellites over the LAN.
- **Tape servers** use TMSCP server software to make their locally connected tapes and any CI or DSSI connected tapes available to satellite nodes over the LAN.
- **Boot servers** are a combination of a MOP server and a disk server that serves one or more AXP or VAX system disks. Boot and disk servers make user and application data disks available across the cluster. These servers should be the most powerful computers in the VMScluster and should use the highest-bandwidth LAN adapters in the cluster. Boot servers must always run the MSCP server software.

  Typically, a boot server is both a management center for the VMScluster and a major resource provider. Its system disk contains the cluster common files for startup, authorization, and queue setup, as well as the root directories from which the satellites are booted and in which their specific system files reside. These root directories, one for each satellite, are created when system managers add satellites to the cluster using the CLUSTER_CONFIG.COM command procedure described in Chapter 7.

  Note

  An AXP system disk cannot be used to boot VAX computers and an VAX system disk cannot be used to boot AXP computers.

Satellites are computers without a local system disk.

AXP

AXP satellites are booted remotely from an AXP boot server or from an AXP MOP server and a disk server serving the AXP system disk.♦

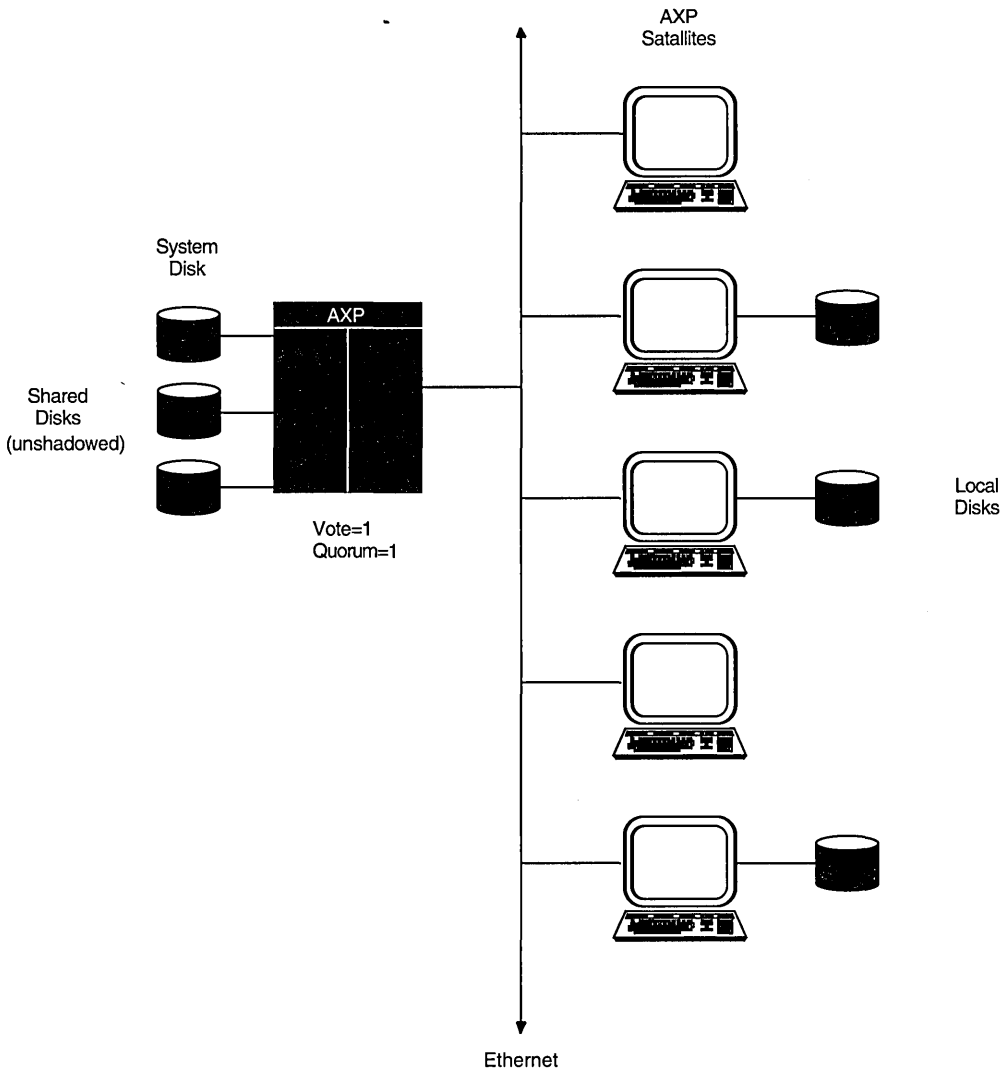
VAX

VAX satellites are booted remotely from a VAX boot server or from a VAX MOP server and a disk server serving the VAX system disk.♦

Generally, satellites are consumers of cluster resources, though they can also provide facilities for disk serving, MOP serving, tape serving, and batch processing. If satellites are equipped with local disks, they can enhance performance by using such local disks for paging and swapping. Section 7.3 describes MOP and disk server functions during satellite booting.

Figure 2–3 shows a local area VMScluster system with a single AXP boot server. Note that, because all computers in this configuration rely on the AXP boot server's system disk, only AXP satellites can be booted into this configuration. Replace all AXP nodes with VAX computers for a valid, single-boot node VAXcluster system.

Figure 2–3 Local Area VMScluster System with Single Boot Server



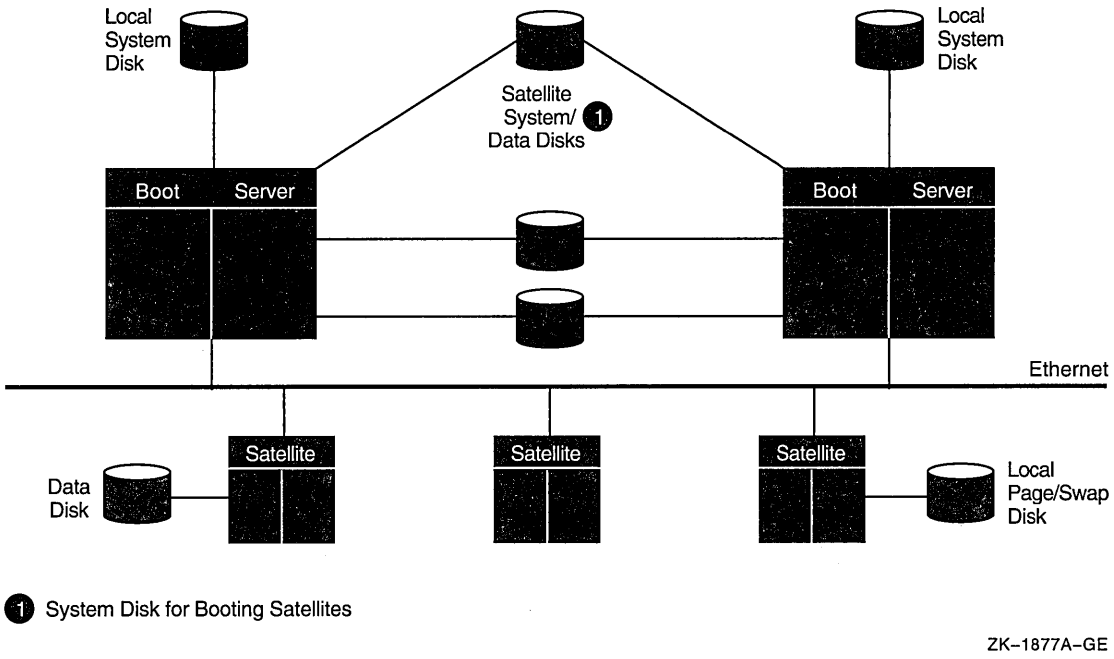
In Figure 2–3, the boot server (and its system disk) is a single point of failure. If the boot server fails, the satellite nodes cannot access any of the shared disks including the system disk. Note that some of the satellite nodes have locally connected disks. If you convert one or more of these into system disks, satellite nodes can boot from their own local system disk.

You can increase the availability of the VMScluster system by using two boot servers with locally connected system disks. A satellite system disk can be dual pathed between the servers.

Figure 2–4 shows such a configuration that uses DSA disks. The boot servers boot from their locally connected system disks, and the satellites boot from the satellite system disk, which contains their root directories. Thus, if one boot server fails, satellites can access their system disk through the other server. If the satellite system disk fails, it can be restored from the remaining boot server, and the satellites can then be rebooted.

To increase availability even further, use a DSSI configuration (see Section 2.3). Any local area VMScluster system can be converted to use a mixture of interconnects (such as CI, DSSI, and FDDI). Refer to Section 7.5.4 for instructions.

Figure 2–4 VMScluster Configuration with Redundant Boot Servers and DSA Disks



2.5 Mixed-Interconnect VMScluster Systems

A mixed-interconnect VMScluster system is any VMScluster system that utilizes more than one interconnect for SCA traffic. Examples of mixed-interconnect VMScluster configurations are:

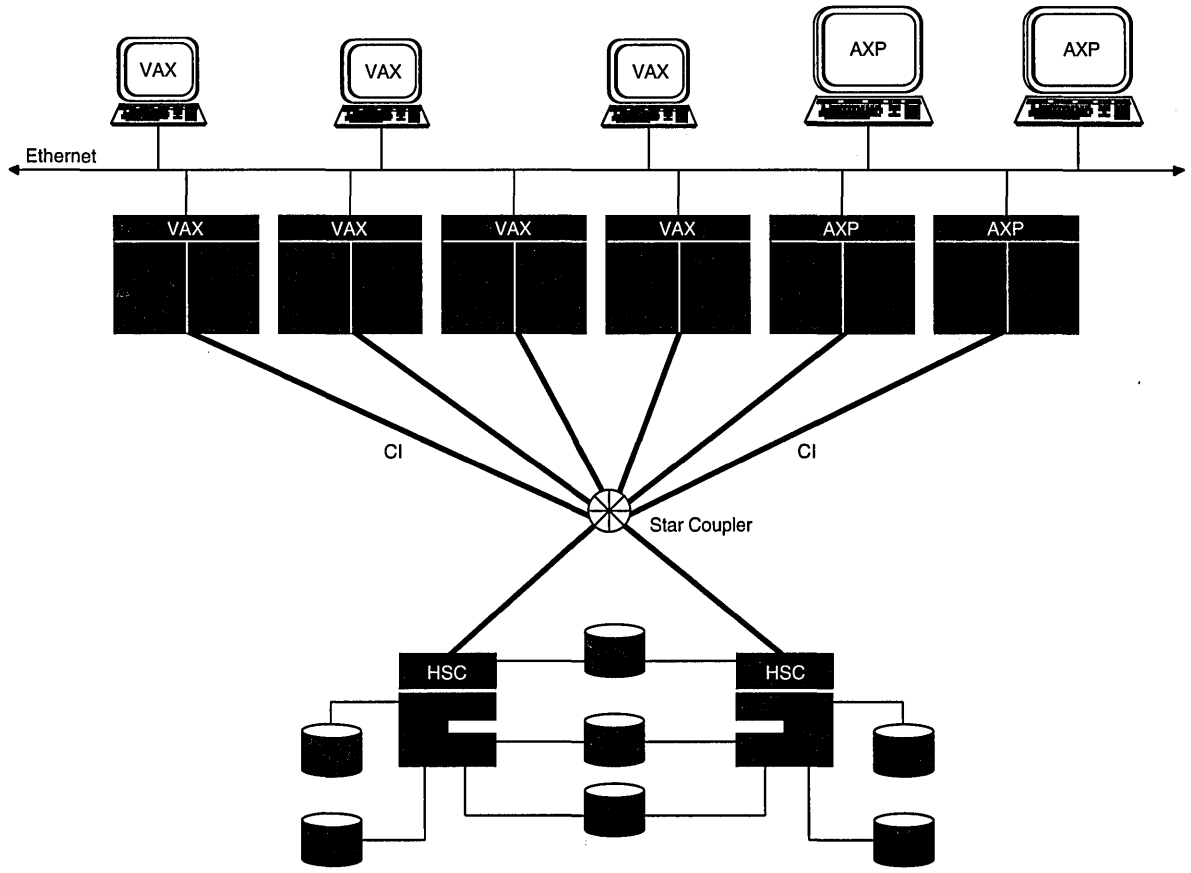
- A CI VMScluster system with satellite workstations on the LAN
- A DSSI VMScluster system with satellite workstations on the LAN
- A CI VMScluster system where one of the VAX or AXP nodes connects to another VAX or AXP node via the DSSI
- Any distributed LAN-based VMScluster system

(DSSI configurations are described in Section 2.3, and configurations that use FDDI are discussed in Section 2.6.)

CI connected computers can serve HSC disks to satellites by means of MSCP server software and drivers; therefore, the satellites can access the large amount of storage that is available through HSC subsystems.

VMScluster systems using a mixture of interconnects provide maximum flexibility in combining CPUs, storage, and workstations into highly available configurations. Figure 2–5 shows a VMScluster system using both CI and Ethernet interconnects.

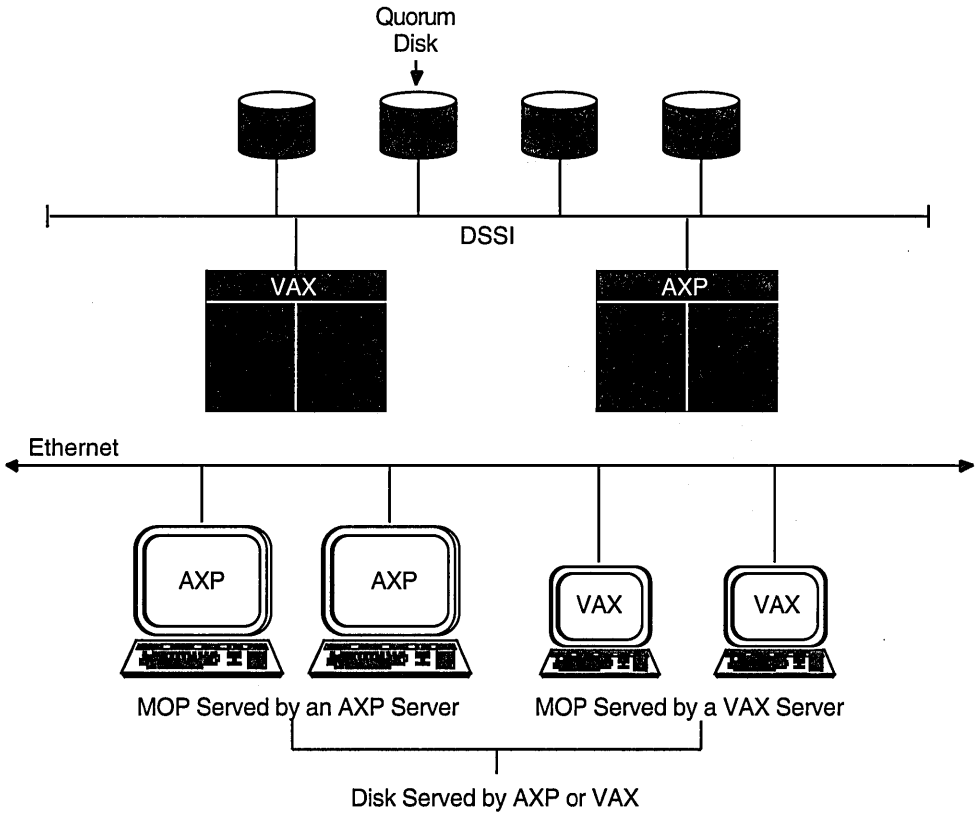
Figure 2–5 VMScluster System Using CI and Ethernet Interconnects



ZK–5946A–GE

Figure 2–6 shows a VMScluster system using both DSSI and Ethernet interconnects. Note that the DSSI configuration shown in Figure 2–6 includes a quorum disk that is fully shared by the AXP and VAX nodes to further increase availability. This configuration can tolerate the failure of either node or of the quorum disk and continue operating. See Section 3.1.1 for more information about the quorum scheme.

Figure 2–6 VMScluster SystemUsing DSSI and Ethernet Interconnects



2.6 FDDI VMScluster Systems

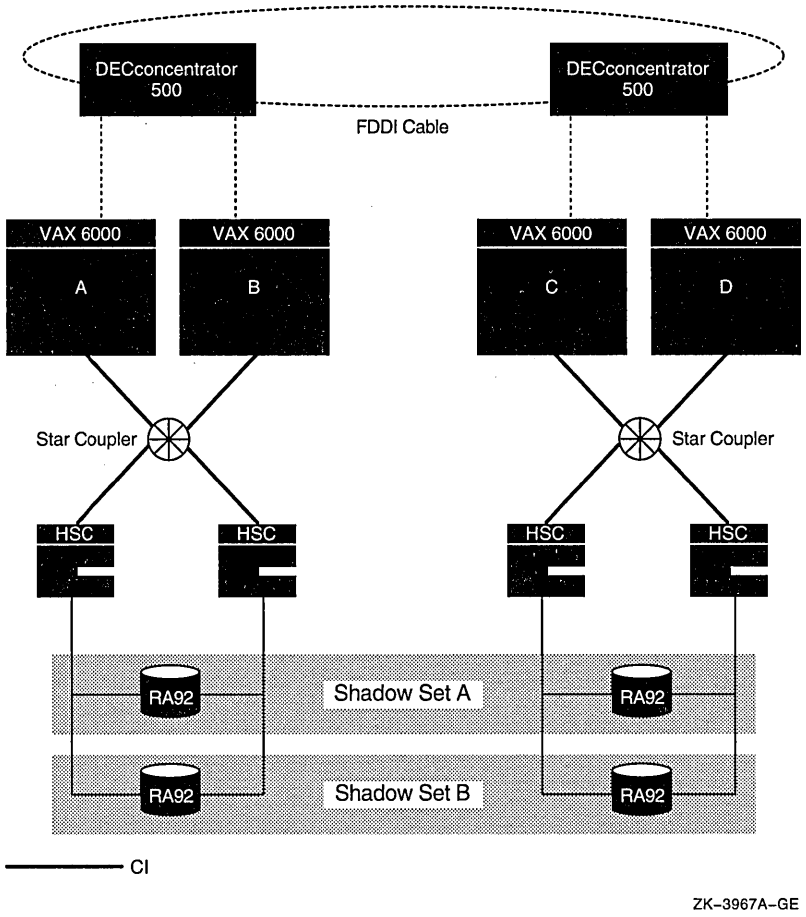
In VMScluster environments, the Fiber Distributed Data Interface (FDDI) is an interconnect that utilizes fiber-optic cables for very high bandwidth over long distances. FDDI uses a token-ring topology with two counterrotating rings. The available bandwidth depends on the target token rotation time (TTRT) and the time required to send the token around an idle ring (ring latency). The ring latency depends on the FDDI configuration. Providing alternative cluster communications paths will help cluster performance because the performance is impacted by increases in ring latency.

A **lobe** is a collection of VAX and/or AXP nodes connected by a CI and star coupler or connected by a DSSI. Such a collection could exist on its own as a VMScluster system, but when two or more lobes are connected by the FDDI, they become one large, multilobe VMScluster system.

2.6.1 Multiple Lobe FDDI VMScluster Systems

Figure 2–7 shows an example of a multilobe FDDI VMScluster system.

Figure 2–7 FDDI VMScluster System: Sample Configuration



The ellipse connecting the two DECconcentrator 500 units represents the dual-direction FDDI cables. Nodes A and B make up one lobe of the VMScluster. Nodes C and D make up a second lobe (perhaps located at a distance from the first lobe). The FDDI fiber cable connects both lobes into a single VMScluster system.

Volume Shadowing for OpenVMS is used to maintain key data storage devices in identical states (shadow sets A and B). Any sectors on the shadowed disks written at one site also will be written at the other site, and vice versa. Disks must be of the same type (for example, all RA92 disks or all RF73 disks).

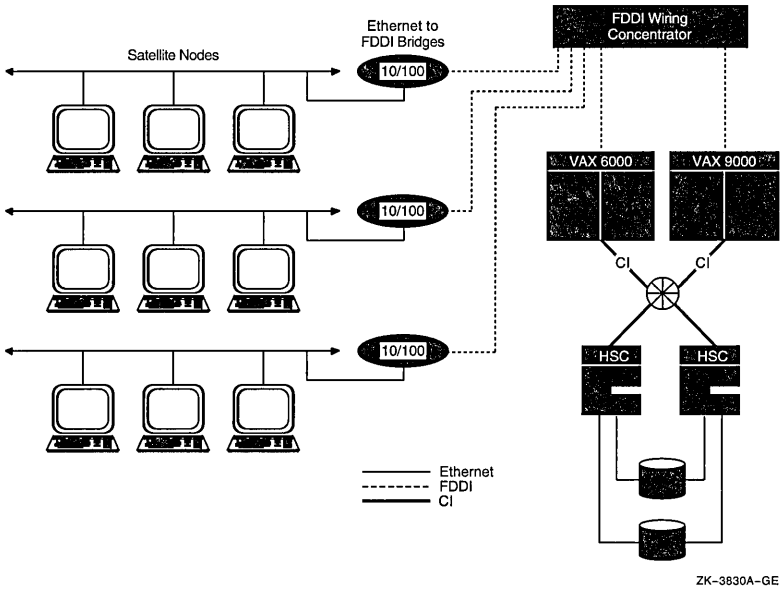
2.6.2 Network Configurations Using FDDI as VMScluster Interconnect

Fiber Distributed Data Interface (FDDI) fiber-optic cable can provide distinct advantages for networked client/server applications. Desktop environments can be integrated into VMScluster systems by using FDDI's higher bandwidth (tenfold speed improvement over Ethernet) and FDDI's ability to connect computing resources that are located up to 40 kilometers (25 miles) away.

Before FDDI, VMScluster systems that were used for desktop integration typically had several large processors acting as MOP and disk servers plus many satellite nodes (clients) taking advantage of the served resources. Ethernet is used in this configuration to connect the satellites to the servers. Typically, satellites (workstations) on multiple Ethernet segments are bridged (using multiple LAN Bridge 200 devices into a DELNI Ethernet wiring concentrator. From the DELNI concentrator, connections are made to the boot and disk server nodes of the CI VMScluster.

VMScluster systems that have heavily utilized Ethernet segments can replace the Ethernet backbone with FDDI to alleviate the Ethernet as a performance bottleneck. As shown in Figure 2–8, FDDI can replace the Ethernet from the bridges to the server CPU nodes. This configuration can increase overall throughput.

Figure 2–8 FDDI in Conjunction with Ethernet in a VMScluster System



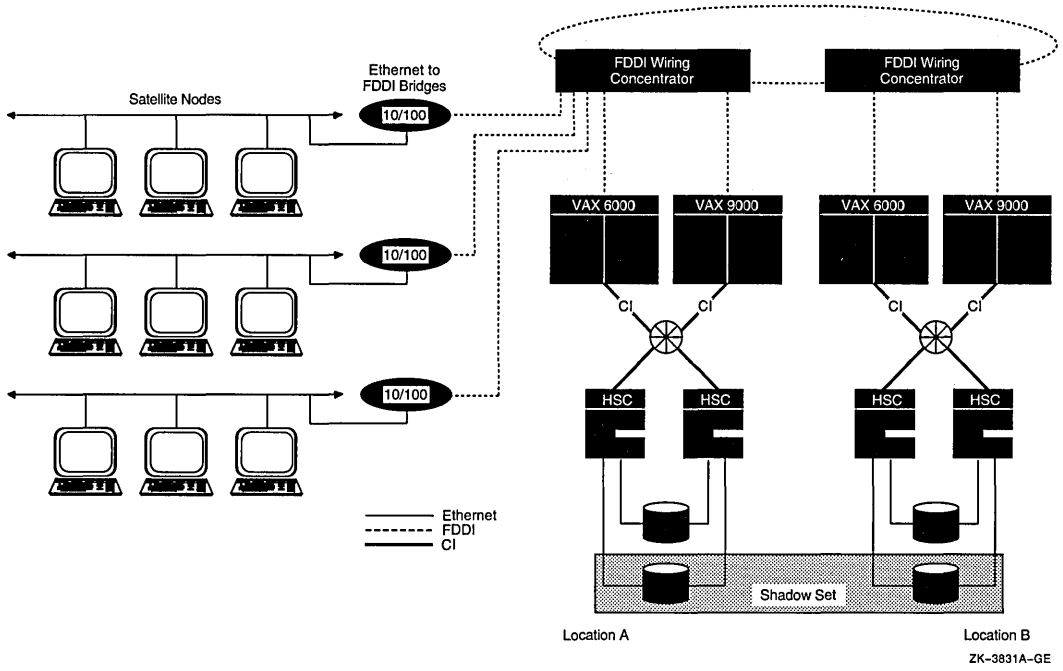
The longer distances provided by FDDI might permit you to create new VMScluster systems in your computing environment. For example, another way of looking at Figure 2–8 is that this VMScluster system is new. The large nodes on the right could have replaced server nodes that previously existed on the individual Ethernet segments.

Currently, there are no storage controllers that connect directly to FDDI. CPU nodes connected to FDDI must have local storage or access to storage over another interconnect. In Figure 2–8, the VAX 6000 and VAX 9000 computers have CI connections for storage. If a VMScluster system has more than one FDDI-connected node, then those CPU nodes will probably use CI or DSSI connections for storage. The VAX 6000 and VAX 9000 computers, connected by CI in Figure 2–8, are considered a lobe of the VMScluster system.

For data center consolidation, FDDI can expand your data center to include additional resources while retaining a single system management domain. Computing resources and their associated storage that are today physically located outside your data center can now be tied in with FDDI; these resources could include standalone systems or another VMScluster system. When

connecting two or more existing VMScluster systems with FDDI, you must create a single new VMScluster system. The previously independent VMScluster systems can remain physically separated within a building or in several buildings. But the systems now logically form a single operating environment, as depicted in Figure 2–9.

Figure 2–9 VMScluster System Consisting of Multiple Data Centers



The multilobe VMScluster configuration shown in Figure 2–9 builds on the configuration in Figure 2–8, in which one CI VMScluster system serves a large number of satellites. The VAX 6000 and VAX 9000 pair at location A and their storage connected by CI comprise one lobe. If more compute power or storage resources are required, additional systems can be connected through FDDI.

2.7 Configuring Multiple CI Adapters

The operating system allows the application of multiple CI interfaces for each CPU and multiple star couplers in a VMScluster system. These VMScluster configurations have many times the data throughput capacity of configurations with a single star coupler.

2.8 Configuring Multiple DSSI Adapters

The operating system allows the application of multiple DSSI interfaces for a CPU, making possible VMScluster systems with many times the data throughput capacity of current VMScluster systems that are configured with a single DSSI adapter.

2.9 Configuring Multiple LAN Adapters

Multiple local area network (LAN) adapters are supported on each local area or mixed-interconnect VMScluster node. Local area VMScluster support for multiple adapters allows PEDRIVER (the port driver for the LAN) to establish more than one channel between the local and remote cluster nodes. A **channel** is a unique network path between two nodes that is represented by a pair of LAN addresses or LAN adapters.

Characteristics of Local Area VMScluster Systems With Multiadapter Support

Local area VMScluster systems with multiadapter support have the following characteristics:

- At boot time, *all* Ethernet and FDDI adapters are automatically configured for local area VMScluster use.
- PEDRIVER automatically detects and creates a new channel between the local node and each remote cluster node for each unique pair of LAN adapters.
- Channel viability is monitored continuously.
- Channel failure does not interfere with node-to-node (virtual circuit) communications as long as there is at least one remaining functioning channel between the nodes.

Requirements for Local Area VMScluster Systems with Multiadapter Support

Configurations for multiadapter local area VMScluster systems must meet the following requirements:

- The MOP server and the system disk server for a given satellite must be connected to the same LAN segment.
- All nodes must have a direct path to all other nodes. A direct path can be a bridged or nonbridged LAN segment.

For each node, DECnet and MOP serving (AXP or VAX, as appropriate) can be performed by only one adapter per extended LAN to prevent LAN address duplication.

The following guidelines are for configuring local area VMScluster systems with multiple LAN adapters. If you configure these systems according to the guidelines, server nodes (nodes serving disks, tape, and lock traffic) can typically use some of the additional bandwidth provided by the added LAN adapters and increase the overall performance of the cluster. However, the performance increase depends on the configuration of your cluster (see Appendix I) and the applications it supports.

Recommendations for Configuring Local Area VMScluster Systems with Multiadapter Support

Configurations with multiadapter local area VMScluster systems should follow these guidelines:

- For VMScluster configurations consisting of AXP nodes and VAX nodes, the VAX computers must run OpenVMS AXP Version 5.5–2.

VAX

- For VAXcluster configurations, the VAX computers must run a minimum of OpenVMS VAX Version 5.5. ♦
- Connect each LAN adapter to a separate LAN segment. A LAN segment can be bridged or nonbridged. Doing this can help provide higher performance and availability in the cluster. The LAN segments can be either Ethernet segments or FDDI rings.
- Distribute satellites equally among the LAN segments. Doing this can help to distribute the cluster load more equally across all of the LAN segments.
- LAN adapters providing MOP service (to AXP or VAX computers, as appropriate) should be distributed among the LAN segments to ensure that LAN failures do not prevent satellite booting.
- For the number of LAN adapters supported per node, refer to the VAXcluster SPD or the VMScluster SPD, as appropriate.

Guidelines for Configuring Highly Available Local Area VMScluster Systems

The following guidelines are for configuring the local area VMScluster to be highly available:

- Bridge cluster LAN segments together to form a single extended LAN.
- Provide redundant LAN segment bridges for failover support.
- Configure LAN bridges to pass the local area VMScluster and MOP multicast messages. You can use any of the following to build the configuration:
  - LAN bridge configuration documentation
  - RBMS
  - DECelms
  - DECmcc

  Refer to the documentation for your LAN bridge and to the documentation for RBMS, DECelms, or DECmcc for more information about configuring LAN bridges to pass these multicast messages.
- Use the Local Area VMScluster Network Failure Analysis program to monitor and maintain network availability (see Section E.1.3).
- Use the troubleshooting suggestions in Appendix G to diagnose performance problems with the SCS layer and the NISCA transport protocol.

2.9.1 Selecting MOP Servers for Availability

When using multiple LAN adapters with multiple LAN segments, you should distribute the connections to LAN segments that provide DECnet and MOP (AXP or VAX, as appropriate) service. The distribution allows MOP servers to downline load satellites even when network component failures occur.

Dual-architecture VMScluster systems complicate this configuration because VAX satellites must be downline loaded via VAX MOP servers, and AXP satellites must be downline loaded via AXP MOP servers.

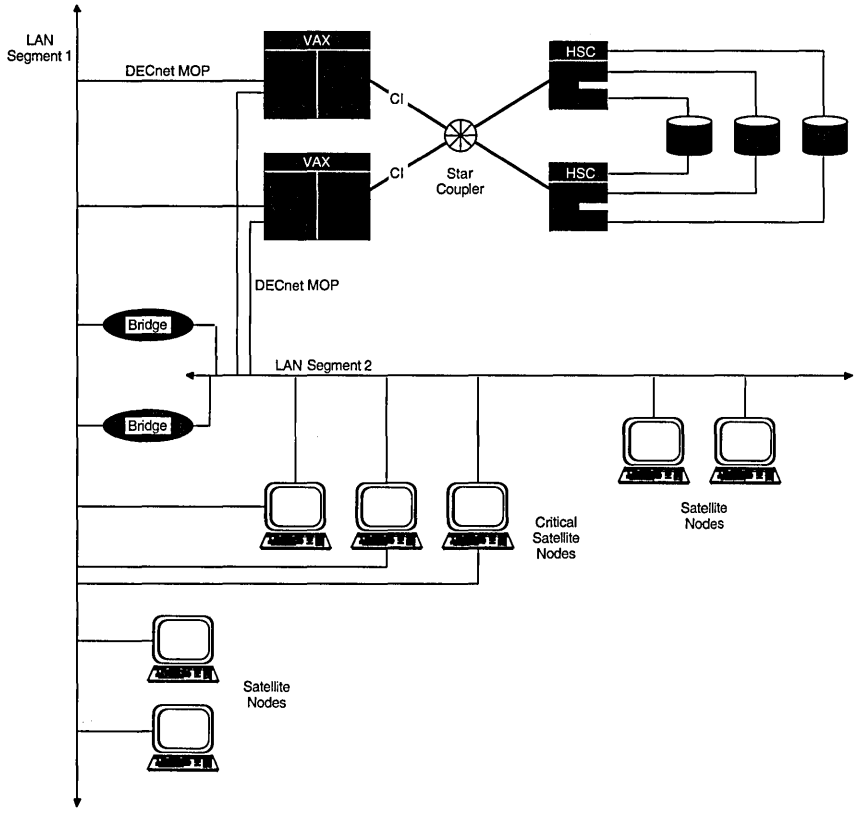
It is important to ensure sufficient MOP servers for both VAX and AXP nodes to provide downline load support for booting satellites. By careful selection of the MOP LAN connection for each MOP server (AXP or VAX, as appropriate) on the network, you can maintain MOP service in the face of network failures.

2.9.2 VMScluster with Two LAN Segments

Figure 2–10 shows a sample configuration for a local area VMScluster system connected to two different LAN segments. The figure illustrates the following points:

- Connecting critical nodes to multiple LAN segments provides increased availability in the event of segment or adapter failure. Disk and tape servers can use some of the network bandwidth provided by the additional network connection. Critical satellites can be booted using the other LAN adapter if one LAN adapter fails.
- Connecting noncritical satellites to only one LAN segment helps to balance the network load by distributing systems equally among the LAN segments. These systems communicate with satellites on the other LAN segment through one of the bridges.
- Only one LAN adapter per node can be used for DECnet and MOP service to prevent duplication of LAN addresses.
- LAN adapters providing MOP service (AXP or VAX, as appropriate) should be distributed among the LAN segments to ensure that LAN failures do not prevent satellite booting.
- Using redundant LAN bridges prevents the bridge from being a single point of failure.

Figure 2–10 Sample Two-LAN Segment VMScluster Configuration



For descriptions of sample local area VMScluster network connections, see Appendix D.

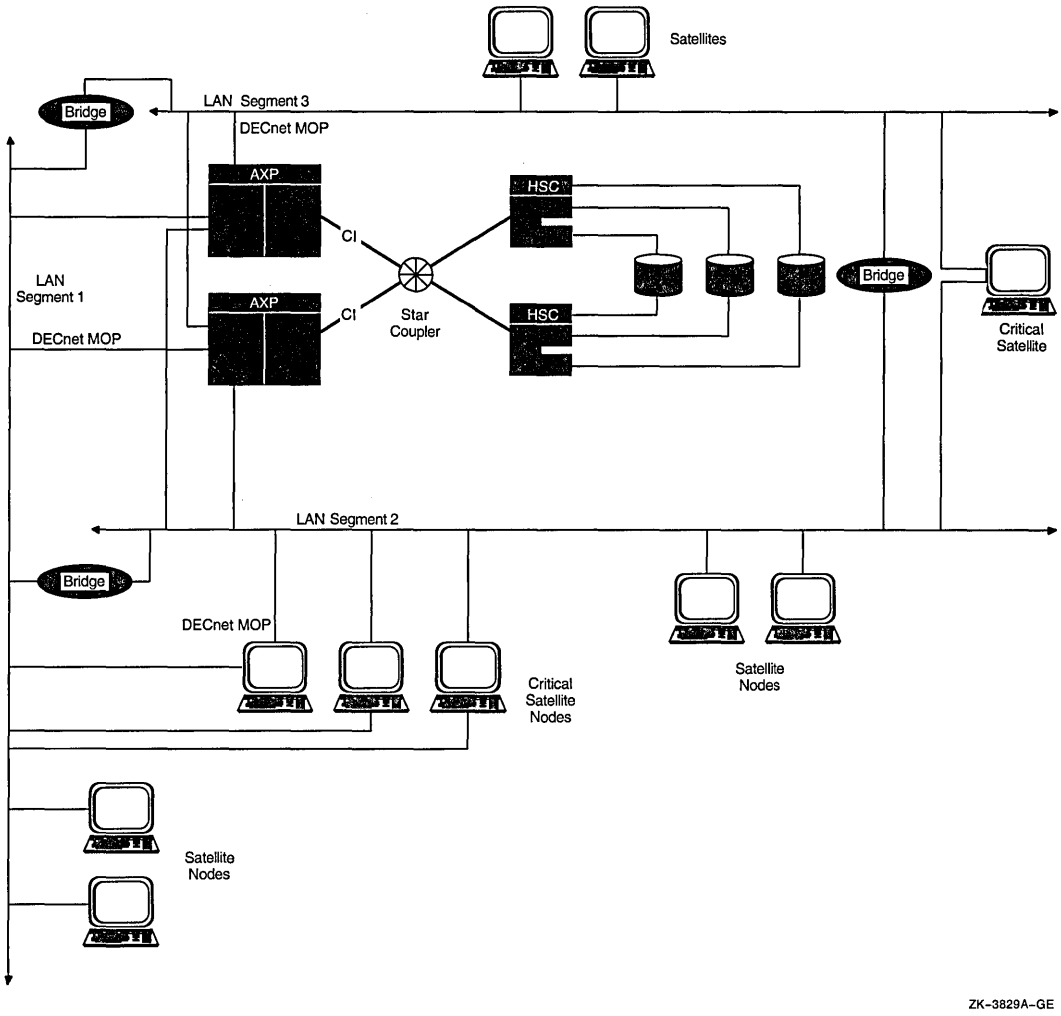
2.9.3 VMScluster with Three LAN Segments

Figure 2–11 shows a sample configuration for a local area VMScluster system connected to three different LAN segments. The figure illustrates the following points:

- Connecting disk and tape servers to two or three LAN segments can help provide higher availability and better I/O throughput.
- Connecting critical satellites to two or more LAN segments can also increase availability. If any of the network components fail, these satellites can use the other LAN adapters to boot and still have access to the critical disk servers.
- Distributing noncritical satellites equally among the LAN segments can help balance the network load.
- A MOP server (AXP or VAX, as appropriate) is provided for each LAN segment.

For descriptions of sample local area VMScluster network connections, see Appendix D.

Figure 2–11 Sample Three-LAN Segment VMScluster Configuration



2.9.4 Guidelines to Allow for LAN Bridge Failover

To achieve high availability, Digital recommends redundant bridges between LAN segments. If one bridge fails, another bridge can support message traffic between the LAN segments. To help ensure that there is little delay between the failure of one bridge and the continuing of message traffic support by another bridge, you should make sure the local area VMScluster follows these guidelines:

- Bridge timers should be set to be faster than VMScluster timers.
- Bridge self-test times should be less than the value for the system parameter RECNXINTERVAL (described in Appendix A).

The time required for one LAN bridge to fail over to another LAN bridge must adhere to the following parameters:

- Root bridge parameters:
  - LISTEN_TIME (default = 15 seconds)
  - FORWARDING_DELAY (default = 30 seconds)
- Bridge self-test time

If the VMScluster is to survive a failover between LAN bridges, you must specify a value for the system parameter RECNXINTERVAL that is greater than the following:

```
Bridge self-test time +
LISTEN_TIME (of the root bridge) +
FORWARDING_DELAY (of the root bridge)
```

The default value of RECNXINTERVAL is 20. For a value that is larger than the default failover time of a LAN bridge, use a value greater than 120. Note that the value of RECNXINTERVAL can be rounded up based on port timers.

A LAN bridge receives a HELLO datagram message from the other bridges in an interval specified by the root bridge parameter HELLO_INTERVAL. If the LAN bridge does not receive the HELLO datagram message within the time specified by the bridge parameter LISTEN_TIME, then the bridge changes the topology and possibly chooses a new root bridge. After the topology change, the bridge delays for the number of seconds specified by the value in the bridge parameter FORWARDING_DELAY. This delay allows the LAN bridge to learn which LAN addresses are on either side of it. The bridge then starts to forward packets.

Decreasing the LISTEN_TIME value allows the bridge to detect topology changes more quickly. If you reduce the LISTEN_TIME parameter value, you should also decrease the value for the HELLO_INTERVAL bridge parameter. However, note that decreasing the value for the HELLO_INTERVAL parameter causes the bridge to send the HELLO datagram messages more frequently and thus increases LAN traffic.

Decreasing the FORWARDING_DELAY value can cause the bridge to forward packets unnecessarily to the other LAN segment. Unnecessary forwarding can temporarily cause more traffic on both LAN segments until the bridge software determines which LAN address is on each side of the bridge.

If you change a parameter on one LAN bridge, you should change that parameter on all bridges to ensure that selection of a new root bridge does not change the value of the parameter. The actual parameter value the bridge uses is the value specified for the root bridge.

2.9.5 Adjusting Maximum Packet Size for FDDI Configurations

For computers running OpenVMS VAX Version 6.0 or OpenVMS AXP Version 1.5, use the NISCS_MAX_PKTSZ system parameter to specify the maximum packet size for VMScluster packets on FDDI. The value of this parameter should be set to either 1498 or 4468.

VAX

For computers running VMS Version 5.5–2 and earlier, FDDI supports transfers using large packets (up to 4468 bytes). PEDRIVER does not use large packets by default, but can take advantage of the larger packet sizes if you increase the LRPSIZE system parameter to 4474 or higher. The LRPSIZE system parameter (described in Appendix A) specifies the size of the large request packets.

PEDRIVER uses the full FDDI packet size if the LRPSIZE is set to 4474 or higher. However, only FDDI nodes connected to the same ring use large packets. Nodes connected to an Ethernet segment restrict packet size to that of an Ethernet packet (1498 bytes). ♦

If your local area VMScluster configuration can take advantage of large FDDI packets, you should increase packet size to 4474 bytes on nodes that have FDDI adapters on a common FDDI ring. Note that PEDRIVER does not combine multiple messages into a single packet. Therefore, increasing the maximum

packet size to use large packets is beneficial only if the work load includes large data transfers.

Configurations that mix CI or DSSI with FDDI for cluster communication typically cannot benefit from using the large packets on the FDDI. Systems in these configurations choose the CI or DSSI path over the FDDI path. Increasing the packet size for FDDI for these mixed configurations increases the demand for physical memory (nonpaged pool) with no corresponding gain.

PEDRIVER detects that a message has traveled across an Ethernet segment by testing the value of the priority field in the frame control byte of the FDDI header (see Section G.4.1.2 for a description of the FDDI header). Each bridge should set the priority field value to 0. (Bridges manufactured by Digital set this value to 0.) PEDRIVER sends all FDDI messages with a nonzero priority field value. If a bridge does not set the priority field value to 0, PEDRIVER logs port errors on the console, and the virtual circuit breaks connections to other FDDI nodes. If this condition occurs, reduce the FDDI packet size to the default value by setting NISCS_MAX_PKTSZ to 1498 or LRPSIZE to 1504 to prevent use of large packets, and reboot the FDDI nodes in the cluster.

If you decide to change the value of the LRPSIZE or NISCS_MAX_PKTSZ parameter, edit the SYS$SPECIFIC:[SYSEXE]MODPARAMS.DAT file to permit AUTOGEN to factor the changed packet size into its calculations.

3

VMScluster Integrity and Security

This chapter describes how to ensure the integrity and security of nodes that make up the VMScluster membership and of the cluster shareable objects.

3.1 Connection Management

The integrity of a VMScluster system is controlled by a software component called the **connection manager**, which determines and coordinates membership in the cluster. The connection manager creates a cluster when the first active computers are booted and then reconfigures the cluster when computers join or leave it.

Computers in a VMScluster system share various data and system resources, such as disk volumes. To achieve the coordination that is necessary to maintain resource integrity, the computers must share a clear sense of cluster membership, which is maintained by the connection manager.

Within a single VMScluster system, the 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**.

3.1.1 The Quorum Scheme

The quorum scheme is based on the arithmetic principle that the whole cannot be divided into multiple parts in such a way that more than one part is greater than half of the whole. (Integer arithmetic is used in this section.)

The quorum scheme functions as follows:

- In a VMScluster system, each **voting member** (AXP or VAX computers with a nonzero value for the system parameter VOTES) contributes a fixed number of votes toward quorum. On satellites, the VOTES value is always set to 0 by default.
- Each active AXP or VAX computer in the cluster (including satellites) indirectly specifies an initial quorum value using the EXPECTED_VOTES system parameter. This parameter is the sum of all votes held by potential cluster members. It is used to derive an estimate of the correct quorum value for the cluster, according to the following formula:

  ```
  Estimated quorum = (EXPECTED_VOTES + 2)/2
  ```

- During certain cluster state transitions, the system dynamically computes the cluster quorum to be the *maximum* of the following:
  - The current cluster quorum value

- The largest of the values calculated from the following formula, where EV is the EXPECTED_VOTES value specified by each computer:

  ```
  (EV+2)/2
  ```

- The value calculated from the following formula, where V is the total of the VOTES system parameter held by all cluster members:

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

The cluster state transitions that cause cluster quorum to be recalculated occur when a computer joins the cluster and when the cluster recognizes a quorum disk. (The role of the quorum disk is discussed in Section 3.1.2.)

- If the current number of votes ever drops below quorum (because of computers leaving the cluster), the remaining cluster members suspend all process activity and all I/O operations to cluster-accessible disks and tapes until sufficient votes are added (that is, enough computers have joined the cluster) to bring the total number of votes to a value greater than or equal to quorum.
- As the cluster configuration changes, the cluster software increases the cluster quorum value; it never decreases the value. (However, system managers can decrease the value; for details, see Section 7.7.5.)

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

3.1.2 Quorum Disk

A quorum disk acts as a virtual computer, adding to the total cluster votes. By establishing a quorum disk in configurations with a small number of voting computers, you can increase the availability of the cluster. Such configurations can tolerate the failure either of the quorum disk or of a computer and continue operating.

Note

Each VMScluster system can include only one quorum disk.

To use a quorum disk, one or more computers must have a direct (non MSCP served) connection to the disk. Such computers are known as **quorum disk watchers**. Computers that cannot access the disk directly rely on the quorum disk watchers for information about the status of votes contributed by the quorum disk.

You should enable as quorum disk watchers any computers that have a direct, active connection to the quorum disk or that have the potential for a direct connection. To enable a computer as a quorum disk watcher, use the CLUSTER_CONFIG.COM CHANGE function described in Section 7.5.3. The procedure prompts for the name of the quorum disk and specifies that name as a value for the DISK_QUORUM system parameter in MODPARAMS.DAT. The procedure also sets an appropriate value for the QDSKVOTES parameter. The number of votes contributed by the quorum disk is equal to the smallest value of the QDSKVOTES system parameter on any quorum disk watcher.

Note

You can also enable the first installed cluster computer as a quorum disk watcher by answering YES when the installation procedure asks whether the cluster will contain a quorum disk.

For the quorum disk's votes to be counted in the total cluster votes, the following conditions must be met:

- On one or more computers capable of becoming watchers, you must specify the same *physical* device name as a value for the DISK_QUORUM system parameter. The remaining computers (which must have a blank value for DISK_QUORUM) recognize the name specified by the first quorum disk watcher with which they communicate.
- At least one quorum disk watcher must have a direct, active connection to the quorum disk. Thus, the quorum disk may be a dual-ported DSA disk, which has a direct, active connection to only one computer at a time.
- The disk must contain a valid format file named QUORUM.DAT in the master file directory (MFD). The QUORUM.DAT file is created automatically after a system specifying a quorum disk has booted into the cluster for the first time. This file is used on subsequent reboots. Note that the file is not created if the system parameter STARTUP_P1 is set to MIN.
- To permit recovery from failure conditions, the quorum disk must be mounted by all disk watchers.
- The VMScluster can include only one quorum disk.
- The quorum disk cannot be a member of a shadow set.

Note that by increasing the quorum disk's votes to one less than the total votes from all systems (and by increasing the value of the EXPECTED_VOTES system parameter by the same amount), you can boot and run the cluster with only one node. This prevents having to wait until more than half of the voting systems are up before you can start using the VMScluster system.

3.1.3 State Transitions

VMScluster state transitions occur when a computer joins or leaves a VMScluster system. The connection manager controls these events to ensure the preservation of data integrity throughout the cluster. A state transition's duration and effect on users (applications) are determined by the reason for the transition, the configuration, and the applications in use.

Every transition goes through one or more phases, depending on whether its cause is the addition of a new VMScluster member or the failure of a current member. If the transition is caused by the addition of a new member, the phases are as follows:

- New member detection

  Early in its boot sequence, a computer seeking membership in a VMScluster system sends messages to current members asking to join the cluster. The first cluster member that receives the membership request acts as the new computer's advocate and proposes reconfiguring the cluster to include the computer in the cluster. While the new computer is booting, no applications are affected.

- Reconfiguration

  All current VMScluster members must establish communications with the new computer. Once communications are established, the new computer is admitted to the cluster. In some cases, the lock database is rebuilt.

If the transition is caused by the failure of a current VMScluster member, the phases are as follows:

- Failure detection

  The duration of this phase depends on the cause of the failure and on how the failure is detected.

  During normal cluster operation, messages sent from one computer to another are acknowledged when received. If a message is not acknowledged within a period determined by VMScluster communications software, the repair attempt phase begins.

  If a cluster member is shut down or fails, the operating system causes datagrams to be sent from the computer shutting down to the other members. These datagrams state the computer's intention to sever communications and to stop sharing resources. Because sending these datagrams is virtually the last activity of a "dying" computer, they are called "last gasp" datagrams. If any current cluster member receives a last-gasp datagram, the "gasping" computer is removed from the cluster. The failure detection and repair attempt phases are bypassed, and the reconfiguration phase begins immediately.

- Repair attempt

  If the communication path to a VMScluster member is broken, attempts are made to repair the path. Repair attempts continue for an interval specified by the RECNXINTERVAL system parameter. (System managers can adjust the value of this parameter to suit local conditions.) Thereafter, the path is considered irrevocably broken, and steps must be taken to reconfigure the VMScluster system so that all computers can once again communicate with each other and so that computers that cannot communicate are removed from the cluster.

- Reconfiguration

  When a VMScluster member fails, the cluster must be reconfigured. One of the remaining computers acts as coordinator and exchanges messages with all other cluster members to determine the configuration of an **optimal subcluster** with the most members and the most votes. This phase, during which all user (application) activity is blocked, usually lasts less than 1 second.

- VMScluster system recovery

  Recovery includes the following stages, some of which can take place in parallel:

  - I/O completion

    When a computer is removed from the cluster, VMScluster software ensures that all I/O operations that are started by the old configuration complete before I/O operations that are generated by the new configuration start. This stage usually has little or no effect on applications.

- Lock database rebuild

  Because the lock database is distributed among all members, some portion of the database might need rebuilding. A rebuild is always performed when a computer leaves the cluster, but only in certain cases when a computer is added.

- Disk mount verification

  This stage occurs only when the failure of a voting member causes quorum to be lost. To protect data integrity, all I/O activity is blocked until quorum is regained. Mount verification is the mechanism used for this purpose.

- Quorum votes validation

  If, when a computer is removed, the remaining members can determine that it has shut down or failed, the votes contributed by the quorum disk are included without delay in quorum calculations that are performed by the remaining members. However, if they cannot determine that the computer has shut down or failed (for example, if a console halt, power failure, or communications failure has occurred), the votes are not included for a period (in seconds) equal to four times the value of QDSKINTERVAL. This period is sufficient to determine that the failed computer is no longer using the quorum disk.

- Disk rebuild

  If the transition is the result of a computer rebooting after a failure, the disks are marked as improperly dismounted, and they must be rebuilt before they can be remounted. The rebuild reclaims space that was cached by the failed computer but never returned, as with a normal dismount. A rebuild makes the disk briefly inaccessible to users unless the rebuild is deferred by using the /NOREBUILD qualifier to the MOUNT commands in the system startup files. (See Section 5.5 for more information about rebuilding disks.)

- Application recovery

  When assessing the effect of a state transition on application users, consider that the application recovery phase includes activities such as replaying a journal file, cleaning up recovery units, and users logging in again because the terminal server has failed over to another computer.

  VMScluster troubleshooting information appears in Appendix C.

3.1.3.1 Managing VMScluster Membership

VMScluster systems use a **cluster group number** and a **cluster password** to allow multiple independent VMScluster systems to coexist on the same extended LAN and to prevent accidental access to a cluster by unauthorized computers.

- The cluster group number uniquely identifies each VMScluster system on a LAN. This number must be from 1 to 4095 or from 61440 to 65535. Note that if you plan to have more than one of these clusters on a LAN, you must coordinate the assignment of cluster group numbers among system managers.
- The cluster password serves as an additional check to ensure the integrity of individual clusters on the same LAN that accidentally use identical cluster group numbers. If each cluster's password is unique, the clusters will form independently. However, if any passwords are identical, errors are generated and listed as error log entries.

The password also prevents an intruder who discovers the cluster group number from joining the cluster. The password must be from 1 to 31 alphanumeric characters in length, including alphanumeric characters, dollar signs ($), and underscores (_).

If all nodes in the VMScluster do not have the same cluster group number and password, error messages are logged in the error log file.

The cluster group number and password are maintained in the cluster authorization file, SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT. This file is created during installation of the operating system if you indicate that you want to set up a local area or mixed-interconnect cluster. The installation procedure then prompts you for the cluster group number and password. Maintaining the integrity of VMScluster membership is described in detail in Section 7.7.8. (If you convert a CI VMScluster to a mixed-interconnect configuration, the file is created when you execute the CLUSTER_CONFIG.COM command procedure, as described in Chapter 7.)

3.2 VMScluster Systems Require a Single Security Domain

VMScluster systems provide a uniform computing environment that is highly scalable, highly available, and secure. A key feature of a VMScluster system is the concept of a single security domain in which individual nodes use a common set of authorizations to mediate access control. The OpenVMS security subsystem ensures that all authorization information and object security profiles are consistent across all nodes in the cluster. In effect, a single security domain ensures that a security check results in the same answer from any node in the cluster.

In a single security environment, authorized users can have processes executing on any VMScluster member. A process, acting on behalf of an authorized individual, requests access to a cluster object and a coordinating node determines the outcome by comparing its copy of the common authorization database with the security profile for the object being accessed. The OpenVMS operating system provides this level of protection for files and queues and further incorporates all other cluster-visible objects, such as devices, volumes, and lock resource domains.

The OpenVMS operating system cannot enforce a level of separation needed to support different security domains on separate cluster members. Therefore, the OpenVMS VAX and OpenVMS AXP operating systems do not support multiple security domains.

Actions of the cluster manager in setting up a VMScluster system can affect the security operations of the system. You can facilitate VMScluster security management using the suggestions discussed in the following sections.

3.2.1 Building a Single Security Domain

The easiest way to ensure a single security domain is to maintain a single copy of each of the following files on one or more disks that are accessible from anywhere in the VMScluster system. When a cluster is configured with multiple system disks, you can use system logical names to ensure that only a single copy of each file exists. The OpenVMS security domain consists of the following files:

SYS$MANAGER:AUDIT_SERVER.DAT (OpenVMS AXP Version 1.5 and VMS Version 5.5–2 and earlier)
SYS$MANAGER:VMS$AUDIT_SERVER.DAT (OpenVMS VAX Version 6.0 only)
SYS$SYSTEM:NETOBJECT.DAT

SYS$SYSTEM:NETPROXY.DAT
SYS$SYSTEM:QMAN$MASTER.DAT
SYS$SYSTEM:RIGHTSLIST.DAT
SYS$SYSTEM:SYSALF.DAT
SYS$SYSTEM:SYSUAF.DAT
SYS$SYSTEM:SYSUAFALT.DAT
SYS$SYSTEM:VMS$OBJECTS.DAT (OpenVMS VAX Version 6.0 only)
SYS$SYSTEM:VMS$PASSWORD_HISTORY.DATA
SYS$SYSTEM:VMSMAIL_PROFILE.DATA
SYS$LIBRARY:VMS$PASSWORD_DICTIONARY.DATA
SYS$LIBRARY:VMS$PASSWORD_POLICY.EXE

Using shared files is not the only way of achieving a single security domain. You may need to use multiple copies of one or more of these files on different nodes in a cluster. For example, on AXP nodes you may choose to deploy system-specific user authorization files (SYSUAFs) to allow for different memory management working-set quotas among different nodes. Such configurations are fully supported as long as the security information available to each node in the cluster is identical.

The remainder of this section describes the security-relevant portions of the files that must be synchronized across all cluster members to ensure that a single security domain exists. In the following list, files noted as "required" contain some data that must be kept synchronized. Files noted as "recommended" contain data that should be synchronized at the discretion of the site-security administrator or system manager. Nonetheless, Digital recommends that you synchronize the recommended files.

Be aware that some of these files are created only on request and may not exist in all configurations. A file can be absent on one node only if it is absent on all nodes. As soon as a required file is created on one node, it must be created or commonly referenced on all remaining cluster members.

- VMS$AUDIT_SERVER.DAT (and AUDIT_SERVER.DAT) [recommended]

  These files contain information related to security auditing. Among the information contained is the list of enabled security auditing events and the destination of the system security audit log file. When more than one version of this file exists, all copies should be updated after any SET AUDIT command. Failure to synchronize multiple versions of this file properly may result in partitioned auditing domains. See Section 3.2.2 for more information about security audit files.

- NETOBJECT.DAT [required]

  This file contains the DECnet object database. Among the information contained in this file is the list of known DECnet server accounts and passwords. When more than one version of this file exists, all copies must be updated after any NCP {SET | DEFINE} OBJECT command. Failure to synchronize multiple versions of this file properly may result in unexplained network login failures and unauthorized network access.

- NETPROXY.DAT [required]

  This file contains the network proxy database. It is maintained by the OpenVMS Authorize utility. When more than one version of this file exists, all copies must be updated after any UAF proxy command. Failure to synchronize multiple versions of this file properly may result in unexplained network login failures and unauthorized network access.

- QMAN$MASTER.DAT [required]

  This file contains the master queue manager database. This file contains the security information for all shared batch and print queues. If two or more nodes are to participate in a shared queuing system, a single copy of this file must be maintained on a shared disk.

- RIGHTSLIST.DAT [required]

  This file contains the rights identifier database. It is maintained by the OpenVMS Authorize utility and by various rights identifier system services. When more than one version of this file exists, all copies must be updated after any change to any identifier or holder records. Failure to synchronize multiple versions of this file properly may result in unauthorized system access and unauthorized access to protected objects.

- SYSALF.DAT [required]

  This file contains the system Autologin facility database. It is maintained by the OpenVMS SYSMAN utility. When more than one version of this file exists, all copies must be updated after any SYSMAN ALF command. Failure to synchronize multiple versions of this file properly may result in unexplained login failures and unauthorized system access.

- SYSUAF.DAT [required]

  This file contains the system user authorization file. It is maintained by the OpenVMS Authorize utility and is modifiable via the $SETUAI system service. When more than one version of this file exists, you must ensure that the fields shown in Table 3–1 are synchronized for each user record.

Table 3–1 Fields in SYSUAF and Associated $SETUAI Item Codes

Internal Name	$SETUAI Item Code
UAF$R_DEF_CLASS	UAI$_DEF_CLASS
UAF$Q_DEF_PRIV	UAI$_DEF_PRIV
UAF$B_DIALUP_ACCESS_P	UAI$_DIALUP_ACCESS_P
UAF$B_DIALUP_ACCESS_S	UAI$_DIALUP_ACCESS_S
UAF$B_ENCRYPT	UAI$_ENCRYPT
UAF$B_ENCRYPT2	UAI$_ENCRYPT2
UAF$Q_EXPIRATION	UAI$_EXPIRATION
UAF$L_FLAGS	UAI$_FLAGS
UAF$B_LOCAL_ACCESS_P	UAI$_LOCAL_ACCESS_P
UAF$B_LOCAL_ACCESS_S	UAI$_LOCAL_ACCESS_S
UAF$B_NETWORK_ACCESS_P	UAI$_NETWORK_ACCESS_P
UAF$B_NETWORK_ACCESS_S	UAI$_NETWORK_ACCESS_S
UAF$B_PRIME_DAYS	UAI$_PRIMEDAYS
UAF$Q_PRIV	UAI$_PRIV
UAF$Q_PWD	UAI$_PWD
UAF$Q_PWD2	UAI$_PWD2

(continued on next page)

Table 3–1 (Cont.) Fields in SYSUAF and Associated $SETUAI Item Codes

Internal Name	$SETUAI Item Code
UAF$Q_PWD_DATE	UAI$_PWD_DATE
UAF$Q_PWD2_DATE	UAI$_PWD2_DATE
UAF$B_PWD_LENGTH	UAI$_PWD_LENGTH
UAF$Q_PWD_LIFETIME	UAI$_PWD_LIFETIME
UAF$B_REMOTE_ACCESS_P	UAI$_REMOTE_ACCESS_P
UAF$B_REMOTE_ACCESS_S	UAI$_REMOTE_ACCESS_S
UAF$R_MAX_CLASS	UAI$_MAX_CLASS
UAF$R_MIN_CLASS	UAI$_MIN_CLASS
UAF$W_SALT	UAI$_SALT
UAF$L_UIC	Not applicable

Failure to synchronize multiple versions of the SYSUAF files properly may result in unexplained login failures and unauthorized system access.

Creation and management of the various elements of a VMScluster common SYSUAF.DAT authorization database is discussed in Appendix B. This appendix also discusses how to consolidate several RIGHTSLIST.DAT files.

Note that the default values for a number of SYSUAF process limits and quotas are higher on AXP computers than they are on VAX computers. In general, the values in a common SYSUAF.DAT file should accommodate the largest requirements in the cluster. Then, on nodes that require smaller quotas, edit the local MODPARAMS.DAT file to adjust the system parameters to more appropriate values. See also *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for help in determining process quotas AXP and VAX computers.

- SYSUAFALT.DAT [required]

  This file contains the system alternate user authorization file. This file serves as a backup to SYSUAF.DAT and is enabled via the SYSUAFALT system parameter. When more than one version of this file exists, all copies must be updated after any change to any authorization records in this file. Failure to properly synchronize multiple versions of this file may result in unexplained login failures and unauthorized system access.

VAX

- VMS$OBJECTS.DAT (VAX only) [required]

  On VAX systems, this file contains the clusterwide object database. Among the information contained in this file are the security profiles for all clusterwide objects. When more than one version of this file exists, all copies must be updated after any change to the security profile of a clusterwide object or after new clusterwide objects are created. Clusterwide objects include: disks, tapes, and resource domains. Failure to synchronize multiple versions of this file properly may result in unauthorized access to protected objects.♦

- VMS$PASSWORD_HISTORY.DATA [recommended]

  This file contains the system password history database. It is maintained by the system password change facility. When more than one version of this file exists, all copies should be updated after any password change. Failure to synchronize multiple versions of this file properly may result in a violation of the system password policy.

- VMSMAIL_PROFILE.DATA [recommended]

  This file contains the system mail database. This file is maintained by the OpenVMS Mail utility and contains mail profiles for all system users. Among the information contained in this file is the list of all mail forwarding addresses in use on the system. When more than one version of this file exists, all copies should be updated after any changes to mail forwarding. Failure to synchronize multiple versions of this file properly may result in unauthorized disclosure of information.

- VMS$PASSWORD_DICTIONARY.DATA [recommended]

  This file contains the system password dictionary. The system password dictionary is a list of English language words and phrases that are not legal for use as account passwords. When more than one version of this file exists, all copies should be updated after any site-specific additions. Failure to synchronize multiple versions of this file properly may result in a violation of the system password policy.

- VMS$PASSWORD_POLICY.EXE [recommended]

  This file contains any site-specific password filters. It is created and installed by the site-security administrator or system manager. When more than one version of this file exists, all copies should be identical. Failure to synchronize multiple versions of this file properly may result in a violation of the system password policy.

3.2.2 Naming the Auditing Log File

The audit server uses an audit server database to oversee security. The database can contain information about events to be audited, location of the security audit log file, event timers, and information used to monitor the consumption of system resources.

VMScluster system managers should ensure that the name assigned to the security audit log file resolves as follows:

VAX

On VAX systems, the default location of the auditing log file is SYS$COMMON:[SYSMGR]SECURITY.AUDIT$JOURNAL.♦

AXP

On AXP systems, the default location of the auditing log file is SYS$COMMON:[SYSMGR]SECURITY_AUDIT.AUDIT$JOURNAL.♦

If you need to relocate the audit log file somewhere other than the system disk (or if you have multiple system disks), it is important to redirect the audit log uniformly across all nodes in the cluster. Use the command SET AUDIT/JOURNAL=SECURITY/DESTINATION=*file-name*, and specify a file name that resolves to the same file throughout the cluster.

Changes are automatically made in the audit server database, SYS$COMMON:[SYSMGR]AUDIT_SERVER.DAT. This database also identifies which events are enabled and how to monitor the audit system's use of resources, and restores audit system settings each time the system is rebooted. For more information, see the security guide.

3.2.3 Location of the VMS$OBJECTS Security Object Database (VAX Only)

The OpenVMS VAX operating system maintains the security characteristics of clusterwide objects in a database called VMS$OBJECTS.DAT, located in SYS$COMMON:[SYSEXE]. VMScluster system managers should ensure that the security object database is present on each node in the VMScluster by specifying a file name that resolves to the same file throughout the cluster, not to a file that is unique to each node.

The database is updated whenever characteristics are modified, and the information is distributed so that all nodes participating in the cluster share a common view of the objects. The security database is created and maintained by the audit server process.

If you relocate the database, be sure the logical name VMS$OBJECTS resolves to the same file for all nodes in a common-environment cluster. To reestablish the logical name after each system boot, define the logical in SYSECURITY.COM. ♦

3.2.4 Network Security

Network security should promote interoperability and uniform security approaches throughout networks. User authentication, VMScluster membership management, and using a security audit log file are three major areas of network security.

VMScluster system managers should also ensure maximum consistency in the use of DECnet software for intracluster communication. The cluster manager should set up a proxy database for nodes that, for example, need to access disks that are not cluster accessible or that use higher level features. See the security guide for your system to understand security for DECnet nodes, including how to set up a common proxy database (NETPROXY) that allows users on a remote node in a network to access data by way of a local account on your system.

Depending on the level of network security required, you might also want to consider how other security mechanisms, such as protocol encryption and decryption, can promote additional security protection across the cluster.

4

Preparing the Cluster Operating Environment

By setting up appropriate startup and other system files, you can prepare the VMScluster operating environment on the first installed computer before adding other computers to the cluster. Depending on your processing needs, you can prepare either a **common-environment** or a **multiple-environment** cluster.

In a common-environment cluster, the operating environment is identical on each computer in the VMScluster because the computers are run from common system files (which may include common system files, common resources, and the same applications). The computers are set up with identical user accounts, the same known images are installed, the same logical names are defined, and mass storage devices and queues are shared. In effect, users in a common-environment cluster can log in to any computer and work in the same operating environment.

In a multiple-environment cluster, the environment (resources, applications) varies from computer to computer, and users can work in environments that are specific to the computer to which they are logged in. A multiple-environment cluster is effective when you want to share data among computers but you want certain computers to serve specialized needs. For example, you might want to set up a three-computer cluster, in which the timesharing environments on two computers are the same, while the third computer is set up exclusively for batch processing of large inventory jobs. In this case, the timesharing computers are set up with a common environment, sharing users, queues, and access to mass storage devices, while the third computer runs in its own restricted environment.

This chapter concentrates on the steps necessary to prepare a common-environment cluster. Approaches for preparing a multiple-environment cluster are also described, but are presented as general guidelines.

Topics include the following:

- Directory structure on a common system disk
- Installing the operating system in the VMScluster environment
- Configuring and starting the DECnet for OpenVMS network
- Coordinating startup command procedures
- Coordinating system files for a common-environment cluster

Once you have prepared the cluster operating environment as described in this chapter, and you have determined your disk and queue configurations using the information in Chapter 5 and Chapter 6, you can build the cluster by following the instructions in Chapter 7.

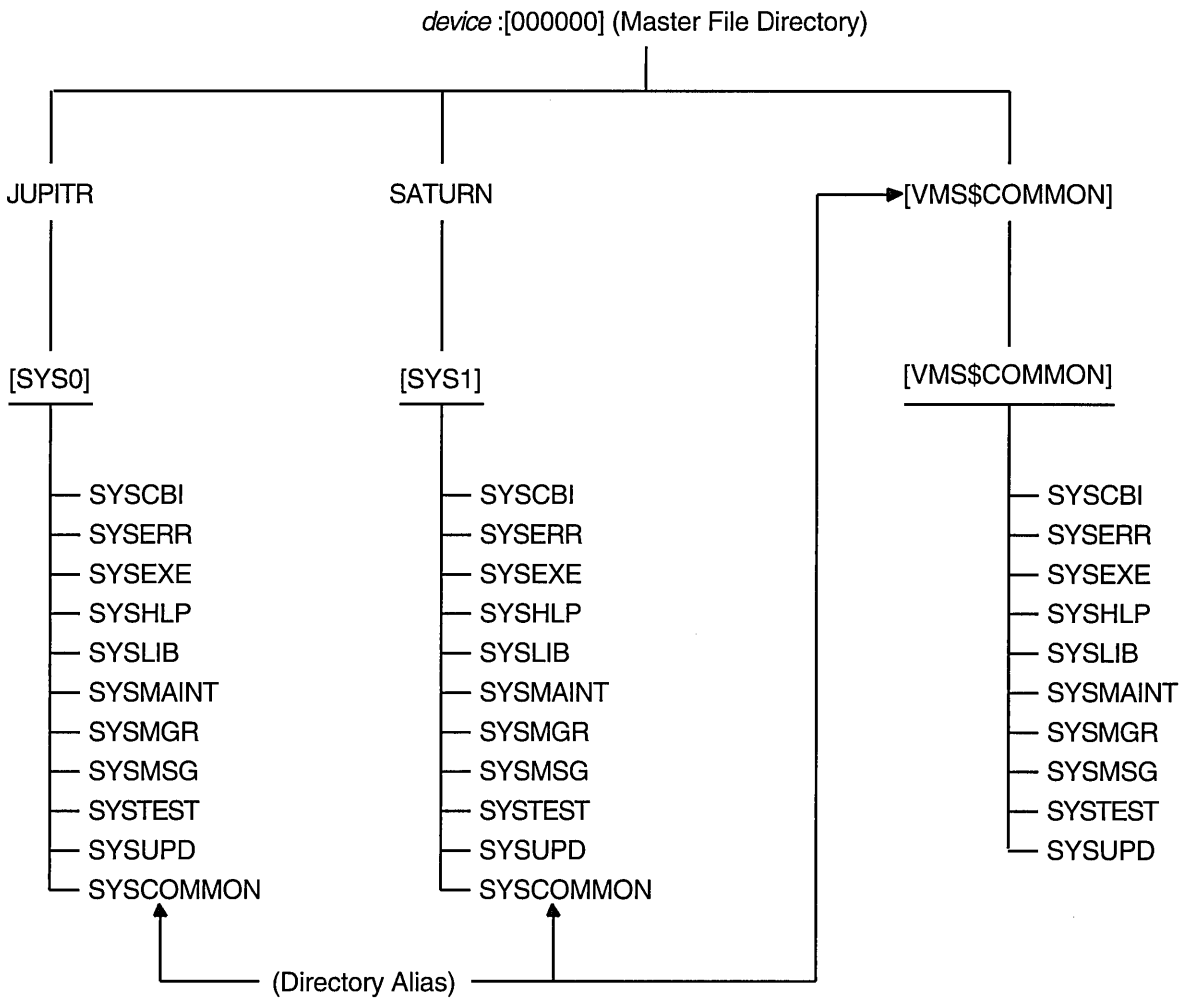
4.1 Directory Structure on Common AXP or VAX System Disks

The installation or upgrade procedure for your operating system generates a **common system disk**, on which most operating system and optional product files are stored in a common root directory. The system disk directory structure is the same on both AXP and VAX systems, Whether the system disk is for AXP or VAX, the entire directory structure—that is, the common root plus each computer's local root—is stored on the same disk. After the installation or upgrade completes, you use the CLUSTER_CONFIG.COM command procedure described in Chapter 7 to create a local root for each new computer and boot it into the cluster.

Each local root contains, in addition to the usual system directories, a [SYS*n*.SYSCOMMON] directory that is an alias for [VMS$COMMON], the cluster common root directory in which cluster common files actually reside. When you add a computer to the cluster, CLUSTER_CONFIG.COM defines the alias.

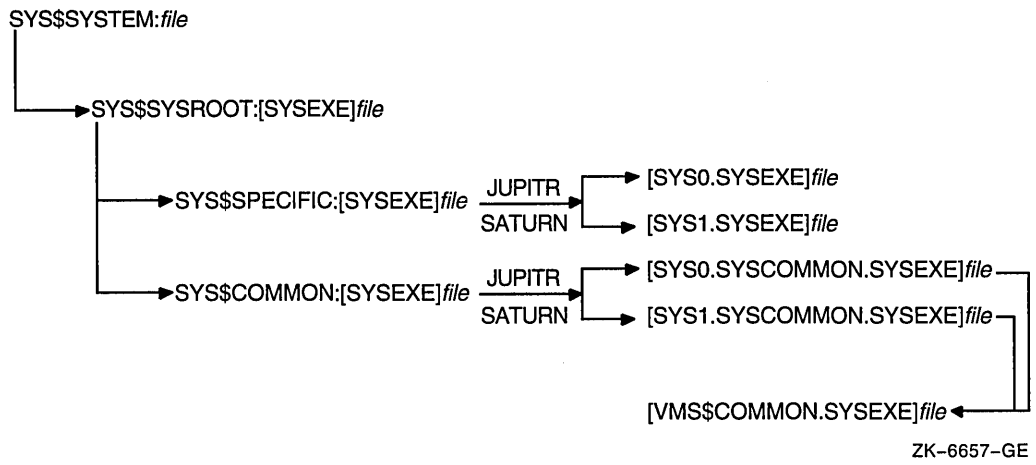
Figure 4–1 illustrates the directory structure set up for computers JUPITR and SATURN, which are run from a common system disk. The disk's master file directory (MFD) contains the local roots (SYS0 for JUPITR, SYS1 for SATURN) and the cluster common root directory, [VMS$COMMON].

Figure 4–1 Directory Structure on a Common System Disk



The logical name SYS$SYSROOT is defined as a search list that points to a local root first (SYS$SPECIFIC) and then to the common root (SYS$COMMON). Thus, the logical names for the system directories (SYS$SYSTEM, SYS$LIBRARY, SYS$MANAGER, and so forth) point to two directories: a local root (for example, SYS$SPECIFIC:[SYSEXE]) and a common root (for example, SYS$COMMON:[SYSEXE]). Figure 4–2 shows how directories on a common system disk are searched when the logical name SYS$SYSTEM is used in file specifications.

Figure 4–2 File Search Order on Common System Disk



It is important to keep this search order in mind when manipulating system files on a common system disk. Computer-specific files must always reside and be updated in the appropriate computer's system subdirectory. For example, MODPARAMS.DAT must reside in SYS$SPECIFIC:[SYSEXE], which is [SYS0.SYSEXE] on JUPITR, and in [SYS1.SYSEXE] on SATURN. Thus, to create a new MODPARAMS.DAT file for JUPITR when logged in on JUPITR, you would enter the following command:

```
$ EDIT SYS$SPECIFIC:[SYSEXE]MODPARAMS.DAT
```

Once the file is created, use the following command to modify it:

```
$ EDIT SYS$SYSTEM:MODPARAMS.DAT
```

Note that if a MODPARAMS.DAT file does not exist in JUPITR's SYS$SPECIFIC:[SYSEXE] directory when you enter this command, but there is a MODPARAMS.DAT file in the directory SYS$COMMON:[SYSEXE], the command edits the MODPARAMS.DAT file in the common directory. If there is no MODPARAMS.DAT file in either directory, the command creates the file in JUPITR's SYS$SPECIFIC:[SYSEXE] directory.

To modify JUPITR's MODPARAMS.DAT when logged in on any other computer that boots from the same common system disk, you enter the following command:

```
$ EDIT SYS$SYSDEVICE:[SYS0.SYSEXE]MODPARAMS.DAT
```

If you want to modify records in the cluster common system authorization file in a cluster with a single cluster common system disk, enter the following commands on any computer:

```
$ SET DEFAULT SYS$COMMON:[SYSEXE]
$ MCR AUTHORIZE
```

If you want to modify records in a computer-specific system authorization file when logged in to another computer that boots from the same cluster common system disk, you must set your default directory to the specific computer. For example, if you have set up a computer-specific system authorization file (SYSUAF.DAT) for computer JUPITR, you must set your default directory to

JUPITR's computer-specific [SYSEXE] directory before invoking AUTHORIZE. This scenario is demonstrated by the following commands:

```
$ SET DEFAULT SYS$SYSDEVICE:[SYS0.SYSEXE]
$ RUN AUTHORIZE
```

4.2 Installing the Operating System

You must perform the installation or upgrade once for each system disk in the VMScluster. However, because several computers normally run from the same cluster common system disk, you need not perform the installation or upgrade on each computer.

Note

An AXP system disk cannot be used to boot VAX computers, and an VAX system disk cannot be used to boot AXP computers.

Refer to the release notes for the required version numbers of hardware and firmware. When mixing versions of the operating system, check the release notes for information about compatibility.

You may want to set up a cluster that has a combination of one or more common system disks and one or more individual system disks. Again, you must do the installation or upgrade once for each system disk, or use the CLUSTER_CONFIG.COM procedure to create a duplicate system disk. For example, if your cluster consists of 10 computers, 4 of which share one common system disk, 4 of which share a second common system disk, and 2 of which each have their own system disk, you would do the installation or upgrade four times.

Note

Note that if your cluster includes multiple common system disks, you must later coordinate system files to define the cluster operating environment, as described in Section 4.5.4.

To perform the installation, follow instructions in the installation and operations guide for your computer. However, before you start the installation, be sure you have determined which *VMScluster system configuration type* you want to create (CI, DSSI, local area, or mixed interconnect), because the installation procedure requests configuration-specific information. (Configuration types are described in Section 2.1.)

Table 4–1 lists the information requested for configurations based on the CI; this information is also required for DSSI-based configurations. Table 4–2 lists the information requested for local area and mixed-interconnect configurations. Typical responses are explained in the tables. Note that initial questions are the same for all configuration types.

All references to the Ethernet are also applicable to FDDI.

If your system disk is on an HSC or ISE subsystem, you must obtain the HSC or ISE subsystem's **disk allocation class** value before starting the installation, because the installation procedure requests that information. (Allocation classes are discussed in detail in Section 5.2.) For example, to obtain the value, enter a

command sequence like the following at the HSC or ISE console. The information displayed includes the allocation class value.

```
Ctrl/C
HSC> SHOW SYS
15-MAY-1993 14:31:43.41  Boot:   13-MAY-1993 11:31:11.41   Up:   51:00
   .
   .
   .
DISK allocation class =   1        TAPE allocation class =   0
Start command file m Disabled

SETSHO - Program Exit
```

If later you want to change the allocation class value, follow the instructions in Section 7.6.

Note

While rebooting at the end of the installation procedure, the system displays messages warning that you must install the operating system software and VAXcluster and VMScluster software licenses. Be sure to install these licenses, as well as the DECnet for OpenVMS license, as soon as the system is available. Procedures for installing licenses are described in the release notes distributed with the software kit and in the *OpenVMS License Management Utility Manual.*

Table 4–1 Information Requested for CI Configurations

Prompt	Response
Will this node be a cluster member (Y/N)?	Enter Y.
What is the node's DECnet node name?	Enter DECnet node name—for example, JUPITR. The DECnet node name can be from 1 to 6 alphanumeric characters in length and cannot include dollar signs ($) or underscores (_).
What is the node's DECnet node address?	Enter DECnet node address—for example, 2.2.
Will the Ethernet be used for cluster communications (Y/N)?	Enter N. The LAN is not used for cluster communications in VMScluster systems that are based only on the CI.
Will JUPITR be a disk server (Y/N)?	Enter Y or N, depending on your configuration requirements. Refer to Section 2.5 and Chapter 5 for information on served cluster disks.
Enter a value for JUPITR's ALLOCLASS parameter:	If the system disk is connected to a dual-ported disk, enter a value from 1 to 255 that will be used on both sides. Otherwise, enter 0 (zero). (For detailed information on allocation classes see Section 5.2.)
Does this cluster contain a quorum disk [N]?	Enter Y or N, depending on your configuration. If you enter Y, the procedure prompts for the name of the quorum disk. Enter the device name of the quorum disk. (For detailed information on quorum disks, see Section 3.1.2.)

Table 4–2 Information Requested for Local Area and Mixed-Interconnect Configurations

Prompt	Response
Will this node be a cluster member (Y/N)?	Enter Y.
What is the node's DECnet node name?	Enter DECnet node name—for example, JUPITR. The DECnet node name may be from 1 to 6 alphanumeric characters in length and cannot include dollar signs ($) or underscores (_).
What is the node's DECnet node address?	Enter DECnet node address—for example, 2.2.
Will the Ethernet be used for cluster communications (Y/N)?	Enter Y. The LAN is required for cluster communications in local area and mixed-interconnect VMScluster systems.
Enter this cluster's group number:	Enter a number in the range from 1–4095 or 61440–65535.
Enter this cluster's password:	Enter the cluster password. The password must be from 1 to 31 alphanumeric characters in length and can include dollar signs ($) and underscores (_).
Reenter this cluster's password for verification:	Reenter the password.
Will JUPITR be a disk server (Y/N)?	Enter Y. In local area and mixed-interconnect configurations, the system disk is always served to the cluster. Refer to Section 2.5 and Chapter 5 for information on served cluster disks.
Will JUPITR serve HSC or RF disks (Y/N)?	Enter a response appropriate for your configuration.
Enter a value for JUPITR's ALLOCLASS parameter:	If the system will serve HSC disks, enter the allocation class value of the HSC. If the system disk is connected to a dual-ported disk, enter a value from 1 to 255 that will be used on both sides. If the system will serve RF disks, assign a nonzero value to the allocation class. Otherwise, enter 0 (zero). (For detailed information about allocation classes, see Section 5.2.)
Does this cluster contain a quorum disk [N]?	Enter Y or N, depending on your configuration. If you enter Y, the procedure prompts for the name of the quorum disk. Enter the device name of the quorum disk. (For detailed information about quorum disks, see Section 3.1.2.)

4.3 Configuring and Starting the DECnet for OpenVMS Network

After you have installed the operating system and the required licenses on the first VMScluster computer, you can configure, tailor, and start the DECnet for OpenVMS network. If you locate certain network files in the SYS$COMMON:[SYSEXE] directory as described in step 3 of the following procedure, other computers can share the data when they join the cluster. The process of configuring the network typically entails several operations:

- Executing the SYS$MANAGER:NETCONFIG.COM command procedure.
- Selecting a single LAN adapter when using multiple LAN adapters on a single extended LAN.
- Making remote node data available clusterwide. Remember that if satellite booting is being used anywhere in the VMScluster system, then the NETNODE_REMOTE.DAT file must not be shared between VAX and AXP processors. This restriction ensures that AXP systems do not try to downline load VAX systems and that VAX systems do not try to downline load AXP systems.

- Defining a VMScluster alias (optional). Establish an alias using Network Control Program (NCP) commands like those shown in step 4 for alias SOLAR. (For more information on the VMScluster alias, refer to the *DECnet for OpenVMS Networking Manual*.) Note that if you plan to define an alias, you must specify that at least one computer operate as a *router* node.

  VAX

  On VAX systems, you can designate a computer as a router node when you execute NETCONFIG.COM (see Example 4–1).♦

  AXP

  On AXP systems, you might need to enable level 1 routing manually because the NETCONFIG.COM procedure does not prompt you with the routing question. Depending on whether the configuration includes all AXP nodes or a combination of VAX and AXP nodes:

  – You must enable level 1 routing manually on one of the AXP nodes for VMScluster systems that consists of AXP nodes only.♦

  – You do not need to enable level 1 routing on an AXP node in dual-architecture VMScluster configurations if one of the VMS Version 5.5–2 nodes is already a routing node.

  – You do not need to enable the DECnet extended function license DVNETRTG on an AXP node if one of the VAX nodes is already a routing node.

  Note further that you must later enable alias operations for other computers, as described in Section 4.3.2.

- Starting the network.

To perform these operations, proceed as follows:

1. Log in as system manager and execute the NETCONFIG.COM command procedure. Enter information about your node when prompted, and answer NO when the procedure asks whether you want network configuration commands to be executed. This allows you to configure a VMScluster alias and to select a LAN adapter when using multiple adapters connected to a single extended LAN.

   Example 4–1 shows a typical NETCONFIG.COM session on an VAX node.

   AXP

   On AXP systems, this session is the same on AXP nodes, except that the question "Do you want to operate as a router?" is not asked. Also, you must manually enable level 1 routers on AXP systems as shown in step 4 for alias SOLAR.♦

Example 4–1 Sample Interactive Network Configuration Session

```
$ @NETCONFIG.COM

          DECnet for OpenVMS network configuration procedure

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

What do you want your DECnet node name to be?  [JUPITR]: Return
What do you want your DECnet address to be?    [2.2]:Return
Do you want to operate as a router?  [NO (nonrouting)]:YESReturn
Do you want a default DECnet account?              [NO]: Return
```

(continued on next page)

Example 4–1 (Cont.) Sample Interactive Network Configuration Session

```
            Here are the commands necessary to set up your system.
    .
    .
    .
Do you want these commands to be executed?        [YES]: Return
    .
    .
    .
The changes have been made.
If you have not already registered the DECnet for OpenVMS key, then do so now.
After the key has been registered, you should invoke the procedure
SYS$MANAGER:STARTNET.COM to start up DECnet for OpenVMS with these changes.

(If the key is already registered) Do you want DECnet started? [YES] NO
$
```

2. NETCONFIG.COM configures all the LAN adapters for DECnet use. However, on a single extended LAN, this violates LAN configuration rules. A single LAN adapter must be chosen to connect to each extended LAN. See Section 2.9.1 for more information about distributing connections to LAN segments.

   For example, to disable DECnet use of the LAN device XQB0, invoke NCP and issue the following commands:

   ```
   $ RUN SYS$SYSTEM:NCP
   NCP> PURGE CIRCUIT QNA-1 ALL
   NCP> PURGE LINE QNA-1 ALL
   NCP> EXIT
   ```

   For further details, see the *DECnet for OpenVMS Networking Manual* and *DECnet for OpenVMS Network Management Utilities.*

3. NETCONFIG.COM creates in the SYS$SPECIFIC:[SYSEXE] directory the permanent remote node database file NETNODE_REMOTE.DAT, in which remote node data is maintained. To make this data available throughout the VMScluster, you must rename the file to the SYS$COMMON:[SYSEXE] directory:

   ```
   $ RENAME SYS$SPECIFIC:[SYSEXE]NETNODE_REMOTE.DAT -
   _$ SYS$COMMON:[SYSEXE]NETNODE_REMOTE.DAT
   ```

   Note

   If satellite booting is being used anywhere in the VMScluster, then the NETNODE_REMOTE.DAT file must not be shared between VAX and AXP processors. This restriction ensures that VAX systems do not try to downline load AXP systems.

   For information about sharing other network data, see the *DECnet for OpenVMS Networking Manual.*

AXP

4. On AXP systems, if you need to enable level 1 routing on one or more AXP nodes, invoke the NCP utility to do so. For example:

```
$ RUN SYS$SYSTEM:NCP
NCP>  DEFINE EXECUTOR TYPE ROUTING IV♦
```

5. If you want to define a VMScluster alias, invoke the NCP utility to do so. For example:

```
$ RUN SYS$SYSTEM:NCP
NCP> DEFINE NODE 2.1 NAME SOLAR
NCP> DEFINE EXECUTOR ALIAS NODE SOLAR
NCP> EXIT
$
```

   The information you specify using these commands is entered in the DECnet for OpenVMS permanent executor database and takes effect when you start the network.

6. Start the network:

```
$ @SYS$MANAGER:STARTNET.COM
```

7. To ensure that the network is started each time a VMScluster computer boots, add the following line to the appropriate startup command file or files:

```
$ @SYS$MANAGER:STARTNET.COM
```

For information about preparing startup command files, see Section 4.4. For more detailed information about DECnet configuration issues and procedures, refer to the *DECnet for OpenVMS Networking Manual.*

4.3.1 Copying Remote Node Databases

Some sites with large networks maintain remote node data in a central database file. If this is the case at your site, and if you want to make the data available clusterwide, you can, *after starting the network*, copy remote node database entries from that central file. For example, if the file resides on node SATURN, you could enter the following NCP commands to copy entries from the permanent database on SATURN to the permanent database on your system disk, and then to update your volatile database:

```
NCP> SET NODE 2.2 NAME SATURN
NCP> COPY KNOWN NODES FROM SATURN USING PERMANENT TO PERMANENT
NCP> SET KNOWN NODES ALL
```

Note that only node names and addresses are copied. See the *DECnet for OpenVMS Networking Manual* for more information about copying node databases.

4.3.2 Enabling VMScluster Alias Operations

If you have defined a VMScluster alias as described in Section 4.3, you can enable alias operations for other computers *after the computers are up and running in the cluster.* To enable such operations (that is, to allow a computer to accept incoming connect requests directed toward the alias), follow these steps:

1. Log in as system manager and invoke the SYSMAN utility:

```
$ RUN SYS$SYSTEM:SYSMAN
```

2. At the SYSMAN> prompt, enter the following commands:

```
SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username LAZARUS       will be used on nonlocal nodes
SYSMAN> SET PROFILE/PRIVILEGES=(OPER,SYSPRV)
SYSMAN> DO MCR NCP SET EXECUTOR STATE OFF
%SYSMAN-I-OUTPUT, command execution on node X...
   .
   .
   .
SYSMAN> DO MCR NCP DEFINE EXECUTOR ALIAS INCOMING ENABLED
%SYSMAN-I-OUTPUT, command execution on node X...
   .
   .
   .
SYSMAN> DO @SYS$MANAGER:STARTNET.COM
%SYSMAN-I-OUTPUT, command execution on node X...
   .
   .
   .
```

4.4 Coordinating Startup Command Procedures

You must coordinate your site-specific SYSTARTUP and SYLOGIN command procedures according to the type of cluster operating environment you want to prepare. For a common-environment cluster, these procedures should perform the same system startup and login functions for each computer. For a multiple-environment cluster, you might want some startup commands to remain specific to certain computers, as described in Section 4.4.2.

In a common-environment cluster, you can prepare SYSTARTUP_VMS.COM procedures using one of the following methods:

- In each computer's SYS$SPECIFIC:[SYSMGR] directory, set up a SYSTARTUP_VMS.COM procedure that performs computer-specific startup functions and then invokes a common SYSTARTUP procedure, typically named SYSTARTUP_COMMON.COM. This procedure is usually located in the SYS$COMMON:[SYSMGR] directory on a common system disk but can reside on any disk, provided that the disk is cluster accessible and is mounted when the procedure is invoked.
- After setting up computer-specific SYSTARTUP_VMS.COM procedures, create a copy of SYSTARTUP_COMMON.COM for each computer. However, if you use multiple SYSTARTUP_COMMON.COM files, you must update all copies whenever you make changes.

To set up a common SYLOGIN procedure, define the logical name SYS$SYLOGIN on each computer to be the full file specification of the procedure. If the common SYLOGIN file is on a cluster-accessible disk, include the command that defines SYS$SYLOGIN in your common SYSTARTUP command file. If the computers use separate duplicate copies of SYLOGIN.COM, include the definition in each computer's specific startup procedure.

For example, the following command defines SYS$SYLOGIN to be the common file [SYSMGR]SYLOGIN.COM on the cluster-accessible disk WORK5:

```
$ DEFINE/SYSTEM/EXEC SYS$SYLOGIN WORK5:[SYSMGR]SYLOGIN
```

Certain startup functions, even in a common-environment cluster, are computer specific. Therefore, you must include commands in the computer-specific startup procedure on each computer to do the following:

- Set up dual-ported and local disks.
- Load device drivers.
- Set up local terminals and terminal server access.
- Invoke the common SYSTARTUP command procedure.

Section 4.4.1 and Section 4.4.2 present guidelines for using common and computer-specific command procedures to build a cluster environment.

4.4.1 Building Startup Procedures for a Common-Environment Cluster

The first step in preparing a common-environment cluster is to build cluster common SYSTARTUP and SYLOGIN command procedures. In a common-environment cluster, each computer executes the common procedures at startup time to define the same operating environment. Because each computer is set up with the common procedure, users can work in the same operating environment on any VMScluster computer.

4.4.1.1 Procedures for Existing Computers

To build procedures for a cluster in which existing computers are to be combined in a VMScluster system, you should compare both the computer-specific SYSTARTUP and SYLOGIN command procedures on each computer and make any adjustments required. For example, you can compare the procedures from each computer and include commands that define the same logical names in your common SYSTARTUP command file.

An easy method of comparing the existing procedures and creating common versions is to log in to each computer (in the single-computer environment) and use the DCL command DIFFERENCES to compare the contents of the files. Another option is to print the existing SYSTARTUP and SYLOGIN command procedure files. You can then use the file listings to compare the procedures. After you have chosen which commands to make common, you can build the common procedures on one of the VMScluster computers.

Note

If you boot nodes into an existing VMScluster using minimum startup (the system parameter STARTUP_P1 is set to MIN), a number of processes (for example, CACHE_SERVER, CLUSTER_SERVER, and CONFIGURE) are not started. Digital recommends that you start these processes manually if you intend to run the VMScluster system for an extended period of time. Extended processing without these processes enabled is not recommended. Refer to the *OpenVMS System Manager's Manual* for more information about starting these processes manually.

4.4.1.2 Procedures for Newly Installed Computers

The strategy for clusters being formed from newly installed operating systems is basically the same as that used for clusters that are to include previously installed systems: include common elements in a common command procedure file (for example, SYSTARTUP_COMMON.COM). With newly installed systems, however, the SYSTARTUP and SYLOGIN command procedure files are empty. Therefore, you must start building the common procedures again.

For example, you could build a common startup command procedure named SYSTARTUP_COMMON.COM and include the commands that you want to be common to all computers. You must decide which of the following elements you want to include in the common procedure:

- Commands that install images.
- Commands that define logical names; for example, the logical name that refers to the location of SYLOGIN.COM.
- Commands that set up queues. (See Chapter 6 for information about setting up cluster queues.)
- Commands that set up and mount physically accessible mass storage devices. (See Chapter 5 for information about setting up cluster disks.)
- Commands that perform any other common startup functions. (See the *OpenVMS System Manager's Manual* for more information about startup command procedures.)

To build a common SYLOGIN.COM command file, include in the file commands that define clusterwide logical names and symbols.

You can include commands that mount cluster-accessible storage devices and that set up queues in the common SYSTARTUP procedure or in separate command files (such as SYS$EXAMPLES:MSCPMOUNT.COM) that are invoked by the common procedure. However, because such commands are computer specific, they must be executed by the local computer. Therefore, you must use conditional logic to control their execution. Sample command files for mounting storage devices and setting up queues are described in Chapter 5 and Chapter 6, respectively.

4.4.2 Building Startup Procedures for a Multiple-Environment Cluster

To build SYSTARTUP and SYLOGIN command files for a multiple-environment cluster, include in the files elements that you want to remain unique to a computer, such as commands to define computer-specific logical names and symbols. These files must be placed in the SYS$SPECIFIC root on each computer.

For example, consider a three-member cluster consisting of computers JUPITR, SATURN, and PLUTO. The timesharing environments on JUPITR and SATURN are the same. However, PLUTO runs applications for a specific user group. In this cluster, you would create common SYSTARTUP and SYLOGIN command procedures for JUPITR and SATURN that define identical environments on these computers. But the command procedures for PLUTO would be different; they would include commands to define PLUTO's special application environment.

4.5 Coordinating System Files for a Common-Environment Cluster

To prepare a common VMScluster user environment, you must coordinate the following system files:

- AUDIT_SERVER.DAT
- NETNODE_REMOTE.DAT[1]
- NETOBJECT.DAT
- NETPROXY.DAT
- QMAN$MASTER.DAT
- RIGHTSLIST.DAT
- SYSALF.DAT (optional file for the Autologin facility)
- SYSUAF.DAT
- SYS$QUEUE_MANAGER.QMAN$JOURNAL
- SYS$QUEUE_MANAGER.QMAN$QUEUES
- VMSMAIL_PROFILE.DATA
- VMS$OBJECTS.DAT (VAX only)

These files, which are part of the operating system, control such functions as user logins, proxy login access, mail, and access to files and job queues. By coordinating these files, you can define either a common-environment or a multiple-environment cluster.

In a common-environment cluster, you use a common version of each system file and place the files in the SYS$COMMON:[SYSEXE] directory on a common system disk or on a disk that is mounted by all cluster nodes (see Section 4.5.4). In a multiple-environment cluster, you would use computer-specific versions of the files and place the files in each computer's SYS$SPECIFIC:[SYSEXE] directory.

Section 4.5.1 describes procedures for coordinating user accounts in common SYSUAF.DAT and NETPROXY.DAT files. Section 4.5.2 and Section 4.5.3 describe procedures for preparing the cluster RIGHTSLIST and VMSMAIL_PROFILE database files, respectively. For detailed information on queue management, refer to Chapter 6. The NETNODE_REMOTE.DAT file is described in Section 4.3.

Note

If you want to set up a common-environment cluster with more than one common system disk, you must coordinate files on each disk and ensure that the disks are mounted with each cluster reboot. Refer to Section 4.5.4 for instructions.

[1] This file cannot be shared between VAX and AXP computers if satellite booting is used anywhere in the VMScluster. See Section 4.3 for more information.

4.5.1 Coordinating User Accounts

In a common-environment cluster, you must coordinate the user accounts from each computer and build common versions of the following files:

- SYSUAF.DAT
- NETPROXY.DAT

Refer also to Section 3.2 for information about additional files that should be maintained on cluster-accessible disks in order to maintain a single security environment.

Note

The default values for a number of SYSUAF process limits and quotas are higher on AXP computers than they are on VAX computers. In general, the values in a common SYSUAF.DAT file should accommodate the largest requirements for the cluster. You can then adjust system parameters to lower values in MODPARAMS.DAT files on individual nodes that require it. See also *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for information about parameter settings on both AXP and VAX computers.

If you are setting up a common-environment cluster that consists of newly installed systems, you can follow the instructions in the *OpenVMS System Manager's Manual* to build these files. Because the SYSUAF.DAT file on new operating systems is empty except for the Digital-supplied accounts, very little coordination is necessary.

However, if the cluster will include one or more computers that have been running with computer-specific SYSUAF.DAT and NETPROXY.DAT files, you must create common versions of the files. Procedures for creating a common SYSUAF.DAT file from computer-specific files are described in Appendix B.

Procedures for creating a common NETPROXY.DAT file are basically the same as those for creating a common SYSUAF.DAT file, except that less coordination is needed when you merge the individual NETPROXY.DAT files. For example, user identification codes (UICs) are not used in the NETPROXY records and therefore need not be coordinated. You should decide which existing proxy login records you want to keep and include these records in the common NETPROXY.DAT file.

Once you have prepared SYSUAF.DAT and NETPROXY.DAT files, you can set up each of them either as a common file on a cluster-accessible disk or as separate duplicate files. Note, however, that if you choose to use duplicate files, you must update all copies whenever you make changes.

If your cluster is running from one common system disk, make sure that SYSUAF.DAT and NETPROXY.DAT are located in the SYS$COMMON:[SYSEXE] directory.

If your cluster is running from any other system disk configuration, you must decide where to locate SYSUAF.DAT and NETPROXY.DAT. Once you have placed these two files in a directory, you must define clusterwide logical names to point to them. Note that the volume containing the files must be local to the node.

Assume that disk WORK5 is shared by all computers in the cluster and that it contains cluster common SYSUAF.DAT and NETPROXY.DAT files. The following commands define system logical names that point to the location of the common files:

```
$ DEFINE/SYSTEM/EXEC SYSUAF WORK5:[SYSEXE]SYSUAF
$ DEFINE/SYSTEM/EXEC NETPROXY WORK5:[SYSEXE]NETPROXY
```

You must add the definitions to the SYLOGICALS.COM command procedure. After you have copied the files to the appropriate directory on the cluster-accessible disk, you should delete these files from the system disk. To ensure that the disk where the common files reside are correctly mounted with each reboot, follow these steps:

1. Copy the SYS$EXAMPLES:CLU_MOUNT_DISK.COM file to the [VMS$COMMON.SYSMGR] directory.
2. Edit SYLOGICALS.COM and include commands to mount, with the appropriate volume label, the disk containing the shared files. For example, if the disk is 1DJA16, include a command like the following:

   ```
   $ @SYS$EXAMPLES:CLU_MOUNT_DISK.COM -
   _$ $1$DJA16: volume-label
   ```

4.5.2 Preparing the Rights Database

The rights database file, RIGHTSLIST.DAT, associates users of the system or cluster with special names called **identifiers**. This file is the basis of a protection scheme that uses access control lists (ACLs). For a detailed description of this scheme, see the security guide. For information about how the rights database is created, refer to the *OpenVMS System Management Utilities Reference Manual*.

The cluster manager or security manager maintains the rights database, adding and removing identifiers as needed. By allowing groups of users to hold identifiers, the manager can create a different kind of group designation than the one based on UICs. This alternative grouping allows the holders of the identifier to make more efficient use of resources. It also permits each user to be a member of multiple overlapping groups.

If your cluster is running from one common system disk, the installation or upgrade procedure places the RIGHTSLIST.DAT file in the directory SYS$COMMON:[SYSEXE]. No further action is required on your part.

If your cluster is running from any other system disk configuration, copy SYS$SYSTEM:RIGHTSLIST.DAT to the directory in which you placed the SYSUAF.DAT and NETPROXY.DAT files. Then define a clusterwide logical name for the RIGHTSLIST.DAT file. Note that the volume containing the file must be local to the node at which you enter the following command. For example:

```
$ DEFINE/SYSTEM/EXEC RIGHTSLIST WORK5:[SYSEXE]RIGHTSLIST
```

You must also add the definition to the SYLOGICALS file.

To ensure that the disk where the common files reside are mounted correctly with each reboot, follow these steps:

1. Copy the SYS$EXAMPLES:CLU_MOUNT_DISK.COM file to the [VMS$COMMON.SYSMGR] directory.
2. Edit SYLOGICALS.COM and include commands to mount, with the appropriate volume label, the disk containing the shared files. For example, if the disk is 1DJA16, include a command like the following:

```
$ @SYS$EXAMPLES:CLU_MOUNT_DISK.COM $1$DJA16: volume-label
```

4.5.3 Preparing the MAIL Database

In a common-environment cluster, you may want to prepare a common mail database to allow users to use the Mail utility (MAIL) to send and read their mail messages from any computer in the cluster.

Each time MAIL executes in a single-system environment, it accesses a database file named SYS$SYSTEM:VMSMAIL_PROFILE.DATA. To set up VMSMAIL_PROFILE.DATA as a common file, define the logical name VMSMAIL_PROFILE to be the complete file specification of the common file by specifying the DEFINE command in the following format:

```
$ DEFINE/SYSTEM/EXEC VMSMAIL_PROFILE file-spec
```

You must make sure that you define the logical name before you invoke MAIL for the first time. When invoked for the first time, MAIL creates the database file, VMSMAIL_PROFILE.DATA, in SYS$SYSTEM by default. By defining VMSMAIL_PROFILE to be the location of a common file on a cluster-accessible disk, you cause MAIL to create and use that file.

If your cluster is running from one common system disk, define VMSMAIL_PROFILE to be SYS$COMMON:[SYSEXE]VMSMAIL_PROFILE and invoke the Mail utility by entering the following two commands:

```
$ DEFINE/SYSTEM/EXEC VMSMAIL_PROFILE SYS$COMMON:[SYSEXE]VMSMAIL_PROFILE
$ MAIL
```

VMSMAIL_PROFILE.DATA is created in the common system directory. You no longer need to use the logical name or make changes to your common SYSTARTUP command file.

If your cluster is running from any other system disk configuration, you must decide where to locate the common VMSMAIL_PROFILE.DATA file. (Typically, you would place this file in the same directory in which SYSUAF.DAT and NETPROXY.DAT reside—for example, WORK5:[SYSEXE].) You then define a logical name for the file and invoke the Mail utility:

```
$ DEFINE/SYSTEM/EXEC VMSMAIL_PROFILE WORK5:[SYSEXE]VMSMAIL_PROFILE
$ MAIL
```

The DEFINE command defines VMSMAIL_PROFILE.DATA to be a file located in [SYSEXE] on the cluster-accessible disk volume WORK5. The first time MAIL is invoked, VMSMAIL_PROFILE.DATA is created in WORK5:[SYSEXE]. Subsequently, MAIL uses this file as the database. You must also add the definitions to the SYLOGICALS command file.

To ensure that the disk where the common files reside are correctly mounted with each reboot, follow these steps:

1. Copy the SYS$EXAMPLES:CLU_MOUNT_DISK.COM file to the [VMS$COMMON.SYSMGR] directory.
2. Edit SYLOGICALS.COM and include commands to mount, with the appropriate volume label, the disk containing the shared files. For example, if the disk is 1DJA16, you would include a command like the following:

```
$ @SYS$SYSDEVICE:[VMS$COMMON.SYSMGR]CLU_MOUNT_DISK.COM -
_$ 1$DJA16: volume-label
```

4.5.4 Coordinating Shared System Files in Clusters

To prepare a common user environment for a VMScluster system that includes more than one common VAX system disk or more than one common AXP system disk, you must coordinate on those disks the system files listed in Section 4.5. In local area and mixed-interconnect clusters, you must also coordinate the SYS$MANAGER:NETNODE_UPDATE.COM file, which is described in Section 7.5.1.2.

Proceed as follows:

1. Edit the file SYS$COMMON:[SYSMGR]SYLOGICALS.COM *on each system disk* and define logical names that specify the location of the cluster common files. For example, if the files will be located on 1DJA16, you could define logical names like the following:

```
$ DEFINE/SYSTEM/EXEC SYSUAF -
      $1$DJA16:[VMS$COMMON.SYSEXE]SYSUAF.DAT
$ DEFINE/SYSTEM/EXEC NETPROXY -
      $1$DJA16:[VMS$COMMON.SYSEXE]NETPROXY.DAT
$ DEFINE/SYSTEM/EXEC RIGHTSLIST -
      $1$DJA16:[VMS$COMMON.SYSEXE]RIGHTSLIST.DAT
$ DEFINE/SYSTEM/EXEC VMSMAIL_PROFILE -
      $1$DJA16:[VMS$COMMON.SYSEXE]VMSMAIL_PROFILE.DATA
$ DEFINE/SYSTEM/EXEC NETNODE_REMOTE -
      $1$DJA16:[VMS$COMMON.SYSEXE]NETNODE_REMOTE.DAT
$ DEFINE/SYSTEM/EXEC NETNODE_UPDATE -
      $1$DJA16:[VMS$COMMON.SYSMGR]NETNODE_UPDATE.COM
$ DEFINE/SYSTEM/EXEC QMAN$MASTER -
      $1$DJA16:[VMS$COMMON.SYSEXE]QMAN$MASTER.DAT
```

Note

If satellite booting is being used anywhere in the VMScluster, then the NETNODE_REMOTE.DAT file must not be shared between VAX and AXP processors. This restriction ensures that VAX systems do not try to downline load AXP systems.

2. To ensure that the system disks are mounted correctly with each reboot, follow these steps:
   a. Copy the SYS$EXAMPLES:CLU_MOUNT_DISK.COM file to the [VMS$COMMON.SYSMGR] directory.
   b. Edit SYLOGICALS.COM and include commands to mount, with the appropriate volume label, the system disk containing the shared files. For example, if the system disk is 1DJA16, include a command like the following:

```
$ @SYS$SYSDEVICE:[VMS$COMMON.SYSMGR]CLU_MOUNT_DISK.COM $1$DJA16: volume-label
```

3. When you are ready to start the queuing system, be sure you have moved the queue and journal files to a cluster-available disk. Any cluster common disk is a good choice if the disk has sufficient space. Enter the following command:

```
$ START/QUEUE/MANAGER $1$DJA16:[VMS$COMMON.SYSEXE]
```

When you execute the CLUSTER_CONFIG.COM command procedure to add computers to a cluster with more than one common system disk, you must use a different device name for each system disk on which computers are added. For this reason, CLUSTER_CONFIG.COM supplies as a default device name the logical volume name (for example, DISK$MARS_SYS1) of SYS$SYSDEVICE: on the local system.

Using different device names ensures that each computer added has a unique root directory specification, even if the system disks contain roots with the same name—for example, DISK$MARS_SYS1:[SYS10] and DISK$MARS_SYS2:[SYS10].

4.6 System Time on the Cluster

When a computer joins the cluster, the cluster attempts to set the joining computer's system time to the current time on the cluster. Although it is likely that the system time will be similar on each cluster computer, there is no assurance that the time will be set. Also, no attempt is made to ensure that the system times remain similar throughout the cluster. (For example, there is no protection against different computers having different clock rates.)

A VMScluster system spanning multiple time zones must use a single, clusterwide common time on all nodes. Use of a common time ensures timestamp consistency (for example, between applications, file system instances) running across the VMScluster members.

Use the SYSMAN command DO SET TIME to set the time across the cluster. Refer to the *OpenVMS DCL Dictionary* for information about the DO SET TIME command.

5

Setting Up and Managing Cluster Disks and Tapes

A VMScluster system can include two types of disk and tape devices:

- Restricted-access devices, which are accessible only by the local computer or computers to which they are directly connected
- Cluster-accessible devices, which are accessible by any computer in the cluster

As system manager, you are responsible for planning, organizing, and setting up the proper cluster device configuration for your site. You must decide which disk and tape devices should have access restricted to the local computer and which should be accessible to the cluster. For example, you may want to restrict access to a particular device to the users on the computer that is directly connected to the device. Alternatively, you may decide to set up a disk as a cluster-accessible device so that any user on any computer can allocate and use it.

You can use the information in this chapter to plan and set up your disk and tape configuration. Topics include the following:

- Cluster-accessible disk and tape devices
- Cluster device-naming conventions
- Shared disks
- Configuring cluster disk and tape devices
- Rebuilding cluster disks
- Using volume shadowing to duplicate data on multiple disks

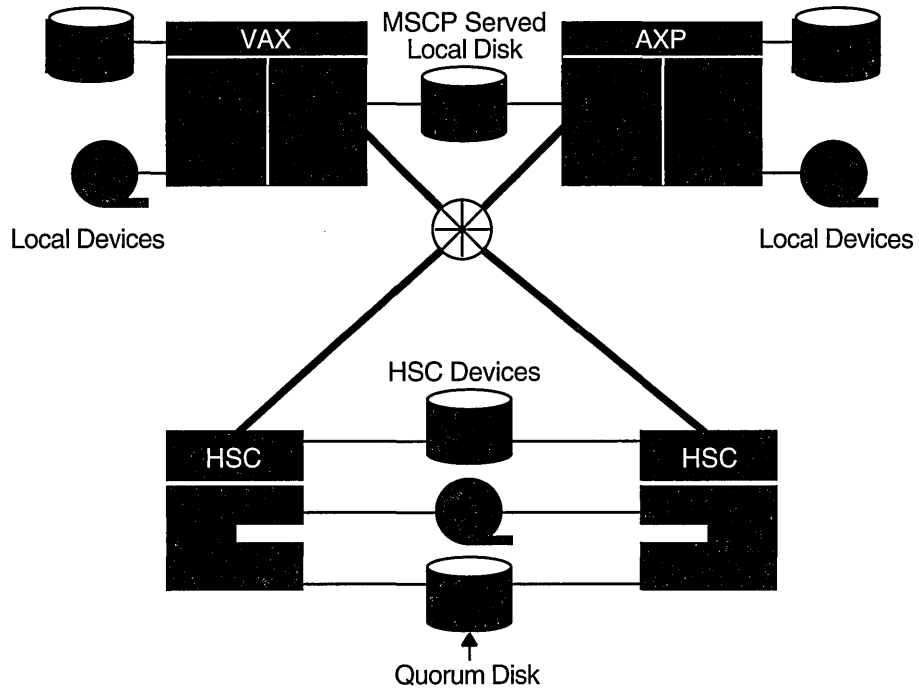
5.1 Cluster-Accessible Disk and Tape Devices

A cluster-accessible device is a disk or tape that multiple computers in the cluster can recognize and access. The following types of devices are cluster accessible:

- HSC disks and tapes
- Served disks
- Served tapes
- Dual-pathed disks and tapes
- DSSI disks and tapes

Figure 5–1 illustrates how disks and tapes might be configured in a VMScluster based on the CI. The HSC disks and tapes and the dual-ported MSCP served local disk are considered cluster accessible.

Figure 5–1 CI Configuration with Shared Disks and Tapes



ZK–1637–GE

5.1.1 HSC Disks and Tapes

An HSC disk or tape is a Digital Storage Architecture (DSA) device that is connected to an HSC subsystem. If an HSC subsystem is connected in a cluster, its disks and tapes are accessible by all VMScluster nodes connected to the same star coupler. These devices can be served to allow access by satellite nodes or by other VAX or AXP computers not connected to the same star coupler. You can set up HSC storage devices (including a quorum disk) to be dual pathed between two HSC subsystems, as shown in Figure 5–1. Dual-pathed disks and tapes are described in Section 5.1.3.

5.1.2 Served Disks and Tapes

The MSCP server and TMSCP server are used to communicate between a computer and a DSA controller. The servers enable a computer to make locally connected devices available to all other cluster members.

Locally connected disks and tapes are not automatically cluster accessible. Access to these devices is restricted to the local computer unless you explicitly set them up as cluster accessible using the MSCP server for disks or the TMSCP server for tapes.

5.1.2.1 MSCP and TMSCP Server Functions

To make a disk accessible to all VMScluster computers, the MSCP server must be loaded on the local computer, and it must be instructed to make the device available across the cluster. MSCP server functions are enabled with the MSCP_LOAD and MSCP_SERVE_ALL system parameters (see Table 5–1).

Similarly, to make a tape accessible to all VMScluster computers, the TMSCP server must be loaded on the local computer, and it must be instructed to make the device available across the cluster. TMSCP server functions are enabled by specifying the appropriate value for the TMSCP_LOAD parameter (see Table 5–1).

By specifying appropriate values for these parameters in a computer's MODPARAMS.DAT file and then running AUTOGEN to reboot the computer, you enable the computer to serve all suitable devices to the cluster early in the boot sequence. You can also use the CLUSTER_CONFIG.COM CHANGE function to perform these operations for disks and tapes. The served devices become accessible with minimal interruption whenever the serving computer reboots. Further, the servers automatically serve any suitable device that is added to the system later. For example, if new drives are attached to an HSC subsystem, the devices become available within seconds after the cables are connected.

Table 5–1 summarizes the system parameter values you can specify to configure the MSCP and TMSCP servers. Initial values are determined by your responses when you execute the installation or upgrade procedure, or when you execute the CLUSTER_CONFIG.COM command procedure described in Chapter 7 to set up your configuration. Note that if you change the values later, you must reboot the computer on which you changed the values before the new values can take effect (see Section 7.5.3).

Table 5–1 Specifying Values for MSCP_LOAD, MSCP_SERVE_ALL, and TMSCP_LOAD Parameters

Parameter	Value	Meaning
MSCP_LOAD	0	Do not load the MSCP server (default value).
	1	Load the MSCP server with attributes specified by MSCP_SERVE_ALL parameter, using the default CPU load capacity.
	>1	Load the MSCP server with attributes specified by the MSCP_SERVE_ALL parameter. Use this value as the CPU load capacity.
MSCP_SERVE_ALL	0	Do not serve any disks (default value).
	1	Serve all available disks.
	2	Serve only locally connected (not HSC) disks.
TMSCP_LOAD	0	Do not load the TMSCP server and do not serve any tapes (default value).
	1	Load the TMSCP server and serve all available tapes, including all local tapes and all multihost tapes with a matching TAPE_ALLOCLASS value.

For a sample use of these parameters, see the discussion of Figure 5–4 in Section 5.2.2.

5.1.2.2 MSCP Load Sharing

MSCP servers monitor their I/O traffic and periodically calculate a load-available rating to indicate available capacity for I/O requests. (TMSCP servers do not perform this monitoring function.)

Load availability is calculated by counting the read and write requests sent to the server and periodically converting this value to requests per second, then subtracting this calculated value from the server's load capacity (also specified in requests per second).

This information is communicated to the MSCP class driver (DUDRIVER and DSDRIVER). When a disk is mounted or a failover occurs, the class driver selects the server with the highest load-available rating to access the disk.

VMScluster systems use MSCP dynamic load balancing to balance the I/O load efficiently among systems within a VMScluster. Dynamic load balancing automatically checks server activity every 5 seconds. If activity to any server is excessive, the work load automatically shifts to other servers in the cluster. Excessive activity is measured according to either:

- The capacity value set by the MSCP_LOAD system parameter
- The CPU type (determined automatically by the operating system)

In addition, the load-balancing algorithm results in better I/O performance, faster I/O response, and a balanced work load among the members of a VMScluster by automatically determining which VMScluster member can provide maximum performance and efficiency when a disk is mounted or required to fail over.

Load balancing is enabled and controlled by the MSCP_LOAD and MSCP_SERVE_ALL system parameters. In most cases, the values established by CLUSTER_CONFIG.COM are appropriate.

The MSCP_SERVE_ALL parameter determines whether the server participates in load sharing. If the parameter is set to 2 (serve only local disks), the server does not monitor its I/O traffic and does not participate in load balancing. Other valid settings for MSCP_SERVE_ALL (0 or 1) result in the server monitoring I/O traffic and communicating load-available information to the class drivers.

The MSCP_LOAD parameter is used to communicate load capacity to the server, in addition to its existing function of controlling the loading of the MSCP server. If the parameter is set to 1, the MSCP server is loaded and its load capacity is set to a default value based on CPU type. If MSCP_LOAD is set to a value greater than 1, the server is loaded and its load capacity set to that value. Setting MSCP_LOAD to 0 disables loading of the MSCP server.

VAX

Table 5–2 shows the load-balancing rating for various VAX CPUs. The table assumes the configuration uses one Ethernet adapter (of the fastest supported type) and that disk I/O is not a bottleneck.♦

Table 5–2 MSCP Load-Balancing Ratings (VAX Only)

CPU Type	Capacity	Comments
MicroVAX II	80	
MicroVAX 3300, 3400	130	CPU limited (embedded adapter)
MicroVAX 3500, 3600, 3800, 3900	120	DELQA limited
VAXstation 3520 VAXstation 3540	45	
VAXstation MicroVAX 2000	20	
VAXstation MicroVAX 3100	45	
VAXstation 4000–60	45	
VAXstation 4000–90	325	SGEC (CPU limited)
VAX 4000–100 MicroVAX 3100–90	400	SGEC (CPU limited)
VAX 4000–200	45	
VAX 4000–300	325	SGEC (CPU limited)
VAX 4000–400 VAX 4000–600	400	SGEC (CPU limited)
VAXft 3000	45	
VAX 11/750	45	
VAX 11/780	70	
VAX 11/785	100	Assume a DELUA
VAX 8200 VAX 8250 VAX 8300 VAX 8350	60	
VAX 85*nn*	340	Assume a DEBNI
VAX 8600 VAX 8650	100	Assume a DELUA
VAX 85*nn* VAX 8700 VAX 88*nn*	340	Assume a DEBNI
VAX 6000–200 VAX 6000–300	200	CPU limited
VAX 6000–400	400	Assume a DEMNA
VAX 6000–500	400	Assume a DEMNA
VAX 6000–600	400	Assume a DEMNA
VAX 7000	400	Assume a DEMNA
VAX 9000	400	Assume a DEMNA

5.1.3 Dual-Pathed Disks and Tapes

A **dual-pathed device** is a disk or tape that is accessible to all the computers in the cluster, not just to the computers that are physically connected to the device. The term "dual-pathed" refers to the two paths through which computers can access a device to which they are not directly connected. If one path fails, the device is accessed over the other path. (Note that with a dual-ported MASSBUS device, a computer directly connected to the device always accesses it locally.) Disks and tapes must be dual pathed between the same type of controllers.

Dual-pathed devices can be any of the following:

- Dual-ported HSC disks or tapes
- Dual-pathed DSA disks or tapes on local UDA, KDA, KDM, and KDB controllers
- DSSI connected integrated storage elements (ISEs)
- Dual-ported MASSBUS disks

5.1.3.1 Dual-Pathed HSC Disks and Tapes

By design, HSC disks and tapes are accessible by all VMScluster nodes that are connected to the same star coupler. Therefore, if they are dual ported, they are automatically dual pathed. Computers connected by CI can access a dual-pathed HSC device by way of a path through either HSC subsystem connected to the device. If one HSC subsystem fails, access fails over to the other subsystem.

You can control failover using software-controllable port selection, as described in Section 5.1.4.1.

Additionally, for each dual-ported HSC device, you can control failover to a specific port using the port select buttons on the front of each drive. By pressing either port select button (A or B) on a particular drive, you can cause the device to fail over to the specified port.

With the port select buttons, you can select alternate ports to balance the device controller work load between two HSC subsystems. For example, you can set half of your disks to use port A and the other half to use port B. The port select buttons also enable you to fail over all the devices manually to an alternate port when you anticipate the shutdown of one of the HSC subsystems.

5.1.3.2 Dual-Pathed DSA Disks and Tapes on Local UDA, KDA, and KDB Controllers

A dual-pathed DSA disk or tape can be failed over between the two computers that serve it to the cluster, provided that:

- The same device controller letter is generated and the same allocation class is specified on each computer, with the result that the device has the same name on both systems.
- Both computers are running the MSCP server for disks the TMSCP server for tapes, or both.

Caution

Failure to observe these requirements can endanger data integrity.

However, because a DSA device can be on line to one controller at a time, only one of the computers can use its local connection to the device. The second computer accesses the device through the MSCP or the TMSCP server. If the computer that

is currently serving the device fails, the other computer detects the failure and fails the device over to its local connection. The device is thereby made available to the cluster once more.

Note

A dual-pathed DSA disk cannot be used as a system disk for a directly connected CPU.

Software-controllable port selection is described in Section 5.1.4.1.

5.1.4 Dual-Pathed VAX 6000 Console Tapes (VAX Only)

VAX

If your VAXcluster system includes two or more VAX 6000 computers, you must ensure the TK console tape drives have names that are unique across the cluster so that naming conflicts do not occur.

Duplicate names are more probable with VAX 6000 computers because TK console tape drives (located in the VAX 6000 cabinet) are automatically named either MUA6 or MUB6. Thus, when you configure a VAXcluster system with more than one VAX 6000 computer, multiple TK console tape drives are likely to have the same name.

You must ensure that any tape drive that is dual pathed between two computers is identified by a unique name that includes a tape allocation class. A tape allocation class name is specified as a numeric value from 0 to 255 followed by the device name, as shown in the following syntax:

$tape-allocation-class$device-name

For example, 1MUA6, 1MUB6, 2MUA6 are all unique device names. The first two have the same tape allocation class but have different controller letters (A and B, respectively). The third device has a different tape allocation class than the first two.

Consider the following methods to ensure a unique access path to VAX 6000 TK console tape drives:

- Set the TAPE_ALLOCLASS system parameter to a unique value on each VAX 6000 system.

  When each VAX 6000 computer in the VAXcluster system has a different TAPE_ALLOCLASS value, you do not need to change the unit number of any of the TK tape drives. However, with this method, note that the tape drives are not TMSCP served across the VAXcluster system. Access to a TK console tape drive is possible only through the VAX 6000 in which it resides. In addition, if the VAX 6000 node becomes unavailable, the tape is also unavailable because there is no tape failover.

- Set the TK console tape unit number to a unique value on each VAX 6000 system.

  For VAXcluster systems in which tapes must be TMSCP served across the cluster, the tape controller letter and unit number of these tape drives must be unique clusterwide and must conform to the cluster device-naming conventions. If controller letters and unit numbers are unique clusterwide, the TAPE_ALLOCLASS system parameter can be set to the same value on multiple VAX 6000 systems.

The unit number of the TK console drives is controlled by the BI bus unit number plug of the TKB70 controller in the VAX 6000 BI backplane. A Digital Services technician should change the unit number so that it is unique from all other controller cards in the BI backplane. The unit numbers available are in the range of 0 to 15 (the default value is 6).

- For VAXcluster systems configured with only two VAX 6000 computers, set up the console tapes with different controller letters.

  If your VAXcluster configuration contains only two VAX 6000 computers, contact a Digital Services technician to move the TKB70 controller card from the BI backplane in one of the VAX computers to the other VAX computer. Moving the controller card has the effect of changing the controller letter of the tape drive without changing the unit number (for example, MUA6 becomes MUB6). (Note that the tape drives can have the same unit number.)♦

5.1.4.1 Specifying a Preferred Path

The operating system supports specifying a preferred path for DSA disks, including RA series disks and disks that are accessed through the MSCP server. (This function is not available for tapes.) If a preferred path is specified for a disk, the MSCP disk class drivers (DUDRIVER and DSDRIVER) use that path as their first attempt to locate the disk and bring it on line with a DCL command MOUNT or failover of an already mounted disk.

In addition, you can initiate failover of a mounted disk to force the disk to the preferred path or to use load-sharing information for disks accessed by MSCP servers.

The preferred path is specified by a $QIO function (IO$_SETPRFPTH), with the P1 parameter containing the address of a counted ASCII string (.ASCIC). This string is the node name of the HSC or OpenVMS system that is to be the preferred path. The node name must match an existing node that is known to the local node, and, if it is an OpenVMS system, it must be running the MSCP server. This function does not move the disk to the preferred path. For more information about the use of the IO$_SETPRFPTH function, refer to the *OpenVMS I/O User's Reference Manual*. See also the PREFER.MAR program in SYS$EXAMPLES.

5.1.4.2 DSSI Connected ISEs

In a DSSI VMScluster system, a DSSI bus can connect as many as eight nodes that can be ISEs or host CPU interfaces. In a dual-host configuration with two CPUs sharing a DSSI bus, as many as six ISEs can be connected between DSSI controllers (see Section 2.3). Simultaneously accessible to both servers, these storage elements can be served to satellites. If one disk server fails, access fails over to the other server and applications continue to run. Any DSSI connected ISE can be used as a system disk. Note that, because most failures occur in system enclosures, you should try to locate the system disk in a storage expansion box, which has a dedicated power supply.

5.1.4.3 Dual-Ported MASSBUS Disks (VAX Only)

VAX

On VAX systems, a dual-ported MASSBUS disk can be connected between two computers if it has the same controller letter and allocation class on both.

Before mounting the disk, enter the DCL command SET DEVICE in the following format on both computers:

```
SET DEVICE/DUAL_PORT device-name
```

Note

A MASSBUS disk can be used either as a dual-ported disk or as a system disk, but not as both.

In clusters with more than two computers, you can set up a dual-ported MASSBUS disk to be cluster accessible through the MSCP server on either or both computers to which the disk is connected. Be sure, however, *not* to use the SYSGEN commands AUTOCONFIGURE or CONFIGURE to configure a dual-ported MASSBUS disk that is already available on the computer through the MSCP server. Establishing a local connection to the disk when a remote path is already known creates two uncoordinated paths to the same disk. Use of these two paths can endanger data integrity on any disk that is mounted on the drive.

If the local path to the disk is not found during the system bootstrap procedure, the MSCP server path from the remote computer is the only available access to the drive. The local path is not found during a boot if any of the following conditions exists:

- The port select switch for the drive is not enabled for the local computer.
- The disk, cable, or adapter hardware for the local path is broken.
- There is sufficient activity on the other port to "mask" the existence of the port.
- The computer is booted in such a way that the SYSGEN command AUTOCONFIGURE ALL in the site-independent startup procedure (SYS$SYSTEM:STARTUP.COM) was not executed.

Use of the disk is still possible through the MSCP server path.

Caution

Under these conditions, do not attempt to add the local path back into the system I/O database using the SYSGEN commands AUTOCONFIGURE or CONFIGURE. The SYSGEN utility is currently unable to detect the presence of the disk's MSCP path and would build a second set of data structures incorrectly. Subsequent events could lead to incompatible and uncoordinated file operations, which might endanger data integrity.

Note that if the disk is not dual ported or is never MSCP served on the remote computer, this restriction does not apply.

To recover the local path to the disk, you must reboot the computer connected to that local path.♦

5.2 Cluster Device-Naming Conventions

To manage cluster devices properly, you must understand the conventions used to identify them. Every cluster device is identified by a unique name, which provides a reliable way to access it in the cluster.

Disk and tape devices that are local to a single VMScluster computer can be accessed by that computer through the traditional device name (for example, DUA1 or MUB6) or through a cluster device name in the format *node-name*$*device-name* (for example, JUPITR$DUA1).

However, a device that is dual pathed between two computers, HSC controllers, or DSSI ISEs must be identified by a unique, path-independent name that includes an **allocation class**. The allocation class is a numeric value from 1 to 255 that the system manager assigns to a pair of hosts (CPUs or HSC controllers) and the dual-pathed devices that the hosts make available to other nodes in the VMScluster. You use one of the following formats to create a disk or tape device name that is unique across the cluster:

$allocation-class$device-name
$tape-allocation-class$device-name

For example, the allocation class device name 1DJA17 identifies a disk that is dual ported between two computers or HSC subsystems that both have an allocation class value of 1. Similarly, the allocation class device name 1MUA12 identifies a dual-pathed tape. Users access the 1DJA17 or 1MUA12 dual-pathed devices through either of the hosts. In this way, if one host with allocation class 1 is not available, the user can gain access to a device specified by that allocation class through the other host of the allocation class.

Each time a computer that is not connected directly to such a device tries to access it, the choice of which path to take is made arbitrarily; no specific path is ever guaranteed. Because the access path is chosen without regard to the names of the computers or HSC subsystems serving the device, an allocation class device name is required to identify the device uniquely.

5.2.1 Rules for Specifying Allocation Class Values

Allocation classes play an important role in determining strategies for configuring and naming disks and tapes. In fact, the operating system uses allocation class values, device types, and unit numbers to determine the configuration of cluster devices.

The following rules apply for specifying allocation class values:

- Computers, HSC subsystems, or DSSI ISEs to which a dual-pathed device is connected must have the same nonzero allocation class value.
- All cluster-accessible devices on computers with a nonzero allocation class value must have unique names throughout the cluster. For example, if two computers have the same allocation class value, it is invalid for both computers to have a disk named DJA0 or a tape named MUA0. This restriction also applies to HSC subsystems.
- Single-ported devices with an allocation class value of 0 can have the same unit number on different computers.
- Systems that serve HSC or DSSI disks to other nodes in the VMScluster must have the same allocation class as the HSC storage subsystems or DSSI ISEs that they serve.

The default allocation class value is 0. Any computer that is serving either multihost (HSC or DSSI) or dual-pathed disks or tapes must be assigned a nonzero allocation class value. An allocation class value of 0 is appropriate only when serving a local single-pathed disk. All of the following must have a nonzero allocation class value:

- HSC subsystems
- Computers that serve HSC disks and tapes
- DSSI ISEs

- Computers connected to dual-pathed disks and tapes
- Members of shadow sets

Caution

Failure to set allocation class values correctly can endanger data integrity and cause locking conflicts that suspend normal cluster operations.

To assign an allocation class value to a VMScluster computer that supports dual-pathed devices:

1. Specify the value with the ALLOCLASS and TAPE_ALLOCLASS system parameters in the MODPARAMS.DAT file.
2. Edit the root directory [SYS*n*.SYSEXE]MODPARAMS.DAT on each node that boots from the system disk. Example 5–1 shows an example of a MODPARAMS.DAT file. The entries are hypothetical and should be used only for example purposes, not as suggestions for specific parameter settings.

Example 5–1 MODPARAMS.DAT file

```
!
! Site-specific AUTOGEN data file.  In a VMScluster where
! a common system disk is being used, this file should reside
! in SYS$SPECIFIC:[SYSEXE], not a common system directory.
!
! Add modifications that you want to make to AUTOGEN's
! hardware configuration data, system parameter calculations, and
! page, swap, and dump file sizes to the bottom of this file.
!
SCSNODE="NODE01"
SCSSYSTEMID=99999
NISCS_LOAD_PEA0=1
VAXCLUSTER=2
MSCP_LOAD=1
MSCP_SERVE_ALL=1
ALLOCLASS=1
TAPE_ALLOCLASS=1
```

3. Invoke AUTOGEN to reset the system parameter values:

```
$ @SYS$UPDATE:AUTOGEN
```

4. Shut down and reboot the entire cluster in order for the new values to take effect. (Shutting down and rebooting the VMScluster is described in Section 7.6.)

Disk class drivers (such as DUDRIVER and DKDRIVER) connect to a given controller as long as the allocation class remains the same. If you accidentally change a device's allocation class on a running VMScluster system, the driver cannot connect to the device. Disabling the connection is done as a precautionary measure to prevent data corruption without loss of system availability. Drivers can reconnect to the disk controller if the allocation class is changed back to its original value, or if the VMScluster is shut down and rebooted.

You must set allocation class values on HSC subsystems while the cluster is shut down. To assign a disk allocation class for an HSC subsystem, specify the value using the HSC console to enter a command in the following format:

```
SET ALLOCATE DISK allocation-class-value
```

To assign a tape allocation class, enter a command in the following format:

```
SET ALLOCATE TAPE tape-allocation-class-value
```

See Section 7.6.3 for an example of setting the HSC allocation class. For complete information about the HSC console commands, refer to the HSC hardware documentation.

For information and an example to help you change a DSSI subsystem allocation class, see Section 7.6.4.

Note that multihost disks and tapes must have matching allocation class values on both the CPU and the controller to be served.

5.2.2 Sample Configurations with Named Devices

Figure 5–2 and Figure 5–3 show how cluster device names are specified for SDI and STI devices that are:

- Dual pathed between HSC devices
- Dual pathed between computers

Figure 5–2 shows a VMScluster configuration with a dual-pathed HSC disk. The disk device name (1DJA17) and tape device name (1MUB6) are derived using the allocation class of the controller. The ALLOCLASS and TAPE_ALLOCLASS system parameters are set to 1 in the MODPARAMS.DAT file on JUPITR and on SATURN. JUPITR and SATURN can access the disk or tape through either HSC subsystem VOYGR1 or VOYGR2.

Figure 5–2 Disk and Tape Dual-Pathed Between HSC Controllers

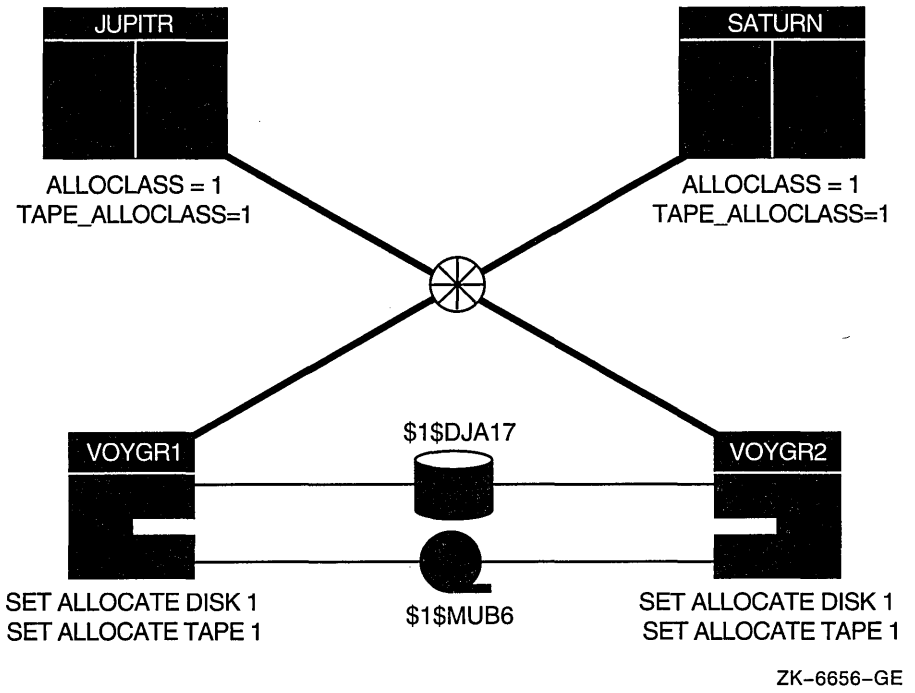
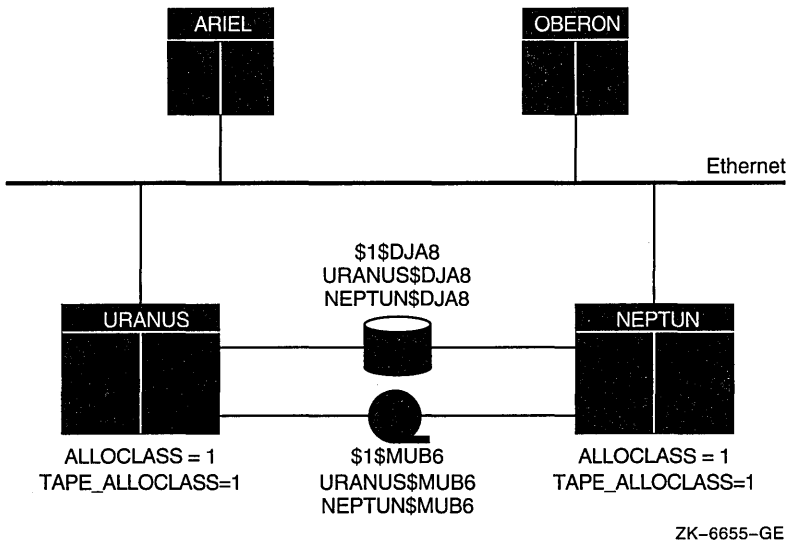


Figure 5–3 shows a DSA disk and tape that are dual pathed between two computers.

Figure 5–3 Disk and Tape Dual-Pathed Between VAX Computers

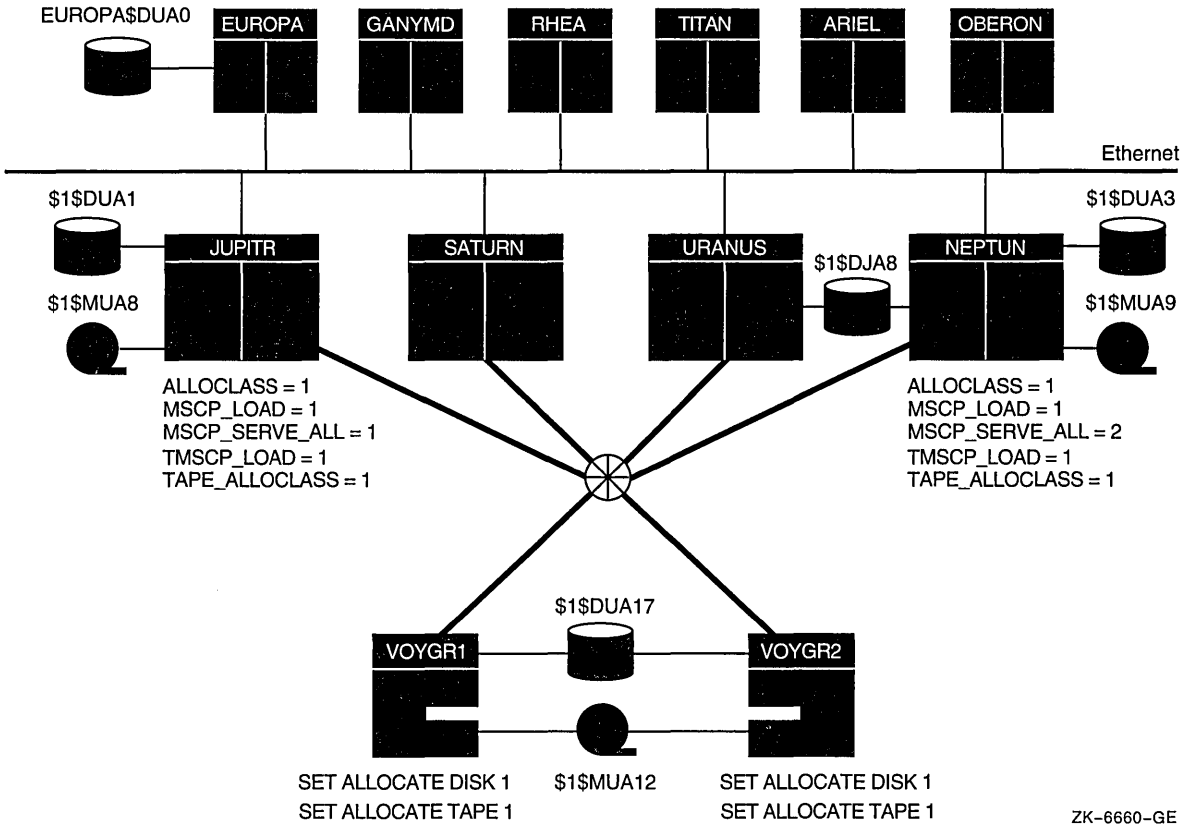


URANUS and NEPTUN can access the disk either locally or through the other computer's MSCP server. When satellites ARIEL and OBERON access the disk using the allocation class device name 1DJA8, access is made arbitrarily through either URANUS or NEPTUN. If, for example, the node URANUS has been shut down, the satellites can access the devices through NEPTUN. When

URANUS reboots, access is again made arbitrarily through either URANUS or NEPTUN.

Figure 5–4 shows how device names are typically specified in a mixed-interconnect cluster. This figure also shows how relevant system parameter values are set in each CI computer's MODPARAMS.DAT file. Note that the values shown for JUPITR are the same for SATURN and URANUS but that NEPTUN has a different value for MSCP_SERVE_ALL.

Figure 5–4 Device Names in a Mixed-Interconnect Cluster



In this configuration, a disk and a tape are dual pathed to the HSC subsystems named VOYGR1 and VOYGR2; these subsystems are connected to JUPITR, SATURN, URANUS and NEPTUN through the star coupler. The MSCP and TMSCP servers are loaded on all four computers (MSCP_LOAD = 1, TMSCP_LOAD = 1) and the ALLOCLASS and TAPE_ALLOCLASS parameters are set to the same value (1) on these computers and on both HSC subsystems. But MSCP_SERVE_ALL is set to 1 on only JUPITR, SATURN, and URANUS. Therefore, only these three computers can serve the devices on VOYGR1 and VOYGR2 to satellites. Because MSCP_SERVE_ALL is set to 2 on NEPTUN, NEPTUN can serve only its local disks.

The HSC disk and tape have allocation class names in the form 1*ddcu*. For example, disk DUA17 is named 1DUA17. On computers connected by CI, the operating system software would also recognize the disk as JUPITR$DUA17 and as either VOYGR1$DUA17 or VOYGR2$DUA17. On satellites, it would recognize the disk as JUPITR$DUA17 or as 1DUA17. This example shows why you should always use an allocation class name like 1DUA17 when you configure

cluster devices: the allocation class name is the only name that all computers recognize at all times.

Note that, for optimal availability, two or more CI connected computers should serve HSC disks and tapes to the cluster.

5.3 Shared Disks

A **shared disk** is a disk that is mounted on a cluster-accessible device by one or more VMScluster computers. Data disks can be shared between AXP and VAX computers in a VMScluster system. System disks can be shared between AXP and VAX processors. However, VAX systems cannot boot from AXP system disks, and AXP systems cannot boot from VAX system disks. Shared disks play a key role in common-environment clusters, because when you place data files or command procedures on a shared disk, computers can share a single copy of each common file (see Chapter 4). Note, however, that a shared disk is a single point of failure for data access by the computers sharing the disk. (See Section 5.6 for information about using volume shadowing for data availability.)

To mount cluster-accessible disks that are to be shared among all computers, specify the same MOUNT command on each computer or specify the MOUNT command with the /CLUSTER qualifier on one or more computers. (Typically, you specify MOUNT/SYSTEM on all computers.) When you execute MOUNT /CLUSTER on one computer, the disk is mounted on every computer that is active in the cluster at the time the command executes. Note that only system or group disks can be mounted across the cluster. Thus, if you specify MOUNT/CLUSTER without the /SYSTEM or /GROUP qualifier, /SYSTEM is assumed. Also note that each cluster disk mounted with the /SYSTEM, /GROUP, or /SHARED qualifiers must have a unique volume label.

If you want to mount a shared disk on some but not all VMScluster computers, execute the same MOUNT command (without the /CLUSTER qualifier) on each computer that shares the disk.

For example, suppose you want all the computers in a three-member cluster to share a disk named COMPANYDOCS. To share the disk, each of the three computers can execute identical MOUNT commands, or one of the three computers can mount COMPANYDOCS using the MOUNT/CLUSTER command, as follows:

```
$ MOUNT/CLUSTER/NOASSIST $1$DUA4: COMPANYDOCS
```

If you want just two of the three computers to share the disk, those two computers must both mount the disk with the same MOUNT command. For example:

```
$ MOUNT/SYSTEM/NOASSIST $1$DUA4: COMPANYDOCS
```

To mount the disk at startup time, include the MOUNT command either in a common command procedure that is invoked at startup time or in the computer-specific startup command file.

5.4 Configuring Cluster Disks

To configure cluster disks, you can create command procedures to set up and mount them. You may want to include commands that set up and mount cluster disks in a separate command procedure file that is invoked by a site-specific SYSTARTUP procedure. Depending on your cluster environment, you can set up your command procedure in either of the following ways:

- As a separate file specific to each computer in the cluster
- As a common computer-independent file

You can set up the common procedure as a shared file on a shared disk, or you can make copies of the common procedure and store them as separate files. With either method, each computer can invoke the common procedure from the site-specific SYSTARTUP procedure.

The MSCPMOUNT.COM file in the SYS$EXAMPLES directory on your system is a sample common command procedure that contains commands typically used to mount cluster disks. The example includes comments explaining each phase of the procedure.

5.5 Rebuilding Cluster Disks

To minimize disk I/O operations (and thus improve performance) when files are created or extended, the OpenVMS file system maintains a cache of preallocated file headers and disk blocks.

If a disk is dismounted improperly—for example, if a system fails or is removed from a cluster without running SYS$SYSTEM:SHUTDOWN.COM—this preallocated space becomes temporarily unavailable. When the disk is remounted, MOUNT scans the disk to recover the space. This is called a **disk rebuild operation**.

On a nonclustered computer, the scan operation merely prolongs the boot process. In a VMScluster system, however, this operation can degrade response time for all user processes in the cluster. While the scan is in progress on a particular disk, most activity on that disk is blocked. User processes that attempt to read or write to files on the disk can experience delays of several minutes or longer, especially if the disk contains a large number of files or has many users.

Because the rebuild operation can delay access to disks during the startup of any VMScluster computer, Digital recommends that procedures for mounting cluster disks use the /NOREBUILD qualifier. When MOUNT/NOREBUILD is specified, disks are not scanned to recover lost space, and users experience minimal delays while computers are mounting disks.

Rebuilding System Disks

System disks are especially critical in this regard because most system activity requires access to a system disk. When a system disk rebuild is in progress, very little activity is possible on any computer that uses that disk. Unlike other disks, the system disk is automatically mounted early in the boot sequence. If a rebuild is necessary, and if the value of the system parameter ACP_REBLDSYSD is 1, the system disk is rebuilt during the boot sequence. (The default setting of 1 for the ACP_REBLDSYSD system parameter specifies that the system disk should be rebuilt.) Exceptions are as follows:

- In local area and mixed-interconnect clusters, however, the ACP_REBLDSYSD parameter should normally be set to 0 on all satellites. This

setting prevents them from rebuilding a system disk when it is mounted early in the boot sequence and eliminates delays caused by such a rebuild when satellites join the cluster.

- In large clusters, a substantial amount of system disk space (some for each computer) might be preallocated to caches, and, if many computers abruptly leave the cluster (for example, during a power failure), this space can become temporarily unavailable. Thus, ACP_REBLDSYSD on disk servers in local area and mixed-interconnect clusters with many computers should be set to the default value of 1, and procedures that mount disks on the boot servers should use the /REBUILD qualifier. While these measures can make boot server rebooting more noticeable, they ensure that system disk space is available after an unexpected shutdown.

Once the cluster is up and running, system managers can submit one or more batch procedures that execute SET VOLUME/REBUILD commands to recover lost disk space. Such procedures can run at a time when users would not be inconvenienced by the blocked access to disks (for example, between midnight and 6 a.m. each day). Because the SET VOLUME/REBUILD command determines whether a rebuild is needed, the procedures can execute the command for each disk that is usually mounted. Note that the procedures run more quickly and cause less delay in disk access if they are executed on powerful computers. Moreover, several such procedures, each of which rebuilds a different set of disks, can be executed simultaneously.

Caution

If any of the following conditions are true when mounting disks, it is essential to run a procedure with SET VOLUME/REBUILD commands on a regular basis to rebuild the disks:

- Disks are mounted with the MOUNT/REBUILD command
- The ACP_REBLDSYSD system parameter is set to 0
- Both of these conditions exist

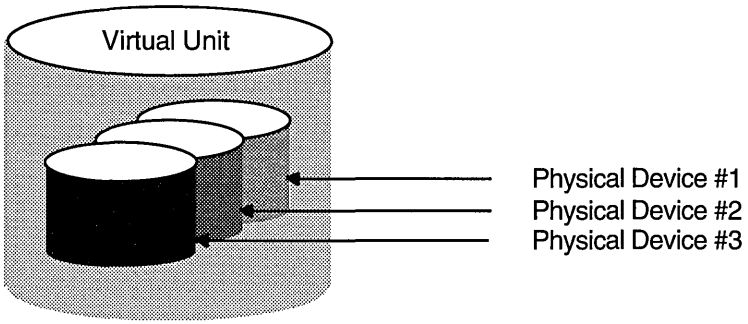
Failure to rebuild disk volumes can result in a loss of free space and in subsequent failures of applications to create or extend files.

5.6 Shadowing Disks Across a VMScluster

Volume shadowing (sometimes referred to as disk mirroring) achieves high data availability by duplicating data on multiple disks. If one disk fails, the remaining disk or disks can continue to service application and user I/O requests.

Volume Shadowing for OpenVMS software is an implementation of RAID 1 (redundant arrays of inexpensive disks) technology. Volume Shadowing for OpenVMS prevents a disk device failure from interrupting system and application operations. By duplicating data on multiple disks, volume shadowing transparently prevents your storage subsystems from becoming a single point of failure because of media deterioration, communication path failure, or through controller or device failure.

You can mount one, two, or three compatible disk volumes to form a **shadow set**. Each disk in the shadow set is known as a shadow set **member**. Volume Shadowing for OpenVMS logically binds the shadow set devices together and represents them as a single virtual device called a **virtual unit**. This means that the multiple members of the shadow set, represented by the virtual unit, appear to operating systems and users as a single, highly available disk.



ZK-5156A-GE

Applications and users read and write data to and from a shadow set using the same commands and program language syntax and semantics that are used for nonshadowed I/O operations. System managers manage and monitor shadow sets using the same commands and utilities they use for nonshadowed disks. The only difference is that access is through the virtual unit, not to individual devices. *Volume Shadowing for OpenVMS* describes the shadowing product capabilities in detail.

5.6.1 Supported Configurations

Volume Shadowing for OpenVMS software provides data availability across the full range of OpenVMS configurations—from single nodes to large VMScluster systems—so you can provide data availabililty where you need it most.

For a single workstation or a large data center, valid shadowing configurations include:

- All MSCP compliant DSA drives:
  - On the same local controller
  - On different local controllers
  - On controllers local to different OpenVMS hosts and accessed through the OpenVMS MSCP server to the VMScluster system
- All DSSI devices.
- All Digital Equipment Corporation SCSI disks and controllers, and some third-party SCSI devices that implement READL (read long) and WRITEL (write long) commands and use the SCSI disk driver (DKDRIVER).

  SCSI disks that do not support READL and WRITEL are restricted because these disks do not support the shadowing data repair (disk bad block errors) capability. Thus, using unsupported SCSI disks can cause members to be removed from the shadow set.

Devices that cannot be shadowed include:

- MicroVAX 2000 RD disks
- Older disk devices (such as MASSBUS, RK07, RL02)
- DECram disks

There are no restrictions on the location of shadow set members beyond the valid disk configurations defined in the SPD for the OpenVMS operating system (SPD 25.01.*xx*), the VAXcluster Software for OpenVMS VAX *Software Product Description* (SPD 29.78.*xx*), and the VMScluster Software for OpenVMS AXP *Software Product Description* (SPD 42.18.*xx*), as appropriate.

You can shadow data disks and system disks. Thus, a system disk need not be a single point of failure for any system that boots from that disk. System disk shadowing becomes especially important for VMScluster systems that use a *common* system disk from which multiple computers boot.

You can mount a maximum of 130 shadow sets (up to 390 disks) in a standalone or VMScluster system. The number of shadow sets supported is independent of controller and device types. The shadow sets can be mounted as public or private volumes. Shadow sets also can be constituents of a bound volume set or a stripe set. Section 5.6.2 describes more about shadowing across VMScluster systems.

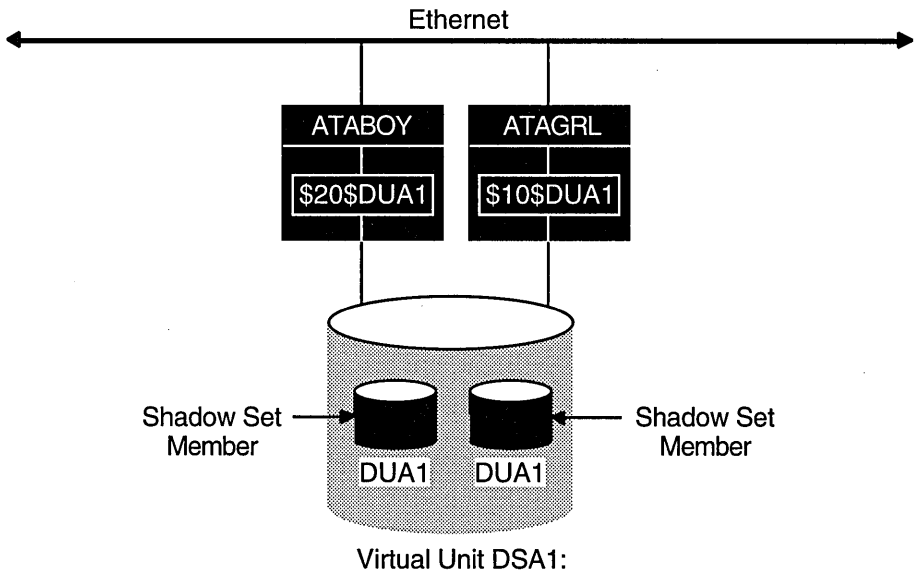
5.6.2 Shadowing Disks Across a VMScluster

The host-based implementation of volume shadowing, by making the storage subsystem independent of the host node, allows disks that are physically distant to be shadowed across a VMScluster system. The controller-independent design of phase II shadowing allows you to manage shadow sets regardless of their controller connection or location in the VMScluster system and helps provide improved data availability and very flexible configurations.

For clusterwide shadowing, members can be located anywhere in a VMScluster system and served by MSCP servers across any supported VMScluster interconnect, including the CI, Ethernet, DSSI, and FDDI. For example, VMScluster systems using FDDI can be up to 25 miles apart, which further increases the availability and disaster tolerance of a system.

Figure 5–5 shows how shadow set member units are on line to local controllers located on different nodes. In the figure, a disk volume is local to each of the nodes ATABOY and ATAGRL. The MSCP server provides access to the shadow set members over the Ethernet. Even though the disk volumes are local to different nodes, the disks are members of the same shadow set. A member unit that is local to one node can be accessed by the remote node over the MSCP server.

Figure 5–5 Shadow Sets Accessed Through the MSCP Server



ZK–2024A–GE

The shadowing software maintains virtual units in a distributed fashion on each node that mounts the shadow set in the VMScluster system. Volume shadowing can provide distributed shadowing because the virtual unit is maintained and managed by each node that has the shadow set mounted.

For shadow sets that are mounted on a VMScluster system, mounting or dismounting a shadow set on one node in the cluster does not affect applications or user functions executing on other nodes in the system. For example, you can dismount the virtual unit from one node in a VMScluster system and leave the shadow set operational on the remaining nodes on which it is mounted.

Other shadowing notes:

- If an individual disk volume is already mounted as a member of an active shadow set, the disk volume cannot be mounted as a standalone disk on another node.
- System disks can be shadowed. All nodes booting from shadowed system disks must have shadowing licensed, mounted, and enabled.
- Volume Shadowing for OpenVMS does not support the shadowing of quorum disks. This is because volume shadowing makes use of the OpenVMS distributed lock manager, and the quorum disk must be utilized before locking is enabled.

6

Setting Up and Managing Cluster Queues

You can use one or several queue managers to manage batch and print queuing activity for an entire VMScluster system. Although a single queue manager is sufficient for most systems, multiple queue managers can be useful for distributing the batch and print work load across nodes in the cluster.

Note

In order to share queuing on dual-architecture VMScluster systems, the AXP systems must be running OpenVMS AXP Version 1.5, and the VAX systems must be running VMS Version 5.5–2 (not Version A5.5–2). The VAX systems cannot run VMS Version 5.5 or OpenVMS VAX Version 6.0.

This chapter discusses queuing topics specific to VMScluster systems. Because queues in a VMScluster system are established and controlled with the same commands used to manage queues on a standalone computer, the discussions in this chapter assume some knowledge of queue management on a standalone system, as described in the *OpenVMS System Manager's Manual.*

6.1 Clusterwide Queues

Once the batch and print queue characteristics are set up, the system manager can rely on the distributed queue manager to make queues available across the cluster. Users can submit jobs to any queue in the cluster because VMScluster computers can share batch and print queues. System managers can also set up generic batch queues that distribute batch processing work loads among computers.

Cluster transitions are handled so that the queuing system is not affected when a node enters or leaves the cluster. The queue manager automatically fails over to another node (VAX or AXP) if the node on which it is running leaves the VMScluster system.

- You can specify more than one node on which a queue can run. This allows the queue to fail over to another node if the node on which the queue is running leaves the cluster.
- Nodes that are newly added to the cluster are served automatically by the queue manager. The system manager does not need to enter a command explicitly to start queuing on the new node.

By default, the queuing system automatically restarts after reboot. This is because when you start the queuing system, the characteristics you define are stored in a queue database. Upon reboot, the operating system automatically restores the queuing system with the parameters defined in the database. Thus, you do not have to include commands in your startup command procedure for queuing.

6.2 Controlling Clusterwide Queues

To control queues, the queue manager maintains a clusterwide queue database that stores information about queues and jobs. Whether you use one or several queue managers, only one queue database is shared across the cluster. Keeping the information for all processes in one database allows jobs submitted from any computer to execute on any queue (provided that the necessary mass storage devices are accessible).

Starting a Queue Manager and Creating the Queue Database

You start up a queue manager using the START/QUEUE/MANAGER command as you would on a standalone computer. However, in a VMScluster system, you might also provide a failover list and a unique name for the queue manager in the following format:

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

The following command example shows how to start a queue manager:

```
$ START/QUEUE/MANAGER/NEW_VERSION/ON=(GEM,STONE,*)
```

In this command:

- START/QUEUE/MANAGER creates a single, clusterwide queue manager named SYS$QUEUE_MANAGER.
- /NEW_VERSION creates a new queue database in SYS$COMMON:[SYSEXE] that consists of the following three files:

  QMAN$MASTER.DAT (master file)
  SYS$QUEUE_MANAGER.QMAN$QUEUES (queue file)
  SYS$QUEUE_MANAGER.QMAN$JOURNAL (journal file)

  Use the /NEW_VERSION qualifier only on the first invocation of the queue manager, or if you want to create a new queue database. If you want to locate the queue database files on other devices or directories, refer to the *OpenVMS System Manager's Manual.*
- /ON=*(node-list)* is an optional qualifier that specifies an ordered list of nodes that can claim the queue manager if the node running the queue manager should exit the cluster. In the preceding example, the queue manager process will start on node GEM. If the queue manager is running on node GEM and GEM leaves the cluster, the queue manager will fail over to node STONE.

  The asterisk wildcard (*) is specified as the last node in the node list to indicate that any remaining, unlisted node can start the queue manager, with no preferred order. Complete node names are required; you cannot specify the asterisk wildcard character as part of a node name. If you want to exclude certain nodes from being eligible to run the queue manager, do not use the asterisk wildcard character in the node list.

  See also Section 6.3.1 for information about the autostart feature.

The preceding command could also include the /NAME_OF_MANAGER=*name* qualifier, as shown in the following example:

```
$ START/QUEUE/MANAGER/NEW_VERSION/ON=(GEM,STONE,*)/NAME_OF_MANAGER=PRINT_MANAGER
```

Using the optional /NAME_OF_MANAGER qualifier allows you to assign a unique name to the queue manager. Unique queue manager names are necessary if you run multiple queue managers. Using the /NAME_OF_MANAGER qualifier causes queue and journal files to be created using the queue manager name (in this case PRINT_MANAGER) instead of the default name SYS$QUEUE_MANAGER. For example, the preceding command creates these files:

```
QMAN$MASTER.DAT
PRINT_MANAGER.QMAN$QUEUES
PRINT_MANAGER.QMAN$JOURNAL
```

Starting Additional Queue Managers

Running multiple queue managers balances the work load by distributing batch and print jobs across the cluster. For example, you might create separate queue managers for batch and print queues in clusters with CPU or memory shortages. This allows you to run the batch queue manager on one node and the print queue manager on a different node.

To start additional queue managers, you include the /ADD and /NAME_OF_MANAGER qualifiers on the START/QUEUE/MANAGER command. Do not specify the /NEW_VERSION qualifier.

```
$ START/QUEUE/MANAGER/ADD/NAME_OF_MANAGER=BATCH_MANAGER
```

Multiple queue managers share one QMAN$MASTER.DAT master file, but an additional queue file and journal file is created for each queue manager. The additional files are named in the format:

- *name_of_manager*.QMAN$QUEUES
- *name_of_manager*.QMAN$JOURNAL

By default, the queue database and its files are located in SYS$COMMON:[SYSEXE]. If you want to relocate the queue database files, refer to the *OpenVMS System Manager's Manual.*

Stopping and Restarting the Queuing System

The following example shows the command to stop a queue manager named PRINT_MANAGER:

```
$ STOP/QUEUE/MANAGER/CLUSTER/NAME_OF_MANAGER=PRINT_MANAGER
```

You must include the /CLUSTER qualifier on the command line whether or not the queue manager is running on a VMScluster system. If you omit the /CLUSTER qualifier, the command stops all queues on the default node without stopping the queue manager. (This has the same effect as entering the STOP /QUEUES/ON_NODE command.)

Once you enter the STOP/QUEUE/MANAGER/CLUSTER command, the queue manager remains stopped, and requests for queuing are denied until you enter the START/QUEUE/MANAGER command (without the /NEW_VERSION qualifier).

VMScluster Systems with Multiple System Disks

For VMScluster systems with multiple system disks, you must specify the locations of both the master file and the queue and journal files for systems that do not boot from the system disk where the files are located. The device and directory that you specify must be accessible across the VMScluster, and they must be defined identically in the SYS$COMMON:SYLOGICALS.COM startup command procedure on every node.

Moving Queue Database Files

The files in the queue database can be relocated from the default location of SYS$COMMON:[SYSEXE] to any disk that is mounted clusterwide or that is accessible to the computers participating in the clusterwide queue scheme. For example, you can enhance system performance by locating the database on a shared disk that has a low level of activity.

The master file QMAN$MASTER can be in a location separate from the queue and journal files, but the queue and journal files must be kept together in the same directory. The queue and journal files for one queue manager can be separate from those of other queue managers.

The directory you specify must be available to all nodes in the cluster. If the directory specification is a concealed logical name, it must be defined identically in the SYS$COMMON:SYLOGICALS.COM startup command procedure on every node in the cluster.

The *OpenVMS System Manager's Manual* contains complete information about creating or relocating the queue database files. See also Section 6.5 for a sample common procedure that sets up a VMScluster batch and print system.

6.3 Cluster Printer Queues

To establish printer queues, you must determine the type of queue configuration that best suits your VMScluster system. You have several alternatives that depend on the number and type of print devices you have on each computer and on how you want print jobs to be processed. For example, make these decisions:

- Which printer queues you want to establish on each computer
- Whether to set up any clusterwide generic queues to distribute print job processing across the cluster
- Whether to set up autostart queues for availability or improved startup time

Once you determine the appropriate strategy for your cluster, you can create your queues. Figure 6–1 shows the printer configuration for a cluster consisting of the active computers JUPITR, SATURN, and URANUS. Section 6.3.1 and Section 6.3.2 describe various methods for establishing and naming the cluster printer queues shown in this configuration.

6.3.1 Setting Up Printer Queues

You set up VMScluster printer queues using the same method that you would use for a standalone computer. However, in a VMScluster system, you must provide a unique name for each queue you create.

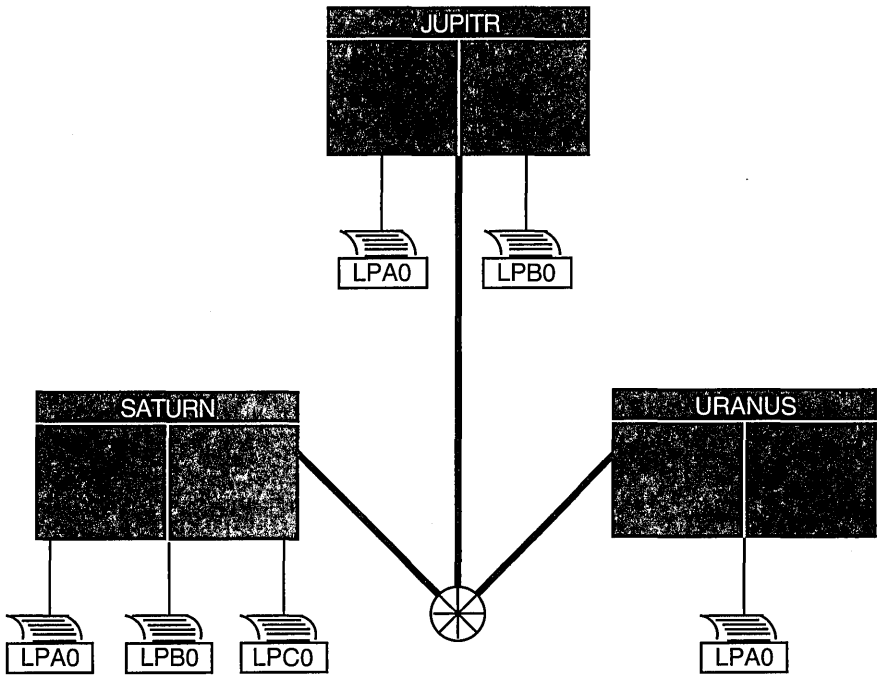
You create and name a printer queue by specifying the INITIALIZE/QUEUE command at the DCL prompt in the following format:

```
INITIALIZE/QUEUE/ON=node-name::device[/START]
[/NAME_OF_MANAGER=] queue-name
```

The /ON qualifier specifies the computer and printer to which the queue is assigned. If you specify the /START qualifier, the queue is started.

If you are running multiple queue managers, you should also specify the queue manager with the /NAME_OF_MANAGER qualifier.

Figure 6–1 Sample Printer Configuration



ZK–1631–GE

You can also use the autostart feature to simplify startup and ensure high availability of execution queue in a VMScluster. If the node on which the autostart queue is running leaves the VMScluster, the queue automatically fails over to the next available node on which autostart is enabled. Autostart is particularly useful on LAT queues. Because LAT printers are usually shared among users of multiple system or in VMScluster systems, many users are affected if a LAT queue is unavailable.

Create an autostart queue with a list of nodes on which the queue can run by specifying the DCL command INITIALIZE/QUEUE in the following format:

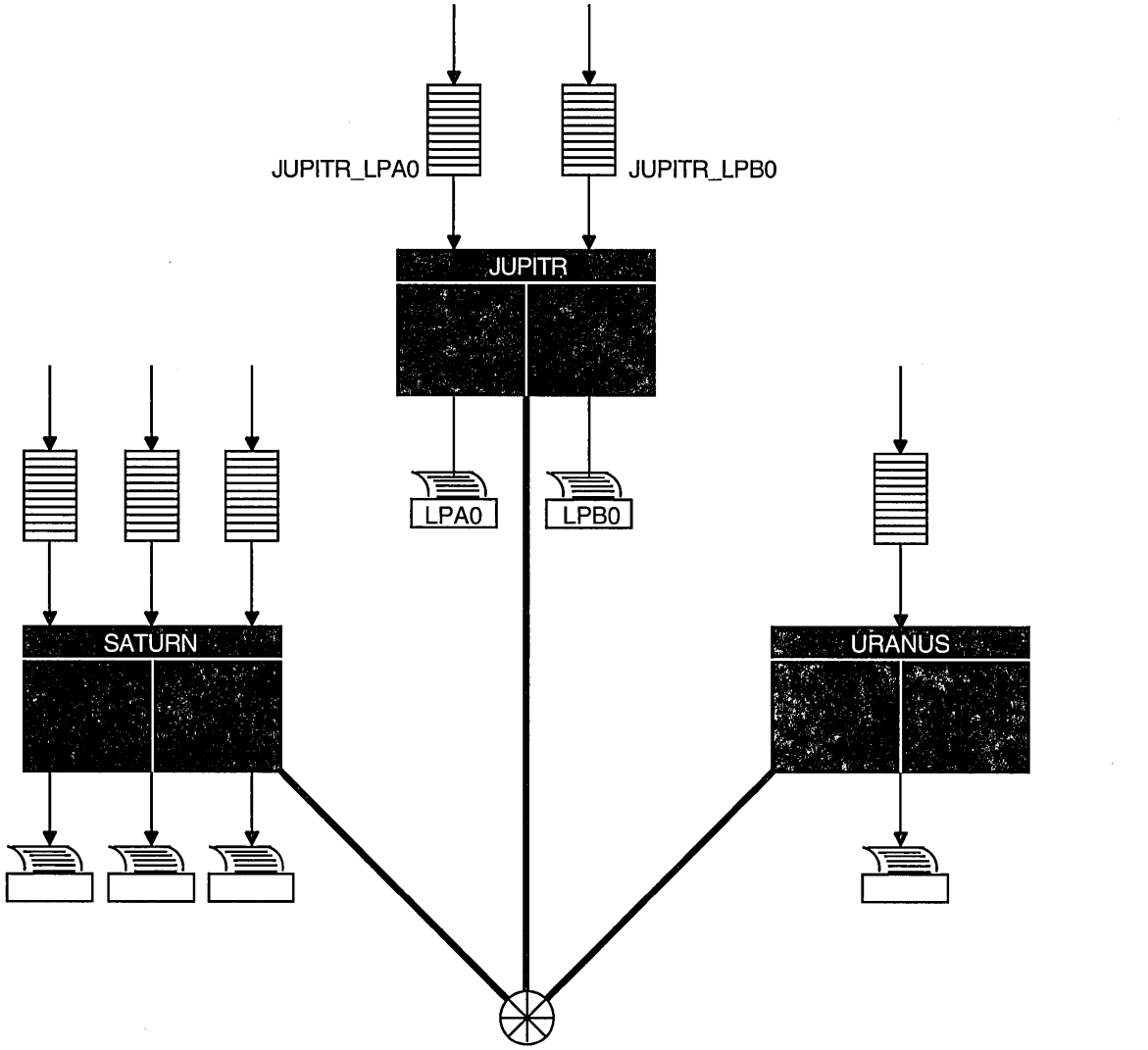
INITIALIZE/QUEUE/AUTOSTART_ON=
(node-name::device:,node-name::device:, . . .) queue-name

When you use 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 cannot begin processing jobs until the ENABLE AUTOSTART /QUEUES command is entered for a node on which the queue can run. Generic queues cannot be autostart queues. Note that the /ON and /AUTOSTART_ON qualifiers are mutually exclusive. (See Section 6.7 for information about setting the time at which autostart is disabled.)

The following commands make the local printer queue assignments for JUPITR shown in Figure 6–2 and start the queues:

```
$ INITIALIZE/QUEUE/ON=JUPITR::LPA0/START/NAME_OF_MANAGER=PRINT_MANAGER JUPITR_LPA0
$ INITIALIZE/QUEUE/ON=JUPITR::LPB0/START/NAME_OF_MANAGER=PRINT_MANAGER JUPITR_LPB0
```

Figure 6–2 Printer Queue Configuration

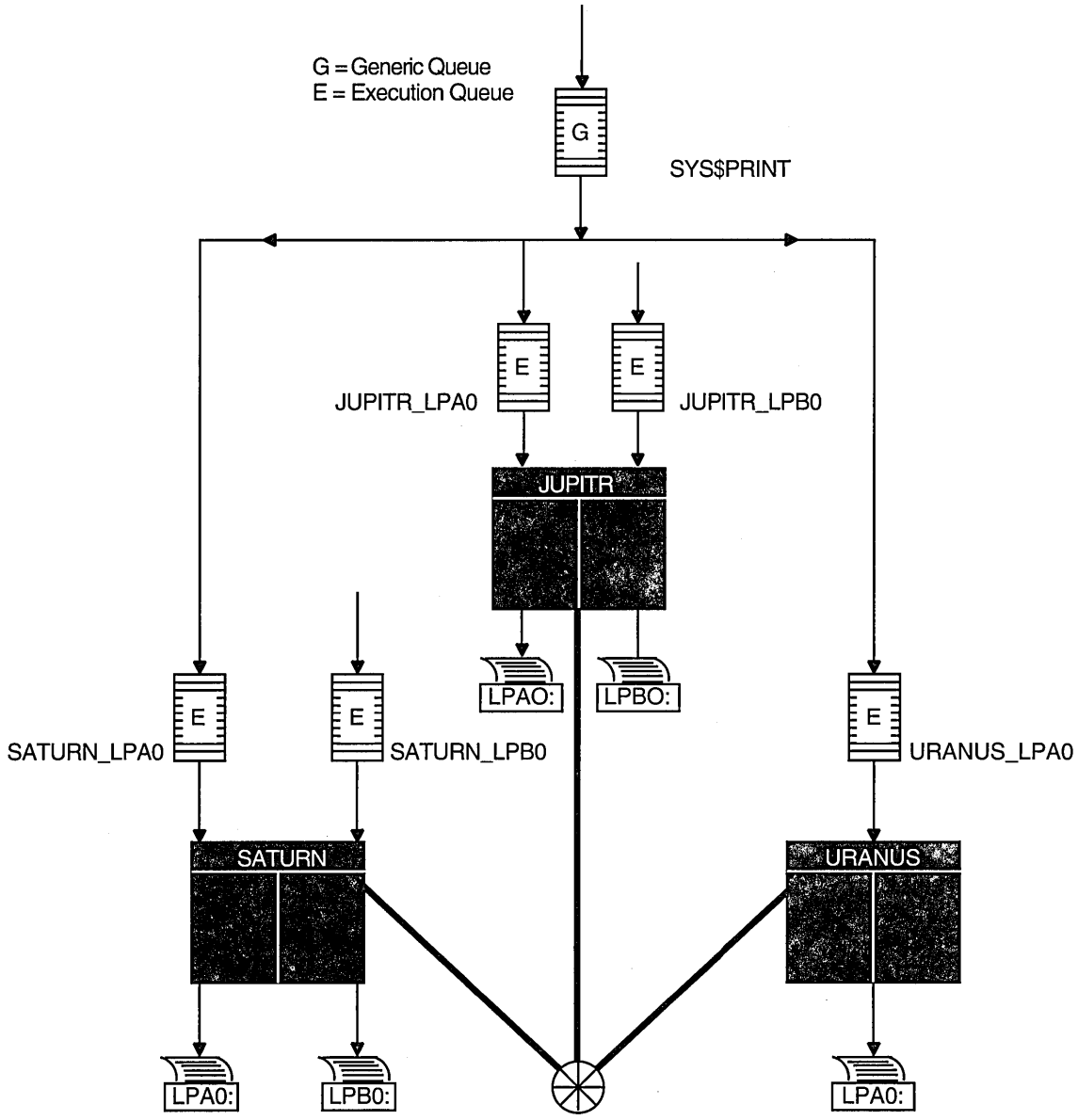


6.3.2 Setting Up Clusterwide Generic Printer Queues

The clusterwide queue database enables you to establish generic queues that function throughout the cluster. Jobs queued to clusterwide generic queues are placed in any assigned printer queue that is available, regardless of its location in the cluster. However, the file queued for printing must be accessible to the computer to which the printer is connected.

Figure 6–3 illustrates a clusterwide generic printer queue in which the queues for all LPA0 printers in the cluster are assigned to a clusterwide generic queue named SYS$PRINT.

Figure 6–3 Clusterwide Generic Printer Queue Configuration



ZK–1634–GE

The following command initializes and starts the clusterwide generic queue SYS$PRINT:

```
$  INITIALIZE/QUEUE/GENERIC=(JUPITR_LPA0,SATURN_LPA0,) -
_$ URANUS_LPA0/START  SYS$PRINT)
```

Jobs queued to SYS$PRINT are placed in whichever assigned printer queue is available. Thus, in this example, a print job from JUPITR that is queued to SYS$PRINT can be queued to JUPITR_LPA0, SATURN_LPA0, or URANUS_LPA0.

A clusterwide generic printer queue needs to be initialized and started only once. The most efficient way to start your queues is to create a common command procedure that is executed by each VMScluster computer (see Example 6–2).

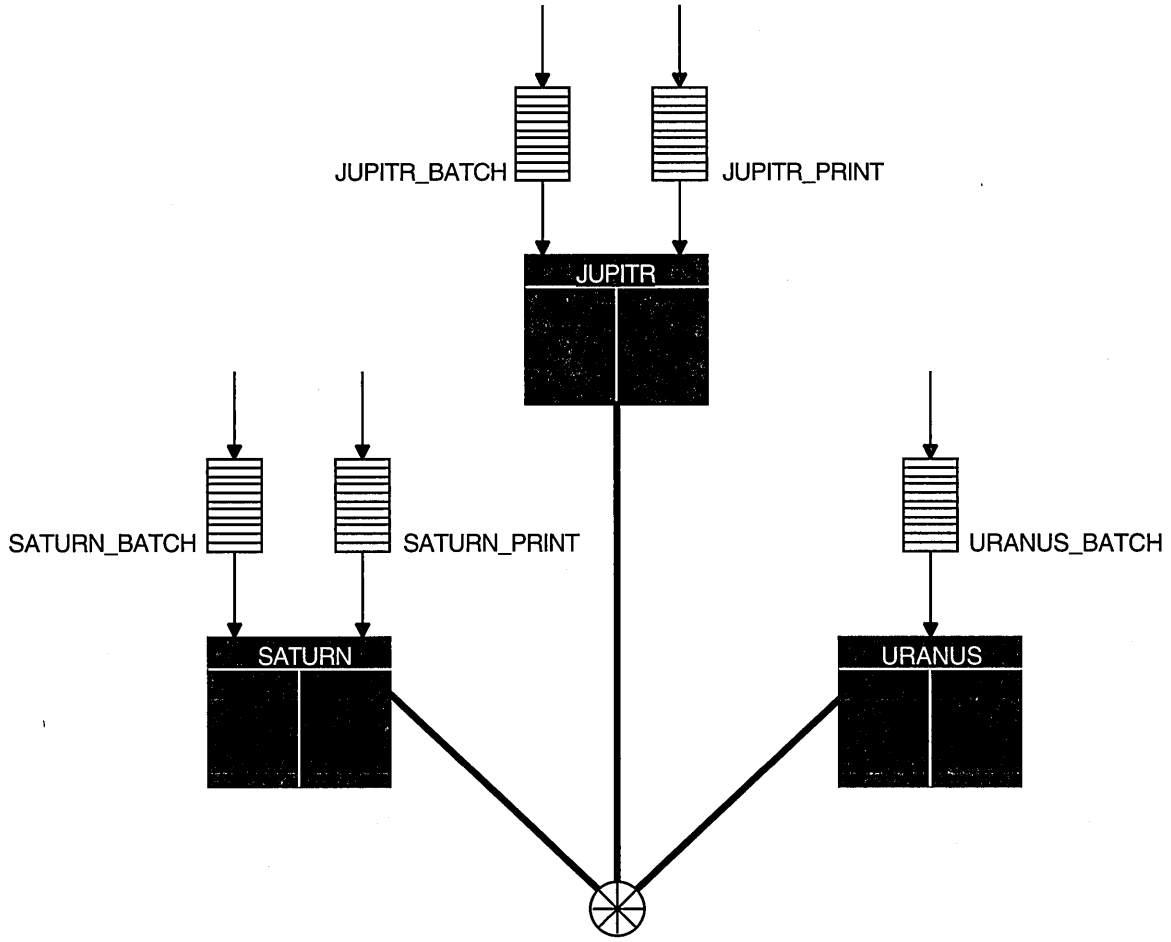
6.4 Cluster Batch Queues

Before you establish batch queues, you should decide which type of queue configuration best suits your cluster. As system manager, you are responsible for setting up batch queues to maintain efficient batch job processing on the cluster. For example, you should do the following:

- Determine what type of processing will be performed on each computer.
- Set up local batch queues that conform to these processing needs.
- Decide whether to set up any clusterwide generic queues that will distribute batch job processing across the cluster.
- Decide whether to use autostart queues for high availability.

Once you determine the strategy that best suits your needs, you can create a command procedure to set up your queues (see Example 6–2). Figure 6–4 shows a batch queue configuration for a cluster consisting of computers JUPITR, SATURN, and URANUS.

Figure 6–4 Sample Batch Queue Configuration



6.4.1 Setting Up Execution Batch Queues

Generally, you set up execution batch queues on each VMScluster computer using the same procedures you use for a standalone computer. For more detailed information about how to do this, see the *OpenVMS System Manager's Manual.*

You create a batch queue with a unique name by specifying the DCL command INITIALIZE/QUEUE/BATCH in the following format:

```
INITIALIZE/QUEUE/BATCH/ON=node::[/START]
[/NAME_OF_MANAGER=] queue-name
```

The /ON qualifier specifies the computer on which the batch queue runs. If you specify the /START qualifier, the queue is started. If you are running multiple queue managers, you should also specify the queue manager with the /NAME_OF_MANAGER qualifier.

You can create an autostart batch queue by specifying the DCL command INITIALIZE/QUEUE/BATCH in the following format:

```
INITIALIZE/QUEUE/BATCH/AUTOSTART_ON=node:::... queue-name
```

When you use 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 cannot begin processing jobs until the ENABLE AUTOSTART /QUEUES command is entered for a node on which the queue can run. Generic queues cannot be autostart queues. Note that you cannot specify both /ON and /AUTOSTART_ON.

The following commands make the local batch queue assignments for JUPITR, SATURN, and URANUS shown in Figure 6–4:

```
$ INITIALIZE/QUEUE/BATCH/ON=JUPITR::/START/NAME_OF_MANAGER=BATCH_QUEUE JUPITR_BATCH
$ INITIALIZE/QUEUE/BATCH/ON=SATURN::/START/NAME_OF_MANAGER=BATCH_QUEUE SATURN_BATCH
$ INITIALIZE/QUEUE/BATCH/ON=URANUS::/START/NAME_OF_MANAGER=BATCH_QUEUE URANUS_BATCH
```

Because batch jobs on each VMScluster computer are queued to SYS$BATCH by default, you should consider defining a logical name to establish this queue as a clusterwide generic batch queue that distributes batch job processing throughout the cluster (see Example 6–2). Note, however, that you should do this only if you have a common-environment cluster. Section 6.4.2 presents guidelines for establishing clusterwide generic batch queues.

6.4.2 Setting Up Clusterwide Generic Batch Queues

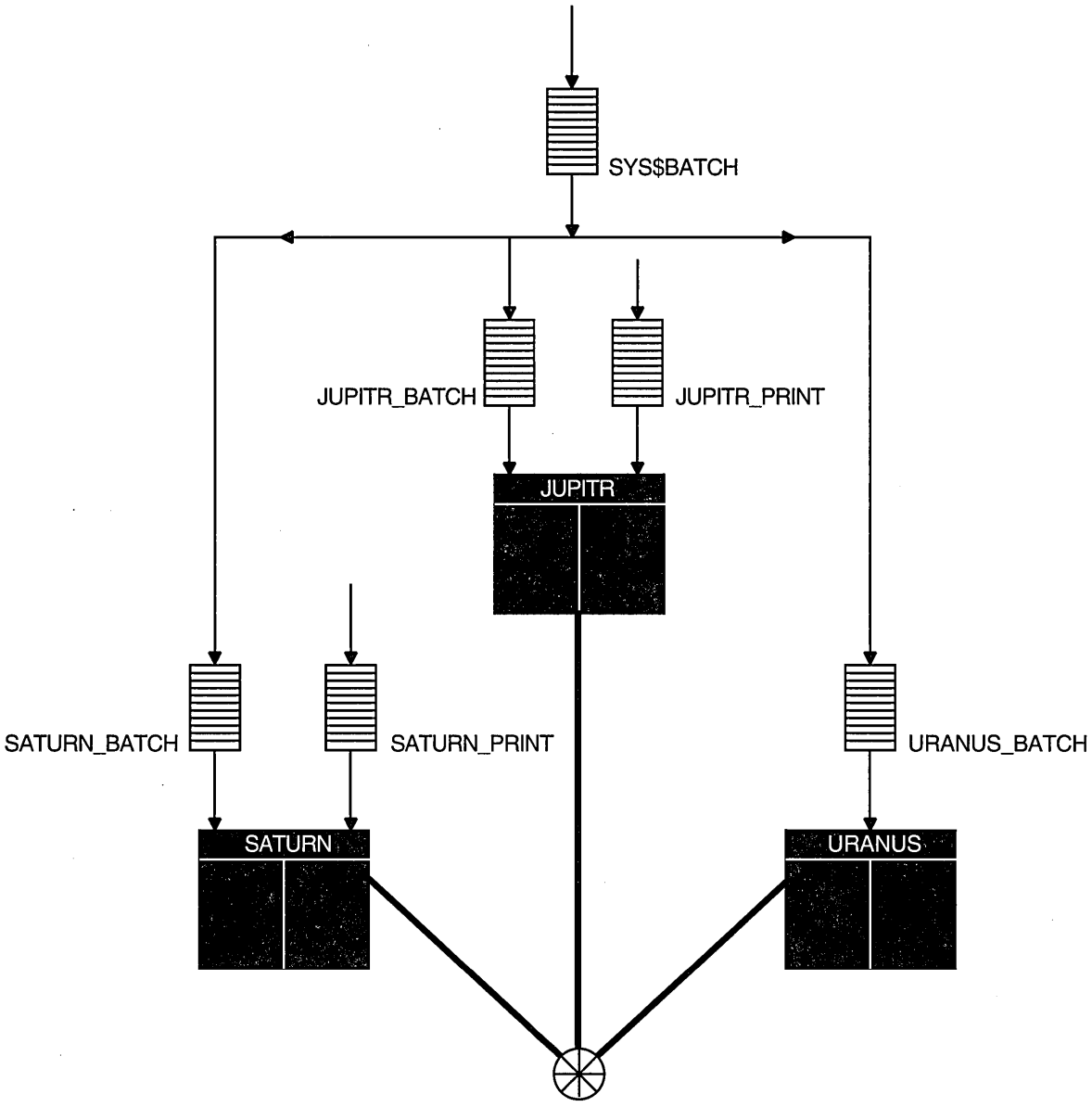
In a VMScluster system, you can distribute batch processing among computers to balance the use of processing resources. You can achieve this workload distribution by assigning local batch queues to one or more clusterwide generic batch queues. These generic batch queues control batch processing across the cluster by placing batch jobs in assigned batch queues that are available. You can create a clusterwide generic batch queue as shown in Example 6–2.

In Figure 6–5, batch queues from each VMScluster computer are assigned to a clusterwide generic batch queue named SYS$BATCH. Users can submit a job to a specific queue (for example, JUPITR_BATCH or SATURN_BATCH), or, if they have no special preference, they can submit it by default to the clusterwide generic queue SYS$BATCH. The generic queue in turn places the job in an available assigned queue in the cluster.

If more than one assigned queue is available, the operating system selects the queue that minimizes the ratio (executing jobs/job limit) for all assigned queues.

A clusterwide generic batch queue needs to be initialized and started only once. The most efficient way to perform these operations is to create a common command procedure that is executed by each VMScluster computer (see Example 6–2).

Figure 6–5 Clusterwide Generic Batch Queue Configuration



6.5 Using a Common Command Procedure to Set Up Cluster Queues

Once you have created queues, you must start them to begin processing batch and print jobs. In addition, you must make sure the queues are started each time the system reboots, by enabling autostart for autostart queues or by entering

START/QUEUE commands for nonautostart queues. To do so, create a command procedure containing the necessary commands.

You can create a common command procedure named, for example, QSTARTUP.COM, and store it on a shared disk. With this method, each node can share the same copy of the common QSTARTUP.COM procedure. Each node invokes the common QSTARTUP.COM procedure from the common version of SYSTARTUP. You can also include the commands to start queues in the common SYSTARTUP file instead of in a separate QSTARTUP.COM file.

Example 6–1 shows commands used to create VMScluster queues.

Example 6–1 Sample Commands for Creating VMScluster Queues

```
$
❶
$  DEFINE/FORM LN_FORM 10 /WIDTH=80 /STOCK=DEFAULT /TRUNCATE
$  DEFINE/CHARACTERISTIC 2ND_FLOOR 2
   .
   .
   .
❷
$  INITIALIZE/QUEUE/AUTOSTART_ON=(JUPITR::LPA0:)/START JUPITR_PRINT
$  INITIALIZE/QUEUE/AUTOSTART_ON=(SATURN::LPA0:)/START SATURN_PRINT
$  INITIALIZE/QUEUE/AUTOSTART_ON=(URANUS::LPA0:)/START URANUS_PRINT
   .
   .
   .
❸
$  INITIALIZE/QUEUE/BATCH/START/ON=JUPITR:: JUPITR_BATCH
$  INITIALIZE/QUEUE/BATCH/START/ON=SATURN:: SATURN_BATCH
$  INITIALIZE/QUEUE/BATCH/START/ON=URANUS:: URANUS_BATCH
   .
   .
   .
❹
$  INITIALIZE/QUEUE/START -
_$ /AUTOSTART_ON=(JUPITR::LTA1:,SATURN::LTA1,URANUS::LTA1) -
_$ /PROCESSOR=LATSYM /FORM_MOUNTED=LN_FORM -
_$ /RETAIN=ERROR /DEFAULT=(NOBURST,FLAG=ONE,NOTRAILER) -
_$ /RECORD_BLOCKING LN03$PRINT
$
$INITIALIZE/QUEUE/START -
_$ /AUTOSTART_ON=(JUPITR::LTA2:,SATURN::LTA2,URANUS::LTA2) -
_$ /PROCESSOR=LATSYM /RETAIN=ERROR -
_$ /DEFAULT=(NOBURST,FLAG=ONE,NOTRAILER) /RECORD_BLOCKING -
_$ /CHARACTERISTIC=2ND_FLOOR LA210$PRINT
$
❺
$  ENABLE AUTOSTART/QUEUES/ON=SATURN
$  ENABLE AUTOSTART/QUEUES/ON=JUPITR
$  ENABLE AUTOSTART/QUEUES/ON=URANUS
❻
$  INITIALIZE/QUEUE/START SYS$PRINT -
_$ /GENERIC=(JUPITR_PRINT,SATURN_PRINT,URANUS_PRINT)
$
❼
$  INITIALIZE/QUEUE/BATCH/START SYS$BATCH -
_$ /GENERIC=(JUPITR_BATCH,SATURN_BATCH,URANUS_BATCH)
$
```

The following are descriptions of each command or group of commands in Example 6–1:

❶ Define all printer forms and characteristics.

❷ Initialize local printer queues. In the example, these queues are autostart queues and are started automatically when the node executes the ENABLE AUTOSTART/QUEUES command. Although the /START qualifier is specified to activate the autostart queues, they do not begin processing jobs until autostart is enabled.

To enable autostart each time the system reboots, add the ENABLE AUTOSTART/QUEUES command to your queue startup command procedure, as shown in Example 6–2.

❸ Initialize and start local batch queues on all nodes, including satellite nodes. In this example, the local batch queues are not autostart queues. For more information about local batch queues, see Section 6.6.

❹ Initialize queues for remote LAT printers. In the example, these queues are autostart queues and are set up to run on one of three nodes. The queues are started on the first of those three nodes to execute the ENABLE AUTOSTART command.

You must establish the logical devices LTA1 and LTA2 in the LAT startup command procedure LAT$SYSTARTUP.COM on each node on which the autostart queue can run. For more information, see the description of editing LAT$SYSTARTUP.COM in the *OpenVMS System Manager's Manual*.

Although the /START qualifier is specified to activate these autostart queues, they will not begin processing jobs until autostart is enabled.

❺ Enable autostart to start the autostart queues automatically. In the example, autostart is enabled on node SATURN first, so SATURN starts the autostart queues that are set up to run one of several nodes.

❻ Initialize and start the generic output queue SYS$PRINT. This is a nonautostart queue (generic queues cannot be autostart queues). However, generic queues are not stopped automatically when a system is shut down, so you do not need to restart the queue each time a node reboots.

❼ Initialize and start the generic batch queue SYS$BATCH. Because this is a generic queue, it is not stopped when the node shuts down. Therefore, you do not need to restart the queue each time a node reboots.

Example 6–2 illustrates the use of a common QSTARTUP command procedure on a shared disk.

Example 6–2 Common Procedure to Start VMScluster Queues

```
$!
$! QSTARTUP.COM -- Common procedure to set up cluster queues
$!
$!
❶
$ NODE = F$GETSYI("NODENAME")
```

(continued on next page)

Example 6–2 (Cont.) Common Procedure to Start VMScluster Queues

```
$!
$! Determine the node-specific subroutine
$!
$ IF (NODE .NES. "JUPITR") .AND. (NODE .NES. "SATURN") .AND. (NODE .NES. "URANUS")
$    THEN
$       GOSUB SATELLITE_STARTUP
$    ELSE
❷
$!
$!      Configure remote LAT devices.
$!
$       SET TERMINAL LTA1: /PERM /DEVICE=LN03 /WIDTH=255 /PAGE=60 -
            /LOWERCASE /NOBROAD
$       SET TERMINAL LTA2: /PERM /DEVICE=LA210 /WIDTH=255 /PAGE=66 -
                /NOBROAD
$       SET DEVICE LTA1: /SPOOLED=(LN03$PRINT,SYS$SYSDEVICE:)
$       SET DEVICE LTA2: /SPOOLED=(LA210$PRINT,SYS$SYSDEVICE:)
❸
$       START/QUEUE/BATCH 'NODE'_BATCH
$       GOSUB 'NODE'_STARTUP
$    ENDIF
$ GOTO ENDING
$!
$! Node-specific subroutines start here
$!
❹
$ SATELLITE_STARTUP:
$!
$! Start a batch queue for satellites.
$!
$ START/QUEUE/BATCH 'NODE'_BATCH
$ RETURN
$!
❺
$JUPITR_STARTUP:
$!
$! Node-specific startup for JUPITR::
$! Setup local devices and start nonautostart queues here
$!
$ SET PRINTER/PAGE=66 LPA0:
$ RETURN
```

(continued on next page)

Example 6–2 (Cont.) Common Procedure to Start VMScluster Queues

```
$!
$SATURN_STARTUP:
$!
$! Node-specific startup for SATURN::
$! Setup local devices and start nonautostart queues here
$!
   .
   .
   .
$ RETURN
$!
$URANUS_STARTUP:
$!
$! Node-specific startup for URANUS::
$! Setup local devices and start nonautostart queues here
$!
   .
   .
   .
$ RETURN
$!
$ENDING:
❻
$! Enable autostart to start all autostart queues
$!
$ ENABLE AUTOSTART/QUEUES
$ EXIT
```

Following are descriptions of each phase of the common QSTARTUP.COM command procedure in Example 6–2:

❶ Determine the name of the node executing the procedure.

❷ On all large nodes, set up remote devices connected via LAT. The queues for these devices are autostart queues and are started automatically when the ENABLE AUTOSTART/QUEUES command is executed at the end of this procedure.

In the example, these autostart queues were set up to run on one of three nodes. The queues start when the first of those nodes executes the ENABLE AUTOSTART/QUEUES command. The queue remains running as long as one of those nodes is running and has autostart enabled.

❸ On large nodes, start the local batch queue. In the example, the local batch queues are nonautostart queues and must be started explicitly with START /QUEUE commands.

❹ On satellite nodes, start the local batch queue.

❺ Each node executes its own subroutine. On node JUPITR, set up the line printer device LPA0:. The queue for this device is an autostart queue and is started automatically when the ENABLE AUTOSTART/QUEUES command is executed.

❻ Enable autostart to start all autostart queues.

6.6 Starting Local Batch Queues

Normally, you use local batch execution queues during startup to run batch jobs to start layered products. For this reason, these queues must be available before printers are configured and before the ENABLE AUTOSTART command is executed in QSTARTUP.COM.

Start the local batch execution queue in each node's startup command procedure SYSTARTUP_VMS.COM. If you use a common startup command procedure, add commands similar to the following to your procedure:

```
$ SUBMIT/PRIORITY=255/NOIDENT/NOLOG/QUEUE='NODE'_BATCH LAYERED_PRODUCT.COM
$ START/QUEUE 'NODE'_BATCH
$ DEFINE/SYSTEM/EXECUTIVE SYS$BATCH 'NODE'_BATCH
```

Submitting the startup command procedure LAYERED_PRODUCT.COM as a high-priority batch job before the queue starts ensures that the job is executed immediately, regardless of the job limit on the queue. If the queue were started before the command procedure was submitted, the queue might reach its job limit by scheduling user batch jobs, and the startup job would have to wait.

6.7 Disabling Autostart During Shutdown

By default the shutdown procedure disables autostart at the beginning of the shutdown sequence. Autostart is disabled to allow autostart queues with failover lists to fail over to another node. Autostart also prevents any autostart queue running on another node in the cluster to fail over to the node being shut down.

You can change the time at which autostart is disabled in the shutdown sequence in one of two ways:

- Define the logical name SHUTDOWN$DISABLE_AUTOSTART as follows:

  ```
  $ DEFINE/SYSTEM/EXECUTIVE SHUTDOWN$DISABLE_AUTOSTART number-of-minutes
  ```

  Set the value of *n* to the number of minutes before shutdown when autostart is to be disabled. You can add this logical name definition to SYLOGICALS.COM. The value of *n* is the default value for the node. If this number is greater than the number of minutes specified for the entire shutdown sequence, autostart is disabled at the beginning of the sequence.

- Specify the DISABLE_AUTOSTART *number-of-minutes* option during the shutdown procedure. (The value you specify for *number-of-minutes* overrides the value specified for the SHUTDOWN$DISABLE_AUTOSTART logical name.)

See the *OpenVMS System Manager's Manual* for more information about changing the time at which autostart is disabled during the shutdown sequence.

7

Building and Maintaining the Cluster

Before you attempt to build your cluster, be sure you have read the previous chapters and made the following preparations:

- Determined the VMScluster configuration type (CI, DSSI, local area, or mixed interconnect)
- Determined whether you want a common-environment or multiple-environment VMScluster system
- Determined how you will set up and distribute the startup and system files that define the operating environment
- Planned your disk, tape, and queue configurations
- Installed or upgraded the OpenVMS AXP operating system on the first AXP computer in the VMScluster system and installed any required licenses
- Installed or upgraded the OpenVMS VAX operating system on the first VAX computer in the VMScluster system and installed any required licenses
- Configured and started the DECnet for OpenVMS (DECnet) network

Once you have made these preparations, you can use the information in this chapter to build and maintain your cluster. Topics include the following:

- CLUSTER_CONFIG.COM functions
- Determining locations and sizes for satellite page and swap files
- Specifying allocation classes in mixed-interconnect clusters
- Configuring the cluster
- Reconfiguring the cluster after a major change
- Maintaining the cluster

7.1 CLUSTER_CONFIG.COM Functions

When you invoke the CLUSTER_CONFIG.COM command procedure, it displays a menu of configuration options. By selecting the appropriate option, you can configure the cluster easily and reliably without invoking any OpenVMS utilities directly. Use CLUSTER_CONFIG.COM to perform these functions:

- Add a computer to the cluster.
- Remove a computer from the cluster.
- Change a computer's characteristics.
- Create a duplicate system disk.

Table 7–1 summarizes the operations that CLUSTER_CONFIG.COM performs for each configuration function.

Table 7–1 Summary of CLUSTER_CONFIG.COM Functions

Function	Operations Performed
ADD	Establish the new computer's root directory on a cluster common system disk and generate the computer's system parameter files (ALPHAVMSSYS.PAR for AXP systems or VAXVMSSYS.PAR for VAX systems), and MODPARAMS.DAT in its SYS$SPECIFIC:[SYSEXE] directory.
	Update the permanent and volatile remote node network databases for the computer on which CLUSTER_CONFIG.COM is executed to add the new computer. If the new computer is a satellite, update SYS$MANAGER:NETNODE_UPDATE.COM on the local computer (see Section 7.5.1.2).
	Generate the new computer's page and swap files (PAGEFILE.SYS and SWAPFILE.SYS).
	Set up a cluster quorum disk (optional).
	Set allocation class (ALLOCLASS or TAPE_ALLOCLASS) value for the new computer, if the computer is being added as a disk or tape server.
	Generate an initial (temporary) startup procedure for the new computer. This initial procedure runs NETCONFIG.COM to configure the network, runs AUTOGEN to set appropriate system parameter values for the computer, and reboots the computer with normal startup procedures.
REMOVE	Delete another computer's root directory and its contents from the local computer's system disk. If the computer being removed is a satellite, update SYS$MANAGER:NETNODE_UPDATE.COM on the local computer.
	Update the permanent and volatile remote node network databases on the local computer.
CHANGE	Enable or disable the local computer as a disk server; enable or disable the local computer as a boot server; enable or disable the Ethernet or FDDI LAN for cluster communications on the local computer; enable or disable a quorum disk on the local computer; change the local computer's ALLOCLASS or TAPE_ALLOCLASS value; change a satellite's LAN hardware address. Procedure displays CHANGE menu and prompts for appropriate information.
CREATE	Duplicate the local computer's system disk and remove all system roots from the new disk.

If you intend to set up a local area or mixed-interconnect cluster, you must do the following before executing CLUSTER_CONFIG.COM:

- Determine locations and sizes for satellite page and swap files.
- Select cluster boot servers and disk servers.
- Determine allocation classes for computers and disks (also applicable for CI configurations).

Guidelines are provided in Section 7.2, Section 7.3, and Section 7.4, respectively.

Note that some configuration functions, such as adding or removing a **voting member** (a computer with a nonzero value for the system parameter VOTES) or enabling or disabling a quorum disk, require one or more additional operations. Refer to Section 7.6 for instructions.

When you remove a computer from or add a computer to a cluster that uses DECdtm services, you must do some extra tasks in order to ensure the integrity of your data. These tasks are described in the chapter on DECdtm services in the *OpenVMS System Manager's Manual.*

If you are not sure whether your cluster uses DECdtm services, enter this command sequence:

```
$ SET PROCESS /PRIVILEGES=SYSPRV
$ RUN SYS$SYSTEM:LMCP
LMCP> SHOW LOG
```

If your cluster uses DECdtm services, the SHOW LOG command will display a list of the files that DECdtm uses to store information about transactions. If your cluster does not use DECdtm services, it will display a "file not found" error message.

7.2 Determining Locations and Sizes for Satellite Page and Swap Files

When you add a computer to the cluster, CLUSTER_CONFIG.COM prompts for the sizes and location of the computer's page (PAGEFILE.SYS) and swap (SWAPFILE.SYS) files. (The default sizes supplied by the procedure are minimums.) Depending on the configuration of your VMScluster system disk and your network, you may realize a performance improvement in local area and mixed-interconnect configurations by locating page and swap files for satellites on a satellite's local disk, if such a disk is available.

To set up page and swap files on a satellite's local disk, CLUSTER_CONFIG.COM creates (in the satellite's [SYS*n*.SYSEXE] directory on the boot server's system disk) the command procedure SATELLITE_PAGE.COM. This procedure executes when AUTOGEN reboots the satellite at the end of CLUSTER_CONFIG.COM. SATELLITE_PAGE.COM performs the following functions:

- Mounts the satellite's local disk with a volume label that is unique in the cluster in the format *node-name*_SCSSYSTEMID
- Installs the page and swap files on the satellite's local disk

If you want to alter the volume label, follow these steps after the satellite has been added to the cluster:

1. Log in as system manager and enter a DCL command in the following format:

   SET VOLUME/LABEL=volume-label device-spec[:]

   Note that the SET VOLUME command requires write access (W) to the index file on the volume. If you are not the volume's owner, you must have either a system user identification code (UIC) or the SYSPRV privilege.

2. Update SATELLITE_PAGE.COM to reflect the new label.

To relocate the satellite's page and swap files (for example, from the satellite's local disk to the boot server's system disk, or the reverse) or to change file sizes, the easiest way is to remove the satellite from the cluster and then add it again using CLUSTER_CONFIG.COM.

7.3 Selecting MOP and Disk Servers

While every local area and mixed-interconnect cluster must include at least one Maintenance Operations Protocol (MOP) and disk server, multiple MOP and disk servers offer the following advantages:

- Higher availability—Satellites can access served disks and boot even if one of the servers is temporarily unavailable.
- Better workload balancing—The task of serving disks to satellites can place a significant load on a server. With multiple servers, this work load is distributed across more computers and LAN adapters.

As a general rule, choose the most powerful computers in the cluster to be used as MOP and disk servers. Low-powered computers can become overloaded when serving many busy satellites or when many satellites boot simultaneously. Note, however, that two or more moderately powered servers can provide better performance than a single high-powered server. Multiple servers give better availability, and they distribute the work load across more LAN adapters. If you have several computers of roughly comparable power, it is reasonable to use them all as boot servers. This arrangement gives optimal load balancing. In addition, if one computer fails or is shut down, others remain available to serve satellites.

After compute power, the most important factor in selecting a server is the speed of its LAN adapter. Servers should be equipped with the highest bandwith LAN adapters in the cluster.

The following list describes what happens during satellite booting:

- The satellite requests MOP service

  This is the original boot request that a satellite sends out across the network. Any node in the VMScluster that has the MOP service enabled and has the LAN address of the particular satellite node in the server's DECnet database can become the MOP server for the satellite.

- The MOP server loads the AXP or VAX system

  The NISCS load-assist agent program, NISCS_LAA, executes on the MOP server and passes certain system parameter values to the NISCA boot driver. NISCS_LAA is a shareable image subroutine that executes from the MOP server and uses system parameters and disk parameters for the satellite root that is being downline loaded. NISCS passes these parameters to the NISCS load image because their values are necessary to establish a NISCA connection. Some of these parameters include:

  - The system disk descriptor
  - The root of the satellite
  - VMScluster system parameters, such as SYSSTEMID, SCSNODE, and NISCS_CONV_BOOT
  - The cluster group code and password

  AXP

  On AXP systems, the MOP server responds to an AXP satellite boot request by downline loading the SYS$SYSTEM:APB.EXE program along with the required parameters.♦

VAX

On VAX systems, the MOP server responds to a VAX satellite boot request by downline loading the SYS$LIBRARY:NISCS_LOAD.EXE program along with the required parameters.♦

The satellite executes the load program, which establishes an SCS connection to a disk server for the satellite system disk and loads the SYSBOOT.EXE program.

7.4 Determining Allocation Class Values

The allocation class value coupled with unit numbers and, for local controllers, with the controller letter, must form a clusterwide unique name for the device.

Before setting up any cluster, you must determine allocation class values for disk servers, tape servers, HSC subsystems, and DSSI ISEs. It is easiest to use the same value for all servers, HSC subsystems, and DSSI ISEs; you can arbitrarily choose a number between 1 and 255. Note, however, that to change the allocation class value on any CI or DSSI connected computer, you must shut down and reboot the entire cluster (see Section 7.6).

As explained in Section 5.2, every device allocation-class name (in the form 1*ddcu*) must be the same for all servers and HSC subsystems that share the devices. For RA series disks, make sure that all the removable unit plugs on all disks of that allocation class are unique. As long as you have no more than 256 such disks, this is easy to accomplish.

Assume, for instance, that 10 disks are dual pathed between the HSC subsystems VOYGR1 and VOYGR2, and assume that 10 others are dual pathed between the HSC subsystems VIKNG1 and VIKNG2. Provided that all 20 disks have unique unit numbers, you can assign the same allocation class value to all four HSC subsystems.

However, if you run out of unique disk unit numbers, you must define unique disk names by using two or more allocation class values for the HSC subsystems. You must also configure one or more computers to serve HSC disks and assign allocation class values accordingly. To perform those operations, you can execute the CLUSTER_CONFIG.COM CHANGE function, which is described in Section 7.5.3.

Additionally, you must ensure that all locally connected disks have unique device names. For example, if SATURN and URANUS each have a single-pathed RA81 disk connected to a local BDA controller with unit plug 0, and if both computers have an allocation class value of 1, then both RA81 disks receive the same device name (1DUA0). Because both disks have the same device name, they appear to the operating system software as the same disk. This condition can endanger data integrity. You can avoid potential problems by selecting a different unit number for one of the disks or by using a different allocation class on one of these nodes.

Note that, because fewer unit numbers are available for MASSBUS or UNIBUS disks, fewer unique device names can be defined. To ensure that device names remain unique in your cluster, you may have to relocate such disks or disqualify a computer as a disk server.

7.5 Configuring the Cluster

To perform configuration functions, execute CLUSTER_CONFIG.COM. Before invoking the procedure, be sure to verify the following:

- You are logged in to the system manager's account on an appropriate computer. If you are building a new local area or mixed-interconnect cluster, you must be logged in to a computer that you want to set up as a boot server. If you are adding a satellite, you must be logged in to a boot server. Note that the process privileges SYSPRV, OPER, CMKRNL, BYPASS, and NETMBX are required, because the procedure performs sensitive system operations.
- The DECnet network is up and running and all computers are connected to the LAN.
- You have at hand the data listed in Table 7–2. Note that some items are configuration specific.
- If your configuration has two or more system disks, you have coordinated cluster common files, as described in Section 4.5.4.
- If you are removing a computer from a cluster that uses DECdtm services, make sure that you have followed the step-by-step instructions in the chapter on DECdtm services in the *OpenVMS System Manager's Manual*. These instructions describe how to remove a computer safely from the cluster, thereby preserving the integrity of your data.

  If you are not sure whether your cluster uses DECdtm services, see Section 7.1.

Sections 7.5.1 through 7.5.6 provide examples of typical interactive CLUSTER_CONFIG.COM sessions. Section 7.6 describes tasks you must perform after executing CLUSTER_CONFIG.COM to make major configuration changes. Although CLUSTER_CONFIG.COM functions the same for both AXP and VAX systems, the command procedure questions and format may appear slightly different according to the type of system.

Caution

You cannot initiate concurrent CLUSTER_CONFIG.COM sessions.

Table 7–2 Data Requested by CLUSTER_CONFIG.COM

Information Required	How to Specify or Obtain
Device name of cluster system disk on which root directories will be created.	System manager specifies. If there is no logical name defined for the SYS$SYSDEVICE device, the default becomes the translation of the SYS$SYSDEVICE: logical name.
Computer's root directory name on cluster system disk.	System manager specifies. Name must be in the form SYS*x*. For computers connected by CI, *x* is a hexadecimal digit in the range 1 through 9 or A through D (for example, SYS1 or SYSA). For satellites, *x* must be in the range from 10 through FFFF. Procedure supplies valid default.

(continued on next page)

Table 7–2 (Cont.) Data Requested by CLUSTER_CONFIG.COM

Information Required	How to Specify or Obtain
Computer's DECnet node name.	Network manager supplies. Name must be from 1 to 6 alphanumeric characters and *cannot* include dollar signs ($) or underscores (_).
Computer's DECnet node address.	Network manager supplies.
Cluster group number and password if the CHANGE function of CLUSTER_CONFIG.COM is run to enable cluster communications over the LAN.	Network manager specifies.

If computer is a satellite on a LAN, satellite's LAN hardware address. Address has the form *xx-xx-xx-xx-xx-xx*. Note that you must include the hyphens when you specify a hardware address.	When DECnet network is running on boot server, proceed as follows:
	• On AXP systems, enter the following command at the satellite's console:
	`>>> SHOW NETWORK`
	Note that you can also use the SHOW CONFIG command.
	• On MicroVAX II and VAXstation II satellite nodes, enter the following commands at the satellite's console:
	`>>> B/100 XQA0`
	`Bootfile: READ_ADDR`
	• On MicroVAX 2000 and VAXstation 2000 satellite nodes, enter the following commands at successive console mode prompts:
	`>>> T 53`
	`2 ?>>> 3`
	`>>> B/100 ESA0`
	`Bootfile: READ_ADDR`
	If the second prompt appears as 3 ?>>>, press the Return key.
	• On MicroVAX 3*xxx* and 4*xxx* series satellite nodes, enter the following command at the satellite's console:
	`>>> SHOW ETHERNET`

Workstation windowing system.	System manager specifies. Workstation software must be installed before workstation satellites are added. If it is not, the procedure indicates that fact.
Location and sizes of page and swap files.	System manager specifies.
Value for local computer's allocation class (ALLOCLASS or TAPE_ALLOCLASS) parameter.	System manager specifies.
Physical device name of quorum disk.	System manager specifies.

7.5.1 Adding a Computer to the Cluster

Once you have made the necessary preparations, you can execute CLUSTER_CONFIG.COM to add a new computer to the cluster.

- If you are setting up a CI cluster or a DSSI cluster, invoke CLUSTER_CONFIG.COM on an active VMScluster computer and select the ADD function.
- If you are setting up a new local area or mixed-interconnect cluster, follow these steps:
  1. Invoke CLUSTER_CONFIG.COM and execute the CHANGE function described in Section 7.5.3 to enable the local computer as a boot server.
  2. After the CHANGE function completes, execute the ADD function to add either CI connected computers or satellites to the cluster. To add satellites, you must be logged in on a cluster boot server.

While adding computers, you may want to disable broadcast messages to your terminal—the ADD function generates many such messages. To disable the messages, you can enter the DCL command REPLY/DISABLE=(NETWORK, CLUSTER).

Whenever you add a voting member to the cluster, you must, after the ADD function completes, reconfigure the cluster, following instructions in Section 7.6. In addition, if you add a CI connected computer that boots from a cluster common system disk, you must create a new default bootstrap command procedure for the computer before booting it into the cluster. For instructions, refer to your computer-specific installation and operations guide.

If your cluster uses DECdtm services, you must create a transaction log for the computer when you have configured it into your cluster. For step-by-step instructions on how to do this, see the chapter on DECdtm services in the *OpenVMS System Manager's Manual*.

If you are not sure whether your cluster uses DECdtm services, see Section 7.1.

Example 7–1 and Example 7–2 illustrate the use of CLUSTER_CONFIG.COM on JUPITR to add, respectively, CI connected computer SATURN and satellite computer EUROPA to the cluster.

Caution

If either the local or the new computer fails before the ADD function completes, you must, after normal conditions are restored, perform the REMOVE function to erase any invalid data and then restart the ADD function.

Example 7–1 Sample Interactive CLUSTER_CONFIG.COM Session to Add a CI Connected Computer as a Boot Server

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for JUPITR.

    Enter choice [1]: Return

    The ADD function adds a new node to the cluster.

    If the node being added is a voting member, EXPECTED_VOTES in all
    other cluster members' MODPARAMS.DAT must be adjusted, and the
    cluster must be rebooted.

    If the new node is a satellite, the network databases on JUPITR are
    updated. The network databases on all other cluster members must be
    updated.

    For instructions, see the VMScluster Systems or OpenVMS manual.

What is the node's DECnet node name? SATURN
What is the node's DECnet address? 2.3
Will SATURN be a satellite [Y]? N
Will SATURN be a boot server [Y]? Return

    This procedure will now ask you for the device name of SATURN's system root.
    The default device name (DISK$VAXVMSRL5:) is the logical volume name of
    SYS$SYSDEVICE:.

What is the device name for SATURN's system root [DISK$VAXVMSRL5:]? Return
What is the name of the new system root [SYSA]? Return
Creating directory tree SYSA...
%CREATE-I-CREATED, $1$DJA11:<SYSA> created
%CREATE-I-CREATED, $1$DJA11:<SYSA.SYSEXE> created
   .
   .
   .
System root SYSA created.
Enter a value for SATURN's ALLOCLASS parameter: 1
Does this cluster contain a quorum disk [N]? Y
What is the device name of the quorum disk? $1$DJA12
Updating network database...
Size of page file for SATURN [10000 blocks]? 50000
Size of swap file for SATURN [8000 blocks]? 20000
Will a local (non-HSC) disk on SATURN be used for paging and swapping? N

     If you specify a device other than DISK$VAXVMSRL5: for SATURN's
     page and swap files, this procedure will create PAGEFILE_SATURN.SYS
     and SWAPFILE_SATURN.SYS in the <SYSEXE> directory on the device you
     specify.
What is the device name for the page and swap files [DISK$VAXVMSRL5:]? Return
%SYSGEN-I-CREATED, $1$DJA11:<SYSA.SYSEXE>PAGEFILE.SYS;1 created
%SYSGEN-I-CREATED, $1$DJA11:<SYSA.SYSEXE>SWAPFILE.SYS;1 created
```

(continued on next page)

Example 7–1 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Add a CI Connected Computer as a Boot Server

```
The configuration procedure has completed successfully.
     SATURN has been configured to join the cluster.

     Before booting SATURN, you must create a new default bootstrap
     command procedure for SATURN. See your processor-specific
     installation and operations guide for instructions.

     The first time SATURN boots, NETCONFIG.COM and
     AUTOGEN.COM will run automatically.

     The following parameters have been set for SATURN:

                  VOTES = 1
                  QDSKVOTES = 1

     After SATURN has booted into the cluster, you must increment
     the value for EXPECTED_VOTES in every cluster member's
     MODPARAMS.DAT. You must then reconfigure the cluster, using the
     procedure described in the VMScluster Systems for OpenVMS manual.
```

Example 7–2 Sample Interactive CLUSTER_CONFIG.COM Session to Add a VAX Satellite with Local Page and Swap Files

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

           1. ADD a node to the cluster.
           2. REMOVE a node from the cluster.
           3. CHANGE a cluster node's characteristics.
           4. CREATE a second system disk for JUPITR.

    Enter choice [1]: Return

    The ADD function adds a new node to the cluster.

    If the node being added is a voting member, EXPECTED_VOTES in all
    other cluster members' MODPARAMS.DAT must be adjusted, and the
    cluster must be rebooted.

    If the new node is a satellite, the network databases on JUPITR are
    updated. The network databases on all other cluster members must be
    updated.

    For instructions, see the VMScluster Systems for OpenVMS manual.

What is the node's DECnet node name? EUROPA
What is the node's DECnet address? 2.21
Will EUROPA be a satellite [Y]? Return
Verifying circuits in network database...

    This procedure will now ask you for the device name of EUROPA's system root.
    The default device name (DISK$VAXVMSRL5:) is the logical volume name of
    SYS$SYSDEVICE:.
```

(continued on next page)

Example 7–2 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Add a VAX Satellite with Local Page and Swap Files

```
What is the device name for EUROPA'S system root [DISK$VAXVMSRL5:]? Return
What is the name of the new system root [SYS10]? Return
Allow conversational bootstraps on EUROPA [NO]? Return
The following workstation windowing options are available:

     1. No workstation software
     2. VWS Workstation Software
     3. DECwindows Workstation Software

Enter choice [1]: 3

Creating directory tree SYS10...
%CREATE-I-CREATED, $1$DJA11:<SYS10> created
%CREATE-I-CREATED, $1$DJA11:<SYS10.SYSEXE> created
   .
   .
   .
System root SYS10 created.
Will EUROPA be a disk server [N]? Return
What is EUROPA's Ethernet hardware address? 08-00-2B-03-51-75
Updating network database...
Size of pagefile for EUROPA [10000 blocks]? 20000
Size of swap file for EUROPA [8000 blocks]? 12000
Will a local disk on EUROPA be used for paging and swapping? YES
Creating temporary page file in order to boot EUROPA for the first time...
%SYSGEN-I-CREATED, $1$DJA11:<SYS10.SYSEXE>PAGEFILE.SYS;1 created

    This procedure will now wait until EUROPA joins the cluster.

    Once EUROPA joins the cluster, this procedure will ask you
    to specify a local disk on EUROPA for paging and swapping.

    Please boot EUROPA now.

Waiting for EUROPA to boot...
   .
   .
   .
(User enters boot command at satellite's console-mode prompt (>>>).
For MicroVAX II, VAXstation II, and MicroVAX 3xxx series satellites,
user enters B XQ.
For MicroVAX 2000 and VAXstation 2000 satellites, user enters B ES.)
   .
   .
   .
The local disks on EUROPA are:

Device            Device      Error     Volume    Free    Trans  Mnt
Name              Status      Count     Label     Blocks  Count  Cnt
EUROPA$DUA0:      Online          0
EUROPA$DUA1:      Online          0

Which disk can be used for paging and swapping? EUROPA$ DUA0:
May this procedure INITIALIZE EUROPA$DUA0: [YES]? NO
```

(continued on next page)

Example 7–2 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Add a VAX Satellite with Local Page and Swap Files

```
Mounting EUROPA$DUA0:...
PAGEFILE.SYS already exists on EUROPA$DUA0:

****************************************

Directory EUROPA$DUA0:[SYS0.SYSEXE]

PAGEFILE.SYS;1      23600/23600

Total of 1 file, 23600/23600 blocks.

****************************************

What is the file specification for the page file on
EUROPA$DUA0: [ <SYS0.SYSEXE>PAGEFILE.SYS ]? Return
%CREATE-I-EXISTS, EUROPA$DUA0:<SYS0.SYSEXE> already exists
This procedure will use the existing pagefile,
EUROPA$DUA0:<SYS0.SYSEXE>PAGEFILE.SYS;.

SWAPFILE.SYS already exists on EUROPA$DUA0:

****************************************

Directory EUROPA$DUA0:[SYS0.SYSEXE]

SWAPFILE.SYS;1      12000/12000

Total of 1 file, 12000/12000 blocks.

****************************************

What is the file specification for the swap file on
EUROPA$DUA0: [ <SYS0.SYSEXE>SWAPFILE.SYS ]? Return
This procedure will use the existing swapfile,
EUROPA$DUA0:<SYS0.SYSEXE>SWAPFILE.SYS;.

     AUTOGEN will now reconfigure and reboot EUROPA automatically.
     These operations will complete in a few minutes, and a
     completion message will be displayed at your terminal.

The configuration procedure has completed successfully.
```

7.5.1.1 Updating Network Data After Adding a Satellite

Whenever you add a satellite, CLUSTER_CONFIG.COM updates both the permanent and volatile remote node network databases on the boot server. However, the volatile databases on other cluster members are not automatically updated. To share the new data throughout the cluster, you must update the volatile databases on all other cluster members. Log in as system manager, invoke the SYSMAN utility, and enter the following commands at the SYSMAN> prompt:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username LAZARUS       will be used on nonlocal nodes
SYSMAN> SET PROFILE/PRIVILEGES=(OPER,SYSPRV)
SYSMAN> DO MCR NCP SET KNOWN NODES ALL
%SYSMAN-I-OUTPUT, command execution on node X...
   .
   .
   .
SYSMAN> EXIT
$
```

Note that the file NETNODE_REMOTE.DAT must be located in the directory SYS$COMMON:[SYSEXE]. However, remember that the file NETNODE_REMOTE.DAT cannot be shared between VAX and AXP processors. See Section 4.3 for more information.

7.5.1.2 Restoring a Satellite's Network Data

The first time you execute CLUSTER_CONFIG.COM to add a satellite, the procedure creates the file NETNODE_UPDATE.COM in the boot server's SYS$SPECIFIC:[SYSMGR] directory. (For a common-environment cluster, you must rename this file to the SYS$COMMON:[SYSMGR] directory, as described in Section 4.5.4.) This file, which is updated each time you add or remove a satellite or change its Ethernet or FDDI hardware address, contains all essential network configuration data for the satellite. If an unexpected condition at your site causes configuration data to be lost, you can use NETNODE_UPDATE.COM to restore it. You can also read the file when you need to obtain data about individual satellites. Note that you may want to edit the file occasionally to remove obsolete entries.

Example 7–3 shows the contents of the file after satellites EUROPA and GANYMD have been added to the cluster.

Example 7–3 Sample NETNODE_UPDATE.COM File

```
$ run sys$system:ncp
    define node EUROPA address 2.21
    define node EUROPA hardware address 08-00-2B-03-51-75
    define node EUROPA load assist agent sys$share:niscs_laa.exe
    define node EUROPA load assist parameter $1$DJA11:<SYS10.>
    define node EUROPA tertiary loader sys$system:tertiary_vmb.exe
    define node GANYMD address 2.22
    define node GANYMD hardware address 08-00-2B-03-58-14
    define node GANYMD load assist agent sys$share:niscs_laa.exe
    define node GANYMD load assist parameter $1$DJA11:<SYS11.>
    define node GANYMD tertiary loader sys$system:tertiary_vmb.exe
```

7.5.1.3 Controlling Clusterwide Broadcast Messages on Satellites and Boot Servers

When a satellite joins the cluster, the operator communication manager (OPCOM) has the following default states:

- For all systems in a VMScluster configuration except workstations:
  - OPA0: is enabled for all message classes.
  - The log file SYS$MANAGER:OPERATOR.LOG is opened for all classes.

- For workstations in a VMScluster configuration, even though the OPCOM process is running:
  - OPA0: is not enabled.
  - No log file is opened.

Table 7–3 shows how to define the following system logical names in the command procedure SYS$MANAGER:SYLOGICALS.COM to override the OPCOM default states.

Table 7–3 OPCOM System Logical Names

System Logical Name	Function
OPC$OPA0_ENABLE	If defined to be true, OPA0: is enabled as an operator. If defined to be false, OPA0: is not enabled as an operator. DCL considers any string beginning with T or Y or any odd integer to be true, all other values are false.

OPC$OPA0_CLASSES	This logical name defines the operator classes to be enabled on OPA0:. The logical name can be a search list of the allowed classes, a list of classes, or a combination of the two. For example:
	`$ DEFINE/SYSTEM OP$OPA0_CLASSES CENTRAL,DISKS,TAPE`
	`$ DEFINE/SYSTEM OP$OPA0_CLASSES "CENTRAL,DISKS,TAPE"`
	`$ DEFINE/SYSTEM OP$OPA0_CLASSES "CENTRAL,DISKS",TAPE`
	You can define OPC$OPA0_CLASSES even if OPC$OPA0_ENABLE is not defined. In this case, the classes are used for any operators that are enabled, but the default is used to determine whether to enable the operator.

OPC$LOGFILE_ENABLE	If defined to be true, an operator log file is opened. If defined to be false, no log file is opened.
OPC$LOGFILE_CLASSES	This logical name defines the operator classes to be enabled for the log file. The logical name can be a search list of the allowed classes, a comma-separated list, or a combination of the two. You can define this system logical even when the OPC$LOGFILE_ENABLE system logical is not defined. In this case, the classes are used for any log files that are open, but the default is used to determine whether to open the log file.
OPC$LOGFILE_NAME	This logical name supplies information that is used in conjunction with the default name SYS$MANAGER:OPERATOR.LOG to define the name of the log file. If the log file is directed to a disk other than the system disk, you should include commands to mount that disk in the SYLOGICALS.COM command procedure.

The OPCOM functions are described in more detail in the *OpenVMS System Manager's Manual.*

The following example shows how to use the OPC$OPA0_CLASSES system logical to define the operator classes to be enabled. The following command prevents SECURITY class messages from being displayed on OPA0:

```
$ DEFINE/SYSTEM OPC$OPA0_CLASSES CENTRAL,PRINTER,TAPES,DISKS,DEVICES, -
_$ CARDS,NETWORK,CLUSTER,LICENSE,OPER1,OPER2,OPER3,OPER4,OPER5, -
_$ OPER6,OPER7,OPER8,OPER9,OPER10,OPER11,OPER12
```

In large clusters, state transitions (computers joining or leaving the cluster) generate many multiline OPCOM messages on a boot server's console device. You can abbreviate such messages by including the DCL command REPLY /DISABLE=CLUSTER in the appropriate site-specific startup command file or by entering the command interactively from the system manager's account.

7.5.2 Removing a Computer from the Cluster

You must shut down a computer before removing it from the cluster. If possible, use the command procedure SYS$SYSTEM:SHUTDOWN.COM to perform an orderly shutdown. Otherwise, halt the computer.

Note

If your cluster uses DECdtm services, you must perform some extra tasks before you remove a computer from your cluster in order ensure the integrity of your data. These tasks are described in the chapter on DECdtm services in the *OpenVMS System Manager's Manual.*

If you are not sure whether your cluster uses DECdtm services, see Section 7.1.

Note that, because the REMOVE function deletes the computer's entire root directory tree, it generates OpenVMS RMS error messages while deleting directory files. You can ignore these messages.

Whenever you remove a voting member, you must, after the REMOVE function completes, reconfigure the cluster according to the instructions in Section 7.6.

Example 7–4 illustrates the use of CLUSTER_CONFIG.COM on JUPITR to remove satellite EUROPA from the cluster.

If the page and swap files for the computer being removed do not reside on the same disk as the computer's root directory tree, the REMOVE function does not delete these files. It displays a message warning that the files will not be deleted, as in Example 7–4. If you want to delete the files, you must do so after the REMOVE function completes.

Example 7–4 Sample Interactive CLUSTER_CONFIG.COM Session to Remove a Satellite with Local Page and Swap Files

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

           1. ADD a node to the cluster.
           2. REMOVE a node from the cluster.
           3. CHANGE a cluster node's characteristics.
           4. CREATE a second system disk for JUPITR.

    Enter choice [1]: 2
```

(continued on next page)

Example 7–4 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Remove a Satellite with Local Page and Swap Files

```
    The REMOVE function disables a node as a cluster member.

        o It deletes the node's root directory tree.

        o It removes the node's network information
          from the network database.

    If the node being removed is a voting member, you must adjust
    EXPECTED_VOTES in each remaining cluster member's MODPARAMS.DAT.
    You must then reconfigure the cluster, using the procedure described
    in the VMScluster Systems for OpenVMS Manual.

What is the node's DECnet node name? EUROPA
Verifying network database...
Verifying that SYS10 is EUROPA's root...

    WARNING - EUROPA's page and swap files will not be deleted.
              They do not reside on $1$DJA11:.

Deleting directory tree SYS10...
%DELETE-I-FILDEL, $1$DJA11:<SYS10>SYSCBI.DIR;1 deleted (1 block)
%DELETE-I-FILDEL, $1$DJA11:<SYS10>SYSERR.DIR;1 deleted (1 block)
   .
   .
   .
System root SYS10 deleted.
Updating network database...
The configuration procedure has completed successfully.
```

7.5.3 Changing a Computer's Characteristics

You select the CHANGE function when you want to accomplish any of the operations described in Table 7–4. When you select this function, CLUSTER_CONFIG.COM displays a menu of CHANGE options. Note that all operations except changing a satellite's LAN (Ethernet or FDDI) hardware address must be executed on the computer whose characteristics you want to change.

Before adding computers in a new local area or mixed-interconnect cluster, you must execute the CHANGE function to enable the first installed computer as a boot server (see Example 7–7).

Caution

Whenever you enable or disable disk-serving or tape-serving functions, you must run AUTOGEN with the REBOOT option to reboot the local computer. For all other change operations (except changing a satellite's hardware address), you must reconfigure the cluster according to the instructions in Section 7.6.

Table 7–4 CLUSTER_CONFIG.COM CHANGE Options

Option	Operation Performed
Enable the local computer as a disk server.	Load the MSCP server by setting, in MODPARAMS.DAT, the value of the MSCP_LOAD parameter to 1 and by setting an appropriate value for the MSCP_SERVE_ALL parameter.
Disable the local computer as a disk server.	Set MSCP_LOAD to 0.
Enable the local computer as a boot server.	If you are setting up a local area or mixed-interconnect cluster, you must execute this operation once before you attempt to add computers to the cluster. You thereby enable DECnet MOP service for the LAN adapter circuit that the computer uses to service operating system load requests from satellites. When you enable the computer as a boot server, it automatically becomes a disk server (if it is not one already), because it must serve its system disk to satellites.
Disable the local computer as a boot server.	Disable DECnet MOP service for the computer's adapter circuit.
Enable the LAN for cluster communications on the local computer.	Load the port driver PEDRIVER by setting the value of the NISCS_LOAD_PEA0 parameter to 1 in MODPARAMS.DAT. Create the cluster security database file, SYS$SYSTEM:[SYSEXE]CLUSTER_AUTHORIZE.DAT, on the local computer's system disk. **Caution:** The VAXCLUSTER system parameter must be set to 2 if the NISCS_LOAD_PEA0 parameter is set to 1. This ensures coordinated access to shared resources in the cluster and prevents accidental data corruption.
Disable the LAN for cluster communications on the local computer.	Set NISCS_LOAD_PEA0 to 0.
Enable a quorum disk on the local computer.	In MODPARAMS.DAT, set an appropriate value for the DISK_QUORUM system parameter; set the value of QDSKVOTES to 1 (default value).
Disable a quorum disk on the local computer.	In MODPARAMS.DAT, set a blank value for the DISK_QUORUM system parameter; set the value of QDSKVOTES to 1.
Change the local computer's allocation class value.	Set a value for the computer's ALLOCLASS parameter in MODPARAMS.DAT.

(continued on next page)

Table 7–4 (Cont.) CLUSTER_CONFIG.COM CHANGE Options

Option	Operation Performed
Change a satellite's LAN hardware address.	Change a satellite's hardware address if its LAN device needs replacement. Both the permanent and volatile network databases, and NETNODE_UPDATE.COM, are updated on the local computer. *You must execute this operation on a computer enabled as a boot server for the satellite.*
Enable the local computer as a tape server.	Load the TMSCP server by setting, in MODPARAMS.DAT, the value of the TMSCP_LOAD parameter to 1.
Disable the local computer as a tape server.	Set TMSCP_LOAD to 0.
Change the local computer's tape allocation class value.	Set a value for the computer's TAPE_ALLOCLASS parameter in MODPARAMS.DAT. The default value is 0. You must specify a nonzero tape allocation class parameter if this node is locally connected to a dual-ported tape, or if it will be serving any multiple-host tapes (for example, TF*nn* or HSC connected tapes) to other cluster members. Satellites usually have TAPE_ALLOCLASS set to 0.

Note

When CLUSTER_CONFIG.COM sets or changes values in MODPARAMS.DAT, the new values are always appended at the end of the file, so that they override earlier values. You may want to edit the file occasionally and delete lines that specify earlier values.

Examples 7–5 through 7–10 show the use of CLUSTER_CONFIG.COM to perform the following operations:

- Enable node URANUS as a disk server (Example 7–5).
- Change node URANUS's ALLOCLASS value (Example 7–6).
- Enable node URANUS as a boot server (Example 7–7).
- Specify a new hardware address for satellite node ARIEL, which boots from URANUS's system disk (Example 7–8).
- Enable node URANUS as a tape server (Example 7–9).
- Change node URANUS's TAPE_ALLOCLASS value (Example 7–10).

Example 7–5 Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Disk Server

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for URANUS.

    Enter choice [1]: 3

    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable Ethernet for cluster communications on URANUS.
       6. Disable Ethernet for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's ALLOCLASS value.
      10. Change a satellite's hardware address.
      11. Enable URANUS as a tape server.
      12. Disable URANUS as a tape server.
      13. Change URANUS's TAPE_ALLOCLASS value.

    Enter choice [1]: Return

Will URANUS serve HSC disks [Y]? Return
Enter a value for URANUS's ALLOCLASS parameter: 2
The configuration procedure has completed successfully.

    URANUS has been enabled as a disk server. MSCP_LOAD has been
    set to 1 in MODPARAMS.DAT. Please run AUTOGEN to reboot URANUS:

           $ @SYS$UPDATE:AUTOGEN GETDATA REBOOT

    If you have changed URANUS's ALLOCLASS value, you must reconfigure the
    cluster, using the procedure described in the VMScluster Systems for
    OpenVMS manual.
```

Example 7–6 Sample Interactive CLUSTER_CONFIG.COM Session to Change the Local Computer's ALLOCLASS Value

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.
```

(continued on next page)

Example 7–6 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Change the Local Computer's ALLOCLASS Value

```
        1. ADD a node to the cluster.
        2. REMOVE a node from the cluster.
        3. CHANGE a cluster node's characteristics.
        4. CREATE a second system disk for URANUS.

    Enter choice [1]: 3

    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable Ethernet for cluster communications on URANUS.
       6. Disable Ethernet for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's ALLOCLASS value.
      10. Change a satellite's hardware address.
      11. Enable URANUS as a tape server.
      12. Disable URANUS as a tape server.
      13. Change URANUS's TAPE_ALLOCLASS value.

    Enter choice [1]: 9

Enter a value for URANUS's ALLOCLASS parameter [2]: 1
The configuration procedure has completed successfully

    If you have changed URANUS's ALLOCLASS value, you must reconfigure the
    cluster, using the procedure described in VMScluster Systems for OpenVMS.
```

Example 7–7 Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Boot Server

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

        1. ADD a node to the cluster.
        2. REMOVE a node from the cluster.
        3. CHANGE a cluster node's characteristics.
        4. CREATE a second system disk for URANUS.

    Enter choice [1]: 3
```

(continued on next page)

Example 7–7 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Boot Server

```
    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable Ethernet for cluster communications on URANUS.
       6. Disable Ethernet for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's ALLOCLASS value.
      10. Change a satellite's hardware address.
      11. Enable URANUS as a tape server.
      12. Disable URANUS as a tape server.
      13. Change URANUS's TAPE_ALLOCLASS value.

    Enter choice [1]: 3

Verifying circuits in network database...
Updating permanent network database...

    In order to enable or disable DECnet MOP service in the volatile
    network database, DECnet traffic must be interrupted temporarily.

    Do you want to proceed [Y]? Return

Enter a value for URANUS's ALLOCLASS parameter [1]: Return
The configuration procedure has completed successfully.

    URANUS has been enabled as a boot server. Disk serving and
    Ethernet capabilities are enabled automatically. If URANUS was
    not previously set up as a disk server, please run AUTOGEN to
    reboot URANUS:

          $ @SYS$UPDATE:AUTOGEN GETDATA REBOOT

    If you have changed URANUS's ALLOCLASS value, you must reconfigure the
    cluster, using the procedure described in VMScluster Systems for OpenVMS.
```

Example 7–8 Sample Interactive CLUSTER_CONFIG.COM Session to Change a Satellite's Hardware Address

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

          1. ADD a node to the cluster.
          2. REMOVE a node from the cluster.
          3. CHANGE a cluster node's characteristics.
          4. CREATE a second system disk for URANUS.

    Enter choice [1]: 3
```

(continued on next page)

Example 7–8 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Change a Satellite's Hardware Address

```
    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable Ethernet for cluster communications on URANUS.
       6. Disable Ethernet for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's ALLOCLASS value.
      10. Change a satellite's hardware address.
      11. Enable URANUS as a tape server.
      12. Disable URANUS as a tape server.
      13. Change URANUS's TAPE_ALLOCLASS value.

    Enter choice [1]: 10

What is the node's DECnet node name? ARIEL
What is the new hardware address [XX-XX-XX-XX-XX-XX]? 08-00-3B-05-37-78
Updating network database...
The configuration procedure has completed successfully.
```

Example 7–9 Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Tape Server

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for URANUS.

    Enter choice [1]: 3

    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable Ethernet for cluster communications on URANUS.
       6. Disable Ethernet for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's ALLOCLASS value.
      10. Change a satellite's hardware address.
      11. Enable URANUS as a tape server.
      12. Disable URANUS as a tape server.
      13. Change URANUS's TAPE_ALLOCLASS value.
```

(continued on next page)

Example 7–9 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Tape Server

```
    Enter choice [1]: 11

Enter a value for URANUS's TAPE_ALLOCLASS parameter [1]:

    URANUS has been enabled as a tape server. TMSCP_LOAD has been
    set to 1 in MODPARAMS.DAT. Please run AUTOGEN to reboot URANUS:

           $ @SYS$UPDATE:AUTOGEN GETDATA REBOOT

    If you have changed URANUS's TAPE_ALLOCLASS value, you must reconfigure
    the cluster, using the procedure described in VMScluster Systems for OpenVMS.
```

Example 7–10 Sample Interactive CLUSTER_CONFIG.COM Session to Change the Local Computer's TAPE_ALLOCLASS Value

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

           1. ADD a node to the cluster.
           2. REMOVE a node from the cluster.
           3. CHANGE a cluster node's characteristics.
           4. CREATE a second system disk for URANUS.

    Enter choice [1]: 3

    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable Ethernet for cluster communications on URANUS.
       6. Disable Ethernet for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's ALLOCLASS value.
      10. Change a satellite's hardware address.
      11. Enable URANUS as a tape server.
      12. Disable URANUS as a tape server.
      13. Change URANUS's TAPE_ALLOCLASS value.

    Enter choice [1]: 13

Enter a value for URANUS's TAPE_ALLOCLASS parameter [1]: 2

    If you have changed URANUS's TAPE_ALLOCLASS value, you must reconfigure
    the cluster, using the procedure described in VMScluster Systems for OpenVMS.)
```

7.5.4 Changing the Cluster Configuration Type

As your processing needs change, you may want to add satellites to an existing CI or DSSI cluster, or you may want to add computers that are based on the CI or DSSI interconnects (or HSC subsystems) to an existing local area cluster. In either case, you can use CLUSTER_CONFIG.COM to help you change the configuration of your existing cluster.

7.5.4.1 Changing an Existing CI or DSSI Cluster to a Mixed-Interconnect Configuration

If you want to convert an existing CI or DSSI cluster to a mixed-interconnect configuration, you must enable cluster communications over the Ethernet interconnect on all computers, and you must enable one or more computers as boot servers. Proceed as follows:

1. Log in as system manager on each computer, invoke CLUSTER_CONFIG.COM, and execute the CHANGE function to enable Ethernet communications. *You must perform this operation on all computers.*

   Note

   You must establish a cluster group number and password on all system disks in the VMScluster before you can successfully add a node using the the CHANGE function of the CLUSTER_CONFIG.COM procedure.

2. Execute the CHANGE function to enable one or more computers as boot servers.
3. Shut down and reboot the cluster, following instructions in Section 7.6.

7.5.4.2 Changing an Existing Local Area Cluster to a Mixed-Interconnect Configuration

Before performing the operations described in this section, be sure that the computers and HSC subsystems you intend to include in your new configuration are correctly installed and checked for proper operation.

The method you use to convert an existing local area cluster to a mixed-interconnect configuration depends on whether your current boot server is capable of being configured as a CI or DSSI computer. Note that the following procedures assume that the system disk containing satellite roots will reside on an HSC disk (for CI configurations) or an RF disk (for DSSI configurations).

- If the boot server is capable of being configured as a CI or DSSI computer, proceed as follows:

  1. Log in as system manager on the boot server and perform an image backup operation to back up the current system disk to a disk on an HSC subsystem or RF storage device. (For more information about backup operations, refer to the *OpenVMS System Management Utilities Reference Manual*.)
  2. Modify the computer's default bootstrap command procedure to boot the computer from the HSC or RF disk, following instructions in the appropriate system-specific installation and operations guide.
  3. Shut down the cluster. Shut down the satellites first, and then shut down the boot server.
  4. Boot the boot server from the newly created system disk on the HSC or RF storage subsystem.
  5. Reboot the satellites.

  If your current boot server cannot be configured as a CI or a DSSI computer, proceed as follows:

  1. Shut down the old local area cluster. Shut down the satellites first, then shut down the boot server.

2. Install the OpenVMS operating system on the new CI computer's HSC system disk or on the new DSSI computer's RF disk, as appropriate. When the installation procedure asks whether you want to enable the LAN for cluster communications, answer YES.
3. When the installation completes, log in as system manager and configure and start the DECnet for OpenVMS network, as described in Chapter 4.
4. Execute the CLUSTER_CONFIG.COM CHANGE function to enable the computer as a boot server.
5. Log in as system manager on the newly added computer and execute CLUSTER_CONFIG.COM's ADD function to add the former local area cluster members (including the former boot server) as satellites.

7.5.5 Converting a Standalone Computer to a VMScluster Computer

You execute CLUSTER_CONFIG.COM on a standalone computer to perform the following operations:

- Add the standalone computer to an existing cluster.
- Set up the standalone computer to form a new cluster, if the computer was not set up as a cluster computer during installation of the operating system.

If your cluster uses DECdtm services, you must create a transaction log for the computer when you have configured it into your cluster. For step-by-step instructions on how to do this, see the chapter on DECdtm services in the *OpenVMS System Manager's Manual*.

If you are not sure whether your cluster uses DECdtm services, see Section 7.1.

Example 7–11 illustrates the use of CLUSTER_CONFIG.COM on standalone computer PLUTO to convert PLUTO to a cluster boot server.

Example 7–11 Sample Interactive CLUSTER_CONFIG.COM Session to Convert a Standalone Computer to a Cluster Boot Server

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    This procedure sets up this standalone node to join an existing
    cluster or to form a new cluster.

What is the node's DECnet node name? PLUTO
What is the node's DECnet address? 2.5
Will the Ethernet be used for cluster communications (Y/N)? Y
Enter this cluster's group number: 3378
Enter this cluster's password:
Re-enter this cluster's password for verification:
Will PLUTO be a boot server [Y]? Return
Verifying circuits in network database...
Enter a value for PLUTO's ALLOCLASS parameter: 1
Does this cluster contain a quorum disk [N]? Return

    AUTOGEN computes the SYSGEN parameters for your configuration
    and then reboots the system with the new parameters.
```

7.5.6 Creating a Duplicate System Disk

As you continue to add AXP computers running on an AXP common system disk or VAX computers running on a VAX common system disk, you might eventually reach the disk's storage or I/O capacity. In that case, you might want to add one or more common system disks to handle the increased load. (Remember that a system disk cannot be shared between VAX and AXP computers.) You can use CLUSTER_CONFIG.COM to set up these disks. Proceed as follows *after* you have coordinated cluster common files as described in Section 4.5.4.

1. Log in as system manager.
2. Place a blank disk in an appropriate drive and spin up the disk.
3. Invoke CLUSTER_CONFIG.COM and select the CREATE function. The procedure prompts for the device names of the current and new system disks, as shown in Example 7–12. It then backs up the current system disk to the new one, deletes all directory roots from the new disk, and mounts that disk clusterwide. Note that OpenVMS RMS error messages are displayed while the procedure deletes directory files. You can ignore these messages.

Example 7–12 Sample Interactive CLUSTER_CONFIG.COM CREATE Session

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a VMScluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for JUPITR.

    Enter choice [1]: 4

    The CREATE function generates a duplicate system disk.

            o It backs up the current system disk to the new system disk.

            o It then removes from the new system disk all system roots.

    WARNING - Do not proceed unless you have defined appropriate
              logical names for cluster common files in your
              site-specific startup procedures. For instructions,
              see the VMScluster Systems for OpenVMS manual.

              Do you want to continue [N]? YES

    This procedure will now ask you for the device name of JUPITR's system root.
    The default device name (DISK$VAXVMSRL5:) is the logical volume name of
    SYS$SYSDEVICE:.

What is the device name of the current system disk [DISK$VAXVMSRL5:]? Return
```

(continued on next page)

Example 7–12 (Cont.) Sample Interactive CLUSTER_CONFIG.COM CREATE Session

```
What is the device name for the new system disk? $1$DJA16:
%DCL-I-ALLOC, _$1$DJA16: allocated
%MOUNT-I-MOUNTED, SCRATCH mounted on _$1$DJA16:
What is the unique label for the new system disk [JUPITR_SYS2]? Return

Backing up the current system disk to the new system disk...

Deleting all system roots...

            Deleting directory tree SYS1...

%DELETE-I-FILDEL, $1$DJA16:<SYS0>DECNET.DIR;1 deleted (2 blocks)
   .
   .
   .
System root SYS1 deleted.

            Deleting directory tree SYS2...

%DELETE-I-FILDEL, $1$DJA16:<SYS1>DECNET.DIR;1 deleted (2 blocks)
   .
   .
   .
System root SYS2 deleted.

All the roots have been deleted.
%MOUNT-I-MOUNTED, JUPITR_SYS2  mounted on _$1$DJA16:

The second system disk has been created and mounted clusterwide.
Satellites can now be added.
```

7.6 Reconfiguring the Cluster After a Major Change

The following operations affect the integrity of the entire cluster. Table 7–5 describes the action you should take after executing each operation.

Table 7–5 Actions Required to Reconfigure a Cluster

Operation	Action
Adding or removing a voting member	Update MODPARAMS.DAT files, then shut down and reboot the entire cluster.
Enabling or disabling the LAN for cluster communications	Reboot the node on which you have enabled or disabled the LAN.
Enabling or disabling a quorum disk	Update MODPARAMS.DAT files and reboot all nodes that specify the quorum disk.
Changing allocation class values	Shut down and reboot the entire cluster.
Changing the cluster group number or password (see Section 7.7.8)	Shut down and reboot the entire cluster.

To perform the following reconfiguration tasks, refer to the corresponding section in this manual:

- Updating the MODPARAMS.DAT files (Section 7.6.1)
- Shutting down the VMScluster (Section 7.6.2)
- Changing the allocation class values (Section 7.6.3)
- Rebooting the VMScluster (Section 7.6.5)

7.6.1 Updating MODPARAMS.DAT Files to Adjust Cluster Quorum

Whenever you add or remove a voting cluster member, or whenever you enable or disable a quorum disk, you must edit MODPARAMS.DAT in all other cluster members' [SYS*x*.SYSEXE] directories and adjust the value for the EXPECTED_VOTES system parameter appropriately. For example, if you add a voting member or if you enable a quorum disk, you must increment the value by the number of votes assigned to the new member (usually 1). If you add a voting member with one vote and enable a quorum disk with one vote on that computer, you must increment the value by 2.

Then, use the DCL command SET CLUSTER/EXPECTED_VOTES equal to the number of votes contributed by each node in the cluster plus the number of votes contributed by the cluster quorum disk.

To ensure that the new values take effect when you reboot, log in as system manager at each computer and run AUTOGEN to propagate the values to the computer's ALPHAVMSSYS.PAR (or VAXVMSSYS.PAR for VAX computers). Enter the following command:

```
$ @SYS$UPDATE:AUTOGEN GETDATA SETPARAMS
```

Be sure *not* to specify the SHUTDOWN or REBOOT option.

Caution

Do not perform this operation until you are ready to shut down and reboot the entire cluster. If a computer fails and then reboots with the new parameters, normal cluster operations can be seriously compromised.

7.6.2 Shutting Down the Cluster

After you have run AUTOGEN to set parameter values correctly, you must shut down the entire cluster. Shut down nonvoting members (such as satellites) before shutting down voting members. Log in as system manager on each computer *locally* and enter the following command to perform an orderly shutdown:

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

When you are prompted for shutdown options, specify CLUSTER_SHUTDOWN. Note that you must run the shutdown procedure and specify this option on each computer. When all computers have reached a point in the procedure where activity is suspended, you must halt each computer at its console. You cannot shut down the entire cluster from one computer. (For more information about the CLUSTER_SHUTDOWN option, see Section 7.7.6.2.)

7.6.3 Changing Allocation Class Values on HSC Subsystems

If you must change allocation class values on any HSC subsystem, you must do so while the entire cluster is shut down. For example, to change the allocation class value to 1, set the HSC internal door switch to the Enable position and enter a command sequence like the following at the appropriate HSC consoles:

```
Ctrl/C
HSC> RUN SETSHO
SETSHO> SET ALLOCATE DISK 1
SETSHO> EXIT
SETSHO-Q Rebooting HSC; Y to continue, Ctrl/Y to abort:? Y
```

Restore the HSC internal door switch setting.

7.6.4 Changing Allocation Class Values on DSSI Subsystems

If it is necessary to change allocation class values on any DSSI subsystem, you must enter a command sequence.

AXP

On AXP systems, to change the allocation class value to 1 for a DSSI node TRACER from the AXP node, enter the following command at the system prompt ($) from the SYSTEM account of the AXP node:

```
$ MC SYSMAN IO CONN FYA0:/NOADAP/DRIVER=SYS$FYDRIVER♦
```

VAX

On VAX systems, to change the allocation class value to 1 for a DSSI node TRACER from a VAX node, enter the following command at the system prompt ($) from the SYSTEM account of the VAX node:

```
$ MCR SYSGEN CONN FYA0:/NOADAP/DRIVER=FYDRIVER♦
```

A subsequent SHOW/DEVICE=FY command display indicates the device is off line, as follows:

```
$ SHOW/DEVICE=FY

Device  Device  Error
Name  Status  Count

FYA0:   offline           0
```

Then, enter the following command sequence to set the allocation class value to 1:

```
$ SET HOST/DUP/SERVER=MSCP$DUP/TASK=PARAMS TRACER
params >set allclass 1

params >write
Changes require controller initialization, ok?[Y/N]Y

Initializing...

%HSCPAD-S-REMPGMEND, Remote program terminated - message number 3.
%PAxx, Port has closed virtual circuit - remote node TRACER

%HSCPAD-S-END, control returned to node node-name

$
```

On a directly connected CPU, the computer must be rebooted after the value is changed.

7.6.5 Rebooting the Cluster

For VMScluster configurations with HSC subsystems, reboot each computer after all HSC subsystems have been set and rebooted, Watch the console listings for unusual messages or warnings.

Caution

In local area and mixed-interconnect clusters, you must reboot boot servers before rebooting satellites.

Note that several new messages might appear. For example, if you have used the CLUSTER_CONFIG.COM CHANGE function to enable cluster communications over the LAN, one message reports that the local area VMScluster security database is being loaded. Then, for every disk-serving computer, another message reports that the MSCP server is being loaded. This message is followed by a list of all the disks being served by that computer. You should verify that all disks are being served in the manner that you specified when you designed the configuration.

For DSSI VMScluster configurations, after all the DSSI subsystems have been set, reboot the system. Watch the console listings for unusual messages or warnings. Note that several new messages may appear. For every disk-serving computer, a message reports that the MSCP server is being loaded. Note that, for DSSI VMScluster environments, there is no message containing the list of all the disks being served by the computer. To verify that all disks are being served in the manner in which you designed the configuration, at the system prompt ($) of the node serving the disks, enter the following command:

```
$ SHOW DEVICE/SERVED

Device:   Status   Total Size   Current  Max Hosts
$1$DIA0    Avail      1954050         0    0         0
$1$DIA2    Avail      1800020         0    0         0
```

7.7 Maintaining the Cluster

Once your cluster is up and running, you can implement routine site-specific maintenance operations—for example, backing up disks or adding user accounts. You should plan to run AUTOGEN with the FEEDBACK option on a regular basis, as described in Section 7.7.1.

You should also maintain records of current configuration data, especially any changes to hardware or software components. Section 7.7.2 lists items that should be included in your records.

If you are managing a local area or mixed-interconnect cluster, it is important to monitor LAN activity. Section 7.7.3 provides information to help you set up a monitoring procedure.

From time to time conditions may occur that require the following special maintenance operations:

- Restoring cluster quorum after an unexpected computer failure
- Executing conditional shutdown operations
- Performing security functions in local area and mixed-interconnect clusters

These operations are discussed in Section 7.7.5, Section 7.7.6, and Section 7.7.8, respectively.

7.7.1 Running AUTOGEN with the FEEDBACK Option

AUTOGEN includes a mechanism called **feedback**. This mechanism examines data collected during normal system operations, and it adjusts system parameters on the basis of the collected data whenever you run AUTOGEN with the FEEDBACK option. For example, the system records each instance of a disk server waiting for buffer space to process a disk request. Based on this information, AUTOGEN can size the disk server's buffer pool automatically to ensure that sufficient space is allocated.

Digital strongly recommends that you use the FEEDBACK option. Without FEEDBACK, it is difficult for AUTOGEN to anticipate patterns of resource usage, particularly in complex configurations. Factors such as the number of computers and disks in the cluster and the types of applications being run require adjustment of system parameters for optimal performance.

You should run AUTOGEN with FEEDBACK frequently. As a cluster grows, settings for many parameters must be adjusted. The settings AUTOGEN chooses for a cluster with three CI computers and five satellites will no longer be appropriate when you add more computers or satellites. In summary, you should run AUTOGEN on a regular basis to compensate for changes in user work loads and whenever you make significant changes in your configuration. For detailed information about AUTOGEN, refer to the *OpenVMS System Manager's Manual*.

7.7.2 Recording Configuration Data

To maintain a VMScluster system effectively, you must keep accurate records about the current status of all hardware and software components and about any changes made to those components. Changes to cluster components can have a significant effect on the operation of the entire cluster. If a failure occurs, you will need to consult your records to diagnose problems.

At a minimum, your configuration records should include the following information:

- SCSNODE and SCSSYSTEMID parameter values for all computers.
- DECnet names and addresses for all computers.
- Current values for cluster-related system parameters, especially ALLOCLASS and TAPE_ALLOCLASS values for HSC subsystems and computers. (Cluster system parameters are described in Appendix A.)
- Names and locations of default bootstrap command procedures for all computers connected with the CI.
- Names of cluster disk and tape devices.
- In local area and mixed-interconnect clusters, LAN hardware addresses for satellites.
- Names of LAN adapters.
- Names of LAN segments or rings.
- Names of LAN bridges.

- Names of wiring concentrators or of DELNI or DEMPR adapters.
- Serial numbers of all hardware components.
- Changes to any hardware or software components (including site-specific command procedures), along with dates and times when changes were made.

Maintaining current records for your configuration is necessary both for routine operations and for eventual troubleshooting activities. (Section E.2.1 describes how to collect information for VMScluster network failure analysis.)

7.7.3 Monitoring LAN Activity

It is important that you monitor LAN (Ethernet or FDDI) activity on a regular basis. Using NCP commands like the following, you can set up a convenient monitoring procedure to report activity for each 12-hour period. Note that DECnet event logging for event 0.2 (automatic line counters) must be enabled. (For detailed information on DECnet for OpenVMS event logging, refer to the *DECnet for OpenVMS Network Management Utilities* manual.) In these sample commands, BNA-0 is the line ID of the Ethernet line.

```
NCP> DEFINE LINE BNA-0 COUNTER TIMER 43200
NCP> SET LINE BNA-0 COUNTER TIMER 43200
```

At every timer interval (in this case, 12 hours), DECnet will create an event that sends counter data to the DECnet event log. If you experience a performance degradation in your cluster, check the event log for increases in counter values that exceed normal variations for your cluster. If all computers show the same increase, there may be a general problem with your Ethernet configuration. If, on the other hand, only one computer shows a deviation from usual values, there is probably a problem with that computer or with its Ethernet interface device.

The following Digital layered products can be used in conjunction with one of Digital's LAN bridges to monitor the LAN traffic levels:

- RBMS
- DECelms
- DECmcc
- LAN Traffic Monitor (LTM)

7.7.4 Performing VMScluster Network Failure Analysis

The operating system provides a sample program in SYS$EXAMPLES to help you analyze local area VMScluster network failures. You can edit and use the program to detect and isolate failed network components. Using the network failure analysis program can help reduce the time required to detect and isolate a failed network component, thereby providing a significant increase in cluster availability. For a description of the network failure analysis program, refer to Appendix E.

7.7.5 Restoring Cluster Quorum After an Unexpected Computer Failure

During the life of a VMScluster system, computers join and leave the cluster. For example, you may need to add more computers to the cluster to extend the cluster's processing capabilities, or a computer may shut down unexpectedly because of a hardware or fatal software error. The connection management software coordinates these cluster transitions and controls cluster operation.

When a computer shuts down unexpectedly, the remaining computers, with the help of the connection manager, reconfigure the cluster, excluding the computer that shut down. The cluster can survive the failure of the computer and continue process operations, as long as the cluster votes total is greater than the cluster quorum value. If the cluster votes total falls below the cluster quorum value, the cluster suspends the execution of all processes.

For process execution to resume, the cluster votes total must be restored to a value greater than or equal to the cluster quorum value. Often, the required votes are added as computers join or rejoin the cluster. However, waiting for a computer to join the cluster and increasing the votes value is not always a simple or convenient remedy. An alternative solution, for example, might be to shut down and reboot all the computers with a lower quorum value.

After the failure of a computer, you may want to run the Show Cluster utility and examine values for the VOTES, EXPECTED_VOTES, CL_VOTES, and CL_QUORUM fields. (See the *OpenVMS System Management Utilities Reference Manual* for a complete description of these fields.) The VOTES and EXPECTED_VOTES fields show the settings for each cluster member; the CL_VOTES and CL_QUORUM fields show the cluster votes total and the current cluster quorum value.

To examine these values, enter the following commands:

```
$ SHOW CLUSTER/CONTINUOUS
COMMAND> ADD VOTES,EXPECTED_VOTES,CL_VOTES,CL_QUORUM
```

Note

If you want to enter SHOW CLUSTER commands interactively, you must specify the /CONTINUOUS qualifier as part of the SHOW CLUSTER command string. If you do not specify this qualifier, SHOW CLUSTER displays cluster status information returned by the DCL command SHOW CLUSTER and returns you to the DCL command level.

If the display from the Show Cluster utility shows the CL_VOTES value equal to the CL_QUORUM value, the cluster cannot survive the failure of any remaining voting member. If one of these computers shuts down, all process activity in the cluster stops.

To prevent the disruption of cluster process activity, you can lower the cluster quorum value. You can use the DCL command SET CLUSTER/EXPECTED_VOTES to adjust the cluster quorum to a value you specify. If you do not specify a value, the operating system calculates an appropriate value for you. You need enter the command on only one computer to propagate the new value throughout the cluster. When you enter the command, the operating system reports the new value.

Normally, you use the SET CLUSTER/EXPECTED_VOTES command only when a computer is leaving the cluster for an extended period. (For more information about this command, see the *OpenVMS DCL Dictionary*.)

For example, if you want to change expected votes to set the cluster quorum to 2, enter the following command:

```
$ SET CLUSTER/EXPECTED_VOTES=3
```

The resulting value is (3 + 2) /2 = 2.

Note that no matter what value you specify for the SET CLUSTER/EXPECTED_VOTES command, you cannot increase quorum to a value that is greater than the number of the votes present, nor can you reduce quorum to a value that is half or fewer of the votes present.

To make the new value active throughout the cluster, you must adjust the EXPECTED_VOTES system parameter in MODPARAMS.DAT files on each VMScluster computer and then reconfigure the cluster according to the instructions in Section 7.6.

When a computer that previously was a cluster member is ready to rejoin, you must reset the EXPECTED_VOTES system parameter to its original value in MODPARAMS.DAT on all computers, and then reconfigure the cluster according to the instructions in Section 7.6. You do not need to use the SET CLUSTER /EXPECTED_VOTES command to increase cluster quorum, because the quorum value is increased automatically when the computer rejoins the cluster.

You can also reduce cluster quorum by selecting one of the cluster-related shutdown options described in Section 7.7.6.

7.7.6 Selecting Cluster Shutdown Options

In addition to the default shutdown option NONE, the OpenVMS AXP and OpenVMS VAX operating systems provide the following options for shutting down VMScluster computers:

- REMOVE_NODE
- CLUSTER_SHUTDOWN
- REBOOT_CHECK
- SAVE_FEEDBACK

These options are described in Section 7.7.6.1, Section 7.7.6.2, Section 7.7.6.3, and Section 7.7.6.4, respectively.

In addition, in response to the "Shutdown options [NONE]:" prompt, you can specify the DISABLE_AUTOSTART=*n* option, where *n* is the number of minutes before autostart queues are disabled in the shutdown sequence. See Section 6.7 for more information.

If you do not select any of these options (that is, if you select the default SHUTDOWN option NONE), the SHUTDOWN procedure performs the normal operations for shutting down a standalone computer. If you want to shut down a computer that you expect will rejoin the cluster shortly, you can specify the default option NONE. In that case, cluster quorum is not adjusted because the operating system assumes that the computer will soon rejoin the cluster.

7.7.6.1 REMOVE_NODE Option

If you want to shut down a computer that you expect will not rejoin the cluster for an extended period, use the REMOVE_NODE option. For example, a computer may be waiting for new hardware, or you may decide that you want to use a computer for standalone operation indefinitely.

When you use the REMOVE_NODE option, the active quorum in the remainder of the cluster is adjusted downward to reflect the fact that the removed computer's votes no longer contribute to the quorum value. The SHUTDOWN procedure readjusts the quorum by issuing the SET CLUSTER/EXPECTED_VOTES command, which is subject to the usual constraints described in Section 7.7.5.

Note that the system manager is still responsible for changing the EXPECTED_VOTES system parameter on the remaining VMScluster computers to reflect the new configuration.

7.7.6.2 CLUSTER_SHUTDOWN Option

When you choose the CLUSTER_SHUTDOWN option, each computer suspends activity, just short of shutting down completely, until all other computers in the cluster have reached the same point in the SHUTDOWN procedure.

You must specify this option on every VMScluster computer, and then shut down every computer individually by halting each computer at its console. If any one computer is not shut down completely, clusterwide shutdown cannot occur. Instead, operations on all other computers are suspended.

Note

Be sure to shut down nonvoting members (such as satellite nodes) before shutting down other computers.

7.7.6.3 REBOOT_CHECK Option

When you choose the REBOOT_CHECK option, the SHUTDOWN procedure checks for the existence of basic system files that are needed to reboot the computer successfully and notifies you if any files are missing. You should replace such files before proceeding. If all files are present, the following informational message appears:

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

Note that you can use the REBOOT_CHECK option separately or in conjunction with either the REMOVE_NODE or the CLUSTER_SHUTDOWN option. If you choose REBOOT_CHECK with one of the other options, you must specify the options in the form of a comma-separated list.

7.7.6.4 SAVE_FEEDBACK Option

Use the SAVE_FEEDBACK option to enable AUTOGEN feedback operation. Note that you should select this option only when a computer has been running long enough to reflect your typical work load. For detailed information about AUTOGEN feedback, see the *OpenVMS System Manager's Manual*.

7.7.7 Rebooting a Satellite with an Operating System on a Local Disk

In some circumstances, cluster software reboots satellites automatically. Before booting a satellite, the boot procedures check for the presence of an operating system on the satellite's local disk. If an operating system is found, that "local" operating system—not the VMScluster operating system—is booted.

If an operating system is installed on a satellite's local disk, you should take one of the following measures before performing any operation that causes an automatic reboot—for example, executing SYS$SYSTEM:SHUTDOWN.COM with the REBOOT option or using CLUSTER_CONFIG.COM to add that satellite to the cluster:

- Rename the directory file *ddcu*:[000000]SYS0.DIR on the local disk to *ddcu*:[000000]SYS*x*.DIR (where SYS*x* is a root other than SYS0, SYSE, or SYSF). Then enter the DCL command SET FILE/REMOVE as follows to remove the old directory entry for the boot image SYSBOOT.EXE:

```
$ RENAME DUA0:[000000]SYS0.DIR DUA0:[000000]SYS1.DIR
$ SET FILE/REMOVE DUA0:[SYSEXE]SYSBOOT.EXE
```

VAX

On VAX systems, for subsequent reboots of VAX computers from the local disk, enter a command in the format B/*x*0000000 at the console-mode prompt (>>>). For example:

```
>>> B/10000000♦
```

- Disable the local disk. For instructions, refer to your computer-specific installation and operations guide. Note that this option is not available if the satellite's local disk is being used for paging and swapping.

7.7.8 Maintaining the Integrity of VMScluster Membership

Because multiple local area and mixed-interconnect clusters coexist on a single extended LAN, the operating system provides mechanisms to ensure the integrity of individual clusters and to prevent access to a cluster by an unauthorized computer.

VMScluster systems use mechanisms to protect the integrity of the cluster to prevent problems that could otherwise occur under circumstances like the following:

- When setting up a new cluster, the system manager specifies a group number identical to that of an existing cluster on the same Ethernet. (This condition is not as unlikely as it may at first appear, because system managers probably do not assign group numbers randomly.) However, if each cluster's password is unique, the new cluster can form independently.
- A satellite user with access to a local system disk tries to join a cluster by executing a conversational SYSBOOT operation at the satellite's console.

The following mechanisms are designed to help system managers ensure the integrity of the cluster:

- A cluster authorization file (SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT), which is initialized during installation of the operating system or during execution of the CLUSTER_CONFIG.COM CHANGE function. The file is maintained with the SYSMAN utility.
- Control of conversational bootstrap operations on satellites.

These mechanisms are discussed in Section 7.7.8.1 and Section 7.7.8.2, respectively.

7.7.8.1 Maintaining Cluster Group Data

The cluster authorization file, SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT, contains the cluster group number and (in encrypted form) the cluster password. The CLUSTER_AUTHORIZE.DAT file is accessible only to users with the SYSPRV privilege.

The purpose of the cluster group number and password is to prevent accidental access to the cluster by an unauthorized computer. Under normal conditions, the system manager specifies the cluster group number and password either during installation or when you run CLUSTER_CONFIG.COM (see Section 7.5.5) to convert a standalone computer to run in a VMScluster system.

Under normal conditions, you need not alter records in the CLUSTER_AUTHORIZE.DAT file interactively. However, if you suspect a security breach, you may want to change the cluster password. In that case, you use the SYSMAN utility to make the change.

Note that if your configuration has multiple system disks, each disk must have a copy of CLUSTER_AUTHORIZE.DAT. You must run the SYSMAN utility to update all copies.

Caution

If you change either the group number or the password, you must reboot the entire cluster. For instructions, see Section 7.6.

To invoke the SYSMAN utility, log in as system manager on a boot server and enter the following command:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN>
```

At the SYSMAN> prompt, you can enter any of the CONFIGURATION commands listed in Table 7–6.

Table 7–6 Summary of SYSMAN CONFIGURATION Commands for Cluster Authorization

Command	Qualifiers	Function
HELP CONFIGURATION SET CLUSTER_AUTHORIZATION	None	Explains the command's functions.
CONFIGURATION SET CLUSTER_AUTHORIZATION		Updates the cluster authorization file, CLUSTER_AUTHORIZE.DAT, in the directory SYS$COMMON:[SYSEXE]. (The SET command creates this file if it does not already exist.)
	/GROUP_NUMBER	Specifies a cluster group number. Group number must be in the range from 1 to 4095 or 61440 to 65535.
	/PASSWORD	Specifies a cluster password. Password may be from 1 to 31 characters in length and may include alphanumeric characters, dollar signs ($), and underscores (_).
CONFIGURATION SHOW CLUSTER_AUTHORIZATION	None	Displays the cluster group number.

Example 7–13 illustrates the use of the SYSMAN utility to change the cluster password.

Example 7–13 Sample SYSMAN Session to Change the Cluster Password

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username LAZARUS       will be used on nonlocal nodes
SYSMAN> SET PROFILE/PRIVILEGES=SYSPRV
SYSMAN> CONFIGURATION SET CLUSTER_AUTHORIZATION/PASSWORD=NEWPASSWORD
%SYSMAN-I-CAFOLDGROUP, existing group will not be changed
%SYSMAN-I-CAFREBOOT, cluster authorization file updated
 The entire cluster should be rebooted.
SYSMAN> EXIT
$
```

7.7.8.2 Controlling Conversational Bootstrap Operations for Satellites

When you add a satellite to the cluster using CLUSTER_CONFIG.COM, the procedure asks whether you want to allow conversational bootstrap operations for the satellite (default is NO). If you press the Return key, the NISCS_CONV_BOOT system parameter in the satellite's system parameter file remains set to 0 to disable such operations. The parameter file (ALPHAVMSSYS.PAR for AXP systems or VAXVMSSYS.PAR for VAX systems) resides in the satellite's root directory on a boot server's system disk (*device*:[SYS*x*.SYSEXE]). You later can enable conversational bootstrap operations for a given satellite at any time by setting this parameter to 1.

For example, to enable such operations for an OpenVMS VAX satellite booted from root 10 on device 1DJA11, you would proceed as follows:

1. Log in as system manager on the boot server.

VAX

2. On VAX systems, invoke the System Generation utility (SYSGEN) and enter the following commands:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE $1$DJA11:[SYS10.SYSEXE]VAXVMSSYS.PAR
SYSGEN> SET NISCS_CONV_BOOT 1
SYSGEN> WRITE $1$DJA11:[SYS10.SYSEXE]VAXVMSSYS.PAR
SYSGEN> EXIT
$♦
```

3. Modify the satellite's MODPARAMS.DAT file so that NISCS_CONV_BOOT is set to 1.

AXP

On an AXP satellite, enter the same commands, replacing VAXVMSSYS.PAR with ALPHAVMSSYS.PAR.♦

7.8 Guidelines for Configuring Large Clusters

This section provides guidelines for configuring VMScluster systems that include many computers—approximately 20 or more—and describes procedures that you might find helpful. Typically, such VMScluster systems are local area or mixed-interconnect configurations with a large number of satellites. Topics include the following:

- Configuring disk server LAN adapters and memory
- Configuring system disks
- Adding computers to an existing cluster
- Setting up a new large cluster

- Defining the VMScluster alias

Note that the recommendations in Section 7.8.1, Section 7.8.2, and Section 7.8.3 can prove beneficial in some clusters with fewer than 20 computers.

7.8.1 Booting Local Area VMScluster Satellites

VMScluster satellite nodes use a single LAN adapter for the initial stages of booting. This section describes how to choose this adapter and how to configure MOP servers. It also provides troubleshooting support for the early stages of booting. The procedures and utilities for configuring and booting satellite nodes are the same or vary only slightly between AXP and VAX systems.

Complete the items in the following checklist before proceeding with satellite booting:

1. If the VMScluster configuration includes both VAX and AXP nodes, remember that the DECnet database must not be shared between AXP and VAX systems in the cluster. Create separate remote DECnet databases (NETNODE_REMOTE.DAT) for VAX and AXP nodes. See Section 4.3 for more information about configuring DECnet networks.

AXP

2. In an OpenVMS AXP network database, specify the APB.EXE downline load file. Do not specify the SYS$SYSTEM:TERTIARY_VMB.EXE tertiary loader as you would for VAX systems. The APB.EXE downline load file is different from the one used for VAX booting.♦

VAX

   In an OpenVMS VAX network database, specify the SYS$SYSTEM:TERTIARY_VMB.EXE tertiary loader. Do not specify the APB.EXE load file as you would for AXP systems.♦

3. If the MOP server node and system disk server node (AXP or VAX) are not already configured as cluster members, follow the directions in Section 7.5 for using the CLUSTER_CONFIG.COM procedure to configure each of the AXP nodes.

4. Run the CLUSTER_CONFIG.COM procedure on the AXP or VAX node for each satellite you want to boot into the VMScluster.

When the nodes are configured as VMScluster members, boot the satellite using the commands described in Section 7.8.1.2.

7.8.1.1 Booting from a Single LAN Adapter

Both AXP and VAX systems support booting from any LAN adapter on a local area VMScluster satellite. You can boot from a specific adapter to work around broken adapters or network problems. You can also use this feature to boot into different clusters, depending on the adapter you use to boot the system.

AXP

The following example shows the boot command on an AXP system. You must enter the command line using lowercase letters at the console prompt (cs>>>).

```
cs>>> b -flags 0,1 eza0
```

In the example:

- The designation -flags stands for the flags command line qualifier, which takes two values: the root number and the conversational boot flag.
- When booting from a disk, the "0" tells the console to use system root [SYS0]. However, for satellite booting, the system root comes from the network database and the 0 is ignored but must be present as a placeholder. In either case, the "1" indicates that the boot should be conversational.

- The designation eza0 is the LAN adapter to be used for booting.

Finally, notice that a load file is not specified in this boot command line. For satellite booting, the load file is part of the node description in the DECnet database. Because a file name is not specified on the boot command, the Alpha primary bootstrap (APB) attempts to use the NISCA transport protocol to access the remote system disk.♦

VAX

To boot a cluster system from a single, alternate adapter on a VAX system, specify the full device name in the boot command, as shown:

```
>>>B XQB0♦
```

If the boot fails, reenter the boot command using the same LAN adapter. If the configuration permits, and the network database is properly set up, then it might be possible to boot from another LAN adapter. See Section C.1.3.4 for information about troubleshooting satellite booting problems.

7.8.1.2 Alternate Adapter Booting

AXP and VAX computers support booting from any LAN adapter on a local area VMScluster satellite with multiple LAN adapters. You can use alternate adapter booting to work around broken adapters and network problems. You can also use this feature to boot into different VMScluster systems, depending on the adapter you use to boot the system.

To use alternate adapter booting, you need the physical address of the alternate LAN adapter. You use the address to update the satellite's node definition in the DECnet database on the MOP servers so that they recognize the satellite.

VAX

On VAX systems, you can find the LAN address of the additional adapters on VAX systems by using one of the following methods:

- Use the console command SHOW ETHERNET.
- Boot the READ_ADDR program using the following commands:

```
>>>B/100 XQB0
Bootfile:READ_ADDR♦
```

VAX

On AXP systems, you can find the LAN address of the additional adapters by using one of the following console commands:

- SHOW CONFIG
- SHOW NETWORK♦

See Section 7.8.1.4 for information about changing the LAN address in the DECnet database and listing additional LAN addresses in the DECnet database.

Once the MOP load has completed, the boot driver starts the NISCA protocol on the LAN adapter used for booting. The NISCA protocol is used to access the system disk server and to complete the operating system load.

7.8.1.3 Booting from Multiple LAN Adapters (AXP Only)

AXP

System availability can be increased by using multiple LAN adapters for booting. To use multiple adapter booting, you need the physical addresses of the additional LAN adapters. Use this address to update the satellite's node definition in the DECnet database on some of the MOP servers so that they recognize the satellite. Additionally, multiple LAN adapters are specified on the boot command line.

On AXP systems, the following command line is the same as that used for booting from a single LAN adapter (see Section 7.8.1.2) except that it lists two LAN adapters, eza0 and ezb0, as the devices from which to boot:

```
cs>>> b -flags 0,1 eza0, ezb0
```

In this example, MOP booting is attempted from the first device (eza0). If that fails, MOP booting is attempted from the next device (ezb0). When booting from network devices, if the MOP boot attempt fails from all devices, then the console starts again from the first device. Note that you can use the SHOW DEVICE or SHOW CONFIG console command to obtain the names of adapters on AXP computers.

Once the MOP load has completed, the boot driver starts the NISCA protocol on all of the LAN adapters, the NISCA protocol is used to access the system disk server and finish loading the operating system. When booting from multiple LAN adapters, access the MOP server and disk server can occur via different LAN adapters. Note that for predictable operation, each LAN adapter on the boot command line must be downline loaded from the same VMScluster system, and must use the same root of the same system disk.

For AXP systems, you must specify an AXP downline load file in the AXP network database.

Note that the DECnet network database entries (on the MOP servers) associated with all devices on the boot command line must specify the use of the same local root (SYS$SPECIFIC). You can check the network database entries by running NCP and entering the SHOW CHARACTERISTICS NODE *node-name* command. (See Appendix C.)♦

7.8.1.4 Changing the LAN Address in the DECnet Database to Allow a Cluster Satellite to Boot with Any Adapter

DECnet for OpenVMS (DECnet), previously known as DECnet–VAX, implements Phase IV of DNA. DECnet supports one LAN hardware address per node definition. To allow a cluster satellite with multiple LAN adapters to use any LAN adapter to boot into the cluster, use one of the following methods:

- Define a synonym node with a different DECnet address. Have the address point to the same cluster satellite root as the existing node definition. You can display the existing node definition with the NCP command SHOW NODE. Then you can use the NCP commands DEFINE NODE or SET NODE to create a synonym. For the command syntax, refer to *DECnet for OpenVMS Network Management Utilities*.
- Create and maintain different DECnet databases on the different boot nodes within the cluster. In each database, list a different LAN address for the same node definition. A system booting from one LAN adapter receives responses from a subset of the MOP servers. The same system booting from a different LAN adapter receives responses from a different subset of the MOP servers.

Once the satellite receives the MOP downline load from the MOP server, the satellite uses the booting LAN adapter to connect to any node serving the system disk. The satellite continues to use the LAN adapters on the boot command line exclusively until after the run-time drivers are loaded. The satellite then switches to using the run-time drivers and starts the local area VMScluster protocol on all of the LAN adapters.

7.8.1.5 Displaying Connection Messages During Cluster Satellite Booting

In previous versions of the OpenVMS operating system, the system did not display connection messages when it accessed the system disk during local area VMScluster satellite booting. For VAX systems running VMS Version 5.4–3 and later and for OpenVMS AXP systems running Version 1.5, a system displays connection messages during a conversational boot. To enable the display, perform the following steps:

1. Enable conversational booting by updating the satellite's system parameters in the ALPHAVMSSYS.PAR file for AXP systems or the VAXVMSSYS.PAR file for VAX systems in the system root on the disk server.
2. Make sure that the NISCS_CONV_BOOT system parameter is set to 1.
3. Perform a conversational boot.

AXP

On AXP systems, enter the following command on the console:

```
>>> b -flags 0,1♦
```

VAX

On VAX systems, set bit <0> in register R5.♦

Displaying connection messages during a satellite boot allows you to determine which system in a large cluster is serving the system disk to a cluster satellite during the boot process. If booting problems occur, you can use this display to help isolate the problem with the system that is currently serving the system disk. Then, if your server system has multiple LAN adapters, you can isolate specific LAN adapters.

Isolate a LAN adapter by disconnecting all but one of the LAN adapters on the server system and then rebooting the satellite. If the satellite boots when it is connected to the system disk server, then a different LAN adapter is at fault. Use a different LAN adapter on the system to reattempt the satellite boot until the satellite does not boot. When the system does not boot, you have located the bad adapter.

See also Appendix C for help with troubleshooting satellite booting problems.

7.8.1.6 Configuring MOP Service

On a boot node, CLUSTER_CONFIG.COM enables the DECnet MOP downline load service on the first circuit that is found in the DECnet database. The circuit state and the service (MOP downline load service) state can be displayed using the following command:

```
$ MCR NCP SHOW CHAR KNOWN CIRCUITS

          .
          .
          .
Circuit = SVA-0

State                   = on
Service                 = enabled
          .
          .
          .
```

This example shows that circuit SVA-0 is in the ON state with the MOP downline service enabled. This is the correct state to support MOP downline loading for local area VMScluster satellites.

Some boot nodes might perform routing between multiple extended LANs. In these configurations, you might want to enable the MOP downline load service on the other DECnet circuits. This allows VMScluster satellite nodes to be connected to both extended LANs and possibly to boot from either LAN. NCP can be used to enable the MOP downline service for each of the circuits. Note that the circuit must be turned off prior to enabling the MOP downline load service.

The following example enables service for the circuit QNA-1:

```
$ MCR NCP SET CIRCUIT QNA-1 STATE OFF
$ MCR NCP SET CIRCUIT QNA-1 SERVICE ENABLE STATE ON
$ MCR NCP DEFINE CIRCUIT QNA-1 SERVICE ENABLE
```

For further details, refer to *DECnet for OpenVMS Network Management Utilities*.

7.8.2 Configuring Disk Server LAN Adapters and Memory

Because disk-serving activity in a large local area or mixed-interconnect VMScluster system can generate a substantial amount of I/O traffic on the LAN, boot and disk servers should use the highest-bandwidth LAN adapters in the cluster. The servers can also use multiple LAN adapters in a single system to distribute the load across the LAN adapters.

In addition, a large local area or mixed-interconnect cluster should include multiple boot and disk servers to enhance availability and to distribute I/O traffic over several cluster nodes.

Relatively little memory is required to serve disks. Even busy boot and disk servers probably require no more than 0.25 to 0.5 MB of physical memory for disk-serving activity. However, if boot and disk servers must also support timesharing users or run batch queues for the cluster, the servers should be configured with memory appropriate for those additional tasks.

7.8.3 Configuring System Disks

Depending on the number of computers to be included in a large cluster, you must evaluate the trade-offs involved in configuring a single system disk or multiple system disks.

While a single system disk is easier to manage, a large cluster might require more system disk I/O capacity than a single system disk can provide. To achieve satisfactory performance, multiple system disks might be needed. However, you should recognize the increased system management efforts involved in maintaining multiple system disks.

7.8.3.1 Concurrent User Activity

In clusters with many workstation satellites, the amount and type of user activity on those satellites (for example, any active batch job or other task created on the workstation by or for the user) influence system disk load and therefore the number of satellites that can be supported by a single system disk. For example, if many users are active or run multiple applications simultaneously, the load on the system disk can be significant. Conversely, in an environment where few users are active simultaneously, or where most users run a single application for extended periods, a single system disk might support a large number of satellites. Note, however, that in these environments significant numbers of I/O requests can be directed to application data disks.

This situation is similar to the traditional timesharing model, because the probability is low that most users are active simultaneously. Thus, while a VMScluster system can be configured on the assumption that all users are constantly active, a smaller and less expensive one can be configured for more typical working conditions. The trade-off is between a more expensive VMScluster system that handles rare peak loads without performance degradation and a less expensive one that handles most normal activity as well as the more expensive one but that suffers some performance degradation during peak load periods.

Note one difference from the traditional timesharing model: In a timesharing system, the most important shared resource is the processing power of a shared computer. But because each workstation user in a VMScluster system has a dedicated computer, a user who runs large compute-bound jobs on that dedicated computer does not significantly affect users of other computers in the VMScluster system.

For clustered workstations, the critical shared resource is a disk server. Thus, if a workstation user runs even a small I/O-intensive job, its effect on other workstations sharing the same disk server might be noticeable.

7.8.3.2 Concurrent Booting Activity

One of the rare times when all VMScluster computers are simultaneously active is during a cluster reboot—for example, after a power failure. All satellites are waiting to reload the AXP or VAX operating system, and as soon as a boot server is available, they begin to boot in parallel. This booting activity places a significant I/O load on the system disk or disks.

VAX

For example, Table 7–7 shows a VAX system disk's I/O activity and elapsed time until login for a single satellite with minimal startup procedures when the satellite is the only one booting. Table 7–8 shows system disk I/O activity and time elapsed between boot server response and login for various numbers of satellites booting from a single system disk. The disk in these examples has a capacity of 40 I/O operations per second.

Note that the numbers in the tables are fabricated and are meant to provide only a generalized picture of booting activity. Elapsed times until login on satellites in any particular cluster depend on the complexity of the site-specific system startup procedures. Computers in clusters with many layered products or site-specific applications require more system disk I/O operations to complete booting operations.

Table 7–7 System Disk I/O Activity and Boot Time for a Single VAX Satellite

Total I/O Requests to System Disk	Average System Disk I/O Operations per Second	Elapsed Time Until Login (Minutes)
4200	6	12

Table 7–8 System Disk I/O Activity and Boot Times for Multiple VAX Satellites

Number of Satellites	I/Os Requested per Second	I/Os Serviced per Second	Elapsed Time Until Login (Minutes)
1	6	6	12
2	12	12	12
4	24	24	12
6	36	36	12
8	48	40	14
12	72	40	21
16	96	40	28
24	144	40	42
32	192	40	56
48	288	40	84
64	384	40	112
96	576	40	168

While the elapsed times shown in Table 7–8 do not include the time required for the boot server itself to reload, they illustrate that the I/O capacity of a single system disk can be the limiting factor for cluster reboot time.♦

Note that you can reduce overall cluster boot time by configuring multiple system disks and by distributing system roots for computers evenly across those disks. This technique has the advantage of increasing overall system disk I/O capacity but has the disadvantage of requiring additional system management effort. For example, installation of layered products or upgrades of the OpenVMS operating system must be repeated once for each system disk.

In a mixed-interconnect VMScluster system, you can use Volume Shadowing for OpenVMS software to increase the I/O capacity of a single system disk. Installations or updates need only be applied once to a volume-shadowed system disk. For clusters with substantial system disk I/O requirements, you can use multiple system disks, each configured as a shadow set.♦

7.8.3.3 Boot Time Costs

When configuring a VMScluster system for minimum boot times, consider the following:

- Cost of workstations being unavailable during a cluster reboot
- Hardware costs of additional disk drives
- Cost of Volume Shadowing for OpenVMS software, if needed
- System management effort required to maintain multiple system disks
- Probability of power interruptions

Note

Sites with stringent demands for high system availability should investigate power conditioning options to minimize power interruption problems.

7.8.3.4 Moving High-Activity Files off System Disks

To reduce I/O activity on system disks, you can move page and swap files for computers off system disks, and you can set up page and swap files for satellites on the satellites' local disks, if such disks are available. You specify the sizes and locations of page and swap files when you run CLUSTER_CONFIG.COM to add computers.

You should also move off the system disk such high-activity files as the following:

SYSUAF.DAT	NETPROXY.DAT	RIGHTSLIST.DAT
ACCOUNTNG.DAT	VMSMAIL_PROFILE.DATA	QMAN$MASTER.DAT

VAX On VAX systems, you can also move the location of the VMS$OBJECTS.DAT file.♦

To specify the location of the files, follow the instructions in Chapter 4.

7.8.3.5 Controlling Dump File Size and Creation

Whether your VMScluster system uses a single common system disk or multiple system disks, you should plan a strategy to manage dump files. Dump file management is especially important for large clusters with a single system disk. For example, on a 256 MB OpenVMS AXP computer, AUTOGEN creates a dump file in excess of 500,000 blocks.

In the event of a software-detected system failure, each computer normally writes the contents of memory to a full dump file on its system disk for analysis. By default, this full dump file is the size of physical memory plus a small number of pages. If system disk space is limited (as is probably the case if a single system disk is used for a large cluster), you may want to specify that no dump file be created for satellites or that AUTOGEN create a selective dump file. The selective dump file is typically 30% to 60% of the size of a full dump file.

You can control dump file size and creation for each computer by specifying appropriate values for the AUTOGEN symbols DUMPSTYLE and DUMPFILE in the computer's MODPARAMS.DAT file. Specify dump files as shown in Table 7–9.

Table 7–9 AUTOGEN Dump File Symbols

Value Specified	Effect
DUMPSTYLE = 0	Full dump file created (default)
DUMPSTYLE = 1	Selective dump file created
DUMPFILE = 0	No dump file created

Caution

Although you can configure computers without dump files, the lack of a dump file can make it difficult or impossible to determine the cause of a system failure.

For example, use the following commands to modify the system dump file size on large-memory systems:

```
$ MCR SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SET DUMPSTYLE 1
SYSGEN> CREATE SYS$SYSTEM:SYSDUMP.DMP/SIZE=70000
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
$ @SHUTDOWN
```

The dump file size of 70,000 blocks is sufficient to cover about 32 MB of memory. This dump file size is usually large enough to encompass the information needed to analyze a system failure.

After the system reboots, you can purge SYSDUMP.DMP.

7.8.3.6 Sharing Dump Files

Another option for saving dump file space is to share a single dump file among multiple computers. This technique makes it possible to analyze isolated computer failures. But dumps are lost if multiple computers fail at the same time or if a second computer fails before you can analyze the first failure. Because boot server failures have a greater impact on cluster operation than do failures of other computers, you should configure full dump files on boot servers to help ensure speedy analysis of problems.

VAX systems cannot share dump files with AXP computers and vice versa. However, you can share a single dump file among multiple AXP computers, and another single dump file among VAX computers. Follow these steps for each operating system:

1. Decide whether to use full or selective dump files.
2. Determine the size of the largest dump file needed by any satellite.
3. Select a satellite whose memory configuration is the largest of any in the cluster and do the following:
   a. Specify DUMPSTYLE = 0 (or DUMPSTYLE = 1) in that satellite's MODPARAMS.DAT file.
   b. Remove any DUMPFILE symbol from the satellite's MODPARAMS.DAT file.
   c. Run AUTOGEN on that satellite to create a dump file.
4. Rename the dump file to SYS$COMMON:[SYSEXE]SYSDUMP-COMMON.DMP or create a new dump file named SYSDUMP-COMMON.DMP in SYS$COMMON:[SYSEXE].
5. For each satellite that is to share the dump file, do the following:
   a. Create a file synonym entry for the dump file in the system-specific root. For example, to create a synonym for the satellite using root SYS1E, enter a command like the following:

      ```
      $ SET FILE SYS$COMMON:[SYSEXE]SYSDUMP-COMMON.DMP -
      _$ /ENTER=SYS$SYSDEVICE:[SYS1E.SYSEXE]SYSDUMP.DMP
      ```

   b. Add the following lines to the satellite's MODPARAMS.DAT file:

      ```
      DUMPFILE = 0
      DUMPSTYLE = 0 (or DUMPSTYLE = 1)
      ```

6. Rename the old system-specific dump file on each system that has its own dump file:

   ```
   $ RENAME SYS$SYSDEVICE:[SYSn.SYSEXE]SYSDUMP.DMP .OLD
   ```

   The value of *n* in the command line is the root for each system (for example, SYS0 or SYS1). Rename the file so that the operating system software does not use it as the dump file when the system is rebooted.

7. Reboot each node so it can map to the new common dump file. The operating system software cannot use the new file for a crash dump until you reboot the system.

8. After you reboot, delete the SYSDUMP.OLD file in each system-specific root. Do not delete any file called SYSDUMP.DMP; instead, rename it, reboot, and then delete it as described in steps 6 and 7.

7.8.4 Adding Computers to an Existing Cluster

When a computer is first added to a cluster, system parameters that control the computer's system resources are normally adjusted in several steps, as follows:

1. CLUSTER_CONFIG.COM sets initial parameters that are adequate to boot the computer in a minimum environment.
2. When the computer boots, AUTOGEN runs automatically to size the static operating system (without using any dynamic FEEDBACK data), and the computer reboots into the production environment.
3. After the newly added computer has been subjected to typical use for a day or more, you should run AUTOGEN with FEEDBACK manually to adjust parameters for the production environment.
4. At regular intervals, and whenever a major change occurs in the cluster configuration or production environment, you should run AUTOGEN with FEEDBACK manually to readjust parameters for the changes.

Because, however, the first AUTOGEN run (initiated by CLUSTER_CONFIG.COM) is performed both in the minimum environment and without FEEDBACK, a newly added computer may be inadequately configured to run in the production environment of some large clusters. For this reason, you might want to implement additional configuration measures like those described in Section 7.8.4.1 and Section 7.8.4.2.

Note

If you boot nodes into an existing VMScluster using minimum startup (the system parameter STARTUP_P1 is set to MIN), a number of processes (for example, CACHE_SERVER, CLUSTER_SERVER, and CONFIGURE) are not started. Digital recommends that you start these processes manually if you intend to run the VMScluster system for an extended period of time. Extended processing without these processes enabled is not recommended. Refer to the *OpenVMS System Manager's Manual* for more information about starting these processes manually.

See also Section 4.4.1 for more information about building startup procedures.

7.8.4.1 Running AUTOGEN with FEEDBACK for Initial Configuration

To ensure that computers are configured adequately for production use when they first join the cluster, you can run AUTOGEN with FEEDBACK automatically as part of the initial boot sequence. Although this step adds an additional reboot before the computer can be used for production work, the computer's performance can be substantially improved for the first few days of use.

When a computer first boots into a large cluster, much of the computer's resource utilization is determined by the current cluster configuration. Factors such as the number of computers, the number of disk servers, and the number of disks available or mounted contribute to a fixed minimum resource requirement. Because this minimum does not change with continued use of the computer, FEEDBACK information about the required resources is immediately valid.

Other FEEDBACK information, however, such as that influenced by normal user activity, is not immediately available, because the only "user" has been the system startup process. If AUTOGEN were run with FEEDBACK at this point, some system values might be set too low.

By running a simulated user load at the end of the first production boot, you can ensure that AUTOGEN has reasonable FEEDBACK information. The User Environment Test Package (UETP) supplied with your operating system contains a test that simulates such a load. You can run this test (the UETP LOAD phase) as part of the initial production boot, and then run AUTOGEN with FEEDBACK before a user is allowed to log in.

To implement this technique, you can create a command file like that in step 1 of the procedure in Section 7.8.4.2, and submit the file to the computer's local batch queue from the cluster common SYSTARTUP procedure. Your command file conditionally runs the UETP LOAD phase and then reboots the computer with AUTOGEN FEEDBACK.

7.8.4.2 Creating a Command File to Run AUTOGEN with FEEDBACK

As shown in the following sample file, UETP lets you specify a typical user load to be run on the computer when it first joins the cluster. The UETP run generates data that AUTOGEN uses to set appropriate system parameter values for the computer when rebooting it with FEEDBACK. Note, however, that the default setting for the UETP user load assumes that the computer is used as a timesharing system. This calculation can produce system parameter values that might be excessive for a single-user workstation, especially if the workstation has large memory resources. Therefore, you might want to modify the default user load setting, as shown in the sample file.

Follow these steps:

1. Create a command file like the following:

```
$!
$!   ***** SYS$COMMON:[SYSMGR]UETP_AUTOGEN.COM *****
$!
$! For initial boot only, run UETP LOAD phase and
$! reboot with AUTOGEN FEEDBACK.
$!
$ SET NOON
$ SET PROCESS/PRIVILEGES=ALL
$!
$! Run UETP to simulate a user load for a satellite
$! with 8 simultaneously active user processes. For a
$! CI connected computer, allow UETP to calculate the load.
$!
$ LOADS = "8"
$ IF F$GETDVI("PAA0:","EXISTS") THEN LOADS = ""
$ @UETP LOAD 1 'loads'
$!
$! Create a marker file to prevent resubmission of
$! UETP_AUTOGEN.COM at subsequent reboots.
$!
$ CREATE SYS$SPECIFIC:[SYSMGR]UETP_AUTOGEN.DONE
$!
$! Reboot with AUTOGEN to set SYSGEN values.
$!
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT FEEDBACK
$!
$ EXIT
```

2. Edit the cluster common SYSTARTUP file and add commands like the following at the end of the file. Assume that queues have been started and that a batch queue is running on the newly added computer. Submit UETP_AUTOGEN.COM to the computer's local batch queue.

```
$!
$ NODE = F$GETSYI("NODE")
$ IF F$SEARCH ("SYS$SPECIFIC:[SYSMGR]UETP_AUTOGEN.DONE") .EQS. ""
$ THEN
$  SUBMIT /NOPRINT /NOTIFY /USERNAME=SYSTEST -
   /QUEUE='NODE'_BATCH SYS$MANAGER:UETP_AUTOGEN

$ WAIT_FOR_UETP:
$  WRITE SYS$OUTPUT "Waiting for UETP and AUTOGEN... ''F$TIME()'"
$  WAIT 00:05:00.00            ! Wait 5 minutes
$  GOTO WAIT_FOR_UETP
$ ENDIF
$!
```

   Note that UETP must be run under the user name SYSTEST.

3. Execute CLUSTER_CONFIG.COM to add the computer.

When you boot the computer, it runs UETP_AUTOGEN.COM to simulate the user load you have specified, and it then reboots with AUTOGEN FEEDBACK to set appropriate system parameter values.

7.8.5 Setting Up a New, Large VMScluster System

When building a new large cluster, you must be prepared to run AUTOGEN and reboot the cluster several times during the installation. The parameters that AUTOGEN sets for the first computers added to the cluster will probably be inadequate when additional computers are added. Readjustment of parameters is critical for boot and disk servers.

One solution to this potential problem is to run UETP_AUTOGEN.COM to reboot computers at regular intervals as new computers are added. You should run the procedure according to the percentage of growth. For example, each time there is a significant percentage increase in the number of computers (from 5% to 10%, from 10% to 20%, and so forth), you should run UETP_AUTOGEN.COM. For best results, the cluster environment should be as close as possible to the final production environment when you run the procedure.

To set up the cluster, you can follow these steps:

1. Configure boot and disk servers using the CLUSTER_CONFIG.COM command procedure.
2. Install all layered products and site-specific applications required for the cluster production environment, or as many as possible.
3. Prepare the cluster startup procedures so that they are as close as possible to those that will be used in the final production environment.
4. Add a small number of satellites (perhaps two or three) using CLUSTER_CONFIG.COM.
5. Reboot the cluster to verify that the startup procedures work as expected.
6. After you have verified that startup procedures work, run UETP_AUTOGEN.COM on every computer's local batch queue to reboot the cluster again and to set initial production environment values. When the cluster has rebooted, all computers should have reasonable parameter settings. However, check the settings to be sure.
7. Add additional satellites to double their number, and then rerun UETP_AUTOGEN on each computer's local batch queue to reboot the cluster, and set values appropriately to accommodate the newly added satellites.
8. Repeat the previous step until all satellites have been added.
9. When all satellites have been added, run UETP_AUTOGEN a final time on each computer's local batch queue to reboot the cluster and to set new values for the production environment.

Note that, for best performance, you might not want to run UETP_AUTOGEN on every computer simultaneously, because the procedure simulates a user load that is probably more demanding than that for the final production environment. A better method is to run UETP_AUTOGEN on several satellites (those with the least recently adjusted parameters) while adding new computers. This technique increases efficiency because little is gained when a satellite reruns AUTOGEN shortly after joining the cluster. For example, if the entire cluster is rebooted after 30 satellites have been added, few adjustments are made to system parameter values for the 28th satellite added, because only two satellites have joined the cluster since that satellite ran UETP_AUTOGEN as part of its initial configuration.

7.8.6 Defining the VMScluster Alias

The VMScluster alias acts as a single network node identifier for a VMScluster system. Computers in the cluster can use the alias for communications with other computers in a DECnet network. A maximum of 64 VMScluster computers can participate in a VMScluster alias. If your cluster includes more than 64 computers, you must determine which 64 should participate in the alias and then define the alias on those computers. For detailed information about the VMScluster alias, refer to the *DECnet for OpenVMS Networking Manual.*

A

Cluster System Parameters

For systems to boot properly into a cluster, certain system parameters must be set on each cluster computer. Table A–1 lists SYSGEN parameters used in cluster configurations.

Note

Some system parameters for AXP computers are in units of pagelets, whereas others are in pages. AUTOGEN determines the hardware page size and records it in the PARAMS.DAT file. When reviewing AUTOGEN recommended values or when setting system parameters with SYSGEN, note carefully which units are required for each parameter. See also *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for more information about system parameter values on AXP and VAX systems.

Table A–1 Cluster SYSGEN Parameters

Parameter	Description
ALLOCLASS	Specifies a numeric value from 0 to 255 to be assigned as the disk allocation class for the computer. The default value is 0.
DISK_QUORUM	The physical device name, in ASCII, of an optional quorum disk. ASCII spaces indicate that no quorum disk is being used. DISK_QUORUM must be defined on one or more cluster computers capable of having a direct (non MSCP served) connection to the disk. These computers are called **quorum disk watchers**. The remaining computers (computers with a blank value for DISK_QUORUM) recognize the name defined by the first watcher computer with which they communicate.
EXPECTED_VOTES	Specifies a setting that is used to derive the initial quorum value. This setting is the sum of all VOTES held by potential cluster members. By default, the value is 1. The connection manager sets a quorum value to a number that will prevent cluster partitioning (see Section 3.1). To calculate quorum, the system uses the following formula: `estimated quorum = (EXPECTED_VOTES + 2)/2`
LOCKDIRWT	(Lock manager directory system weight). Determines the portion of lock manager directory to be handled by this system. The default value is adequate for most systems.

(continued on next page)

Table A–1 (Cont.) Cluster SYSGEN Parameters

Parameter	Description
†LRPSIZE	For VAX computers running VMS Version 5.5–2 and earlier, the LRPSIZE parameter specifies the size, in bytes, of the large request packets. The actual physical memory consumed by a large request packet is LRPSIZE plus overhead for buffer management. Normally, the default value is adequate. For VAX computers, the value of LRPSIZE affects the transfer size used by VAX nodes on an FDDI ring (see Section 2.9.5).
MSCP_LOAD	Controls whether the MSCP server is loaded. Specify 1 to load the server, and use the default CPU load rating. A value greater than 1 loads the server and uses this value as a constant load rating. By default, the value is set to 0 and the server is not loaded.
MSCP_SERVE_ALL	Specifies MSCP disk-serving functions when the MSCP server is loaded. The default value of 0 specifies that no disks are served. A value of 1 specifies that all available disks are served. A value of 2 specifies that only locally connected (not HSC) disks are served.
NISCS_CONV_BOOT	During booting as a VMScluster satellite, specifies whether conversational bootstraps are enabled on the computer. The default value of 0 specifies that conversational bootstraps are disabled. A value of 1 enables conversational bootstraps.
NISCS_LAN_OVRHD	Specifies the amount of overhead in the LAN packet to be reserved for use by other LAN protocols. Set this parameter to 18 when DESNCs are in the network configuration. This setting prevents packet fragmentation and slow cluster performance when using DESNCs. Set this parameter to 0 when encapsulating devices (such as DESNCs) are not in the network configuration. This setting reduces the network overhead on long transfers and thus improves cluster performance.
NISCS_LOAD_PEA0	Specifies whether the port driver (PEDRIVER) is to be loaded to enable cluster communications over the local area network (LAN). The default value of 0 specifies that the driver is not loaded. A value of 1 specifies that that driver is loaded. **Caution:** If the NISCS_LOAD_PEA0 parameter is set to 1, the VAXCLUSTER system parameter must be set to 2. This ensures coordinated access to shared resources in the VMScluster and prevents accidental data corruption.
NISCS_MAX_PKTSZ	Specifies the maximum packet size (client data) used by PEDRIVER in the range of 1498 to 4468. This parameter allows the customer to increase the packet size for use on FDDI-to-FDDI communications paths. Note that PEDRIVER always allocates memory to support this packet size, so a value larger than 1498 when only using Ethernet communications paths will cause excessive usage of nonpaged pool.
NISCS_PORT_SERV	Specifies whether data checking is enabled for the computer. The default value of 0 specifies that data checking is disabled. The bit setting for the parameter value is: • Bit <0>—When set, enables data checking on transmit • Bit <1>—When set, enables data checking on receive • Bits <31:2>—Reserved; must be 0
QDSKVOTES	Specifies the number of votes contributed to the cluster votes total by a quorum disk. The maximum is 127, the minimum is 0, and the default is 1. This parameter is used only when DISK_QUORUM is defined.

†VAX specific

(continued on next page)

Table A–1 (Cont.) Cluster SYSGEN Parameters

Parameter	Description
QDSKINTERVAL	Specifies, in seconds, the disk quorum polling interval. The maximum is 32767, the minimum is 1, and the default is 10. Lower values trade increased overhead cost for greater responsiveness. Digital recommends that this parameter be set to the same value on each cluster computer.
RECNXINTERVAL	Specifies, in seconds, the interval during which the connection manager attempts to reconnect a broken connection to another computer. If a new connection cannot be established during this period, the connection is declared irrevocably broken, and either this computer or the other must leave the cluster. This parameter trades faster response to certain types of system failures for the ability to survive transient faults of increasing duration. Digital recommends that you set this parameter to the same value on each cluster computer. This parameter also affects the tolerance of the VMScluster system for LAN bridge failures (see Section 2.9.4).
TAPE_ALLOCLASS	Specifies a numeric value from 0 to 255 to be assigned as the tape allocation class for tape devices connected to the computer. The default value is 0.
TIMVCFAIL	Specifies the time required for an adapter or virtual circuit failure to be detected. Digital recommends that you use the default value. Digital also recommends that you decrease this value only in VMScluster systems of three or fewer CPUs, that you use the same value on each computer in the cluster, and that you use dedicated LAN segments for cluster I/O.
TMSCP_LOAD	Controls whether the TMSCP server is loaded. Specify a value of 1 to load the server and set all available TMSCP tapes served. By default, the value is set to 0, and the server is not loaded.
VAXCLUSTER	Controls whether the computer should join or form a cluster and controls coordinated access to shared resources. This parameter accepts the following three values: • 0—Specifies that the computer will not participate in a cluster • 1—Specifies that the computer should participate in a cluster if hardware supporting SCS is present (CI, Ethernet, DSSI, or FDDI) • 2—Specifies that the computer should participate in a cluster You should always set this parameter to 2 on computers intended to run in a cluster, to 0 on computers that boot from a UDA disk controller and are not intended to be part of a cluster, and to 1 (the default) otherwise. **Caution:** If the NISCS_LOAD_PEA0 system parameter is set to 1, the VAXCLUSTER parameter must be set to 2. This ensures coordinated access to shared resources in the VMScluster system and prevents accidental data corruption. Data corruption is likely to occur on shared resources if the NISCS_LOAD_PEA0 parameter is set to 1 and the VAXCLUSTER parameter is set to 0.
VOTES	Specifies the number of votes toward a quorum to be contributed by the computer. The default is 1.
SCS Parameters	
PANUMPOLL	Specifies the number of ports to poll at each interval. Digital recommends that you set this parameter to the same value on each cluster computer.

(continued on next page)

Table A–1 (Cont.) Cluster SYSGEN Parameters

Parameter	Description
PASTIMOUT	Specifies the interval at which the CI port driver performs time-based bookkeeping operations. This interval is also the period after which a start handshake datagram is assumed to have timed out. The default value is adequate on most systems. Digital recommends that you set this parameter to the same value on each VMScluster computer.
PASTDGBUF	Specifies the number of datagram receive buffers to queue for the CI port driver's configuration poller, that is, the maximum number of start handshakes that can be in progress simultaneously. The default value is adequate on most systems. Digital recommends that you set this parameter to the same value on each VMScluster computer.
PAMAXPORT	Specifies the maximum number of CI ports the CI port driver polls for a broken port-to-port virtual circuit or for a failed remote computer. You can decrease this parameter in order to reduce polling activity if the hardware configuration has fewer than 16 ports. For example, if the configuration has a total of five ports that are assigned port numbers 0 through 4, then you should set PAMAXPORT to 4. The default for this parameter is 15 (poll for all possible ports 0 through 15). Digital recommends that you set this parameter to the same value on each cluster computer.
PANOPOLL	Disables CI polling for ports if set to 1. (The default is 0.) When PANOPOLL is set, a computer will not discover that another computer has shut down or powered down promptly and will not discover a new computer that has booted. This parameter is useful when you want to bring up a computer detached from the rest of the cluster for checkout purposes. It is equivalent to uncabling the computer from the star coupler. PANOPOLL = 0 is the normal setting and is required if you are booting from an HSC controller.
PAPOLLINTERVAL	Specifies, in seconds, the polling interval the CI port driver uses to poll for a newly booted computer, a broken port-to-port virtual circuit, or a failed remote computer. This parameter trades polling overhead against quick response to virtual circuit failures. Digital recommends that you use the default value for this parameter. Digital recommends that you set this parameter to the same value on each cluster computer.
PAPOOLINTERVAL	Specifies, in seconds, the interval at which the PA port driver checks for available nonpaged pool after a failure to allocate. The default value is adequate on most systems.
PASANITY	Controls whether the port sanity timer is enabled to permit remote computers to detect a computer that has been halted or retained at IPL 7 for a prolonged period. This parameter is normally set to 1 and should be set to 0 only when debugging with XDELTA. PASANITY is a dynamic parameter (altered the next time the port is initialized) and has a default value of 1.

(continued on next page)

Table A–1 (Cont.) Cluster SYSGEN Parameters

Parameter	Description
PRCPOLINTERVAL	Specifies, in seconds, the polling interval used to look for SCS applications, such as the connection manager and MSCP disks, on other computers. Each computer is polled, at most, once each interval. This parameter trades polling overhead against quick recognition of new computers or servers as they appear. Digital recommends that you set this parameter to 15, which is the default.
SCSBUFFCNT	Specifies the number of CI buffer descriptors configured for all CI ports on the computer. These buffer descriptors are also allocated and used by some Ethernet controllers.
SCSCONNCNT	Specifies the total number of SCS connections that are configured for use by all system applications. The default value is adequate on most systems.
SCSMAXMSG	Specifies the SCS maximum sequenced message size. The default value is adequate on most systems.
SCSMAXDG	Specifies the maximum number of bytes of application data in one datagram. The default value is adequate on most systems.
SCSFLOWCUSH	Specifies the lower limit for receive buffers at which point SCS starts to notify the remote SCS of new receive buffers. For each connection, SCS tracks the number of receive buffers available. SCS communicates this number to the SCS at the remote end of the connection. However, SCS does not need to do this for each new receive buffer added. Instead, SCS notifies the remote SCS of new receive buffers if the number of receive buffers falls as low as the SCSFLOWCUSH value. The default value is adequate on most systems.

SCSSYSTEMID[1]	Specifies a number that identifies the computer. SCSSYSTEMID is the low-order 32 bits of the 48-bit system identification number. If the computer is running DECnet, calculate SCSSYSTEMID using this formula:
	`(1024 * a) + n`
	The variables equate to the following values:
	• The variable *a* is the DECnet area.
	• The variable *n* is the DECnet node number within the area.
	For example, if the DECnet address is 2.211, calculate SCSSYSTEMID as follows:
	`SCSSYSTEMID = (1024 * 2) + 211`
	If the computer is not running DECnet and it is in a VMScluster system, SCSSYSTEMID must be unique within the VMScluster.

SCSSYSTEMIDH	Specifies the high-order 16 bits of the 48-bit system identification number. This parameter must be set to 0. It is reserved by Digital for future use.
SCSNODE[1]	Specifies the name of the computer. This parameter is not dynamic. If the computer is running DECnet, SCSNODE must specify the DECnet computer name (limited to 6 characters). If the computer is not running DECnet and it is in a VMScluster, SCSNODE must specify a string of up to 8 characters that is unique name in the cluster.

[1]Once a computer has been recognized by another computer in the cluster, you cannot change the SCSSYSTEMID or SCSNODE parameter without changing both.

(continued on next page)

Table A–1 (Cont.) Cluster SYSGEN Parameters

Parameter	Description
SCSRESPCNT	Specifies the total number of response descriptor table entries configured for use by all system applications.

System parameters, including cluster and volume shadowing system parameters, are specified in the *OpenVMS System Management Utilities Reference Manual.*

B

Building a Common SYSUAF.DAT File

This appendix provides guidelines for building a common user authorization file from computer-specific files. For more detailed information about how to set up a computer-specific authorization file, see the descriptions in the *OpenVMS AXP Guide to System Security* and the *OpenVMS VAX Guide to System Security*, as appropriate.

To build a common SYSUAF.DAT file, follow these steps:

1. Print a listing of SYSUAF.DAT on each computer. To print this listing, invoke AUTHORIZE and specify the AUTHORIZE command LIST as follows:

   ```
   $ SET DEF SYS$SYSTEM
   $ RUN AUTHORIZE
   UAF> LIST/FULL [*,*]
   ```

2. Use the listings to compare the accounts from each computer. On the listings, mark down any necessary changes.

   One such change is to delete any accounts that you no longer need. You should also make sure that each user account in the cluster has a unique UIC.

   For example, VMScluster member VENUS may have a user account JONES that has the same UIC as user account SMITH on computer MARS. When computers VENUS and MARS are joined to form a cluster, accounts JONES and SMITH will exist in the cluster environment with the same UIC. If the UICs of these accounts are not differentiated, each user will have the same access rights to various objects in the cluster. In this case, you should assign each account a unique UIC.

   Make sure that accounts that perform the same type of work have the same group UIC. Accounts in a single-computer environment probably follow this convention. However, there may be groups of users on each computer that will perform the same work in the cluster but that have group UICs unique to their local computer. As a rule, the group UIC for any given work category should be the same on each computer in the cluster. For example, data entry accounts on VENUS should have the same group UIC as data entry accounts on MARS.

   Note that if you change the UIC for a particular user, you should also change the owner UICs for that user's existing files and directories. You can use the DCL commands SET FILE and SET DIRECTORY to make these changes. These commands are described in detail in the *OpenVMS DCL Dictionary*.

3. Choose the SYSUAF.DAT file from one of the computers to be a master SYSUAF.DAT. Note that the default values for a number SYSUAF process limits and quotas are higher on an AXP computer than they are on a VAX computer. See *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for information about setting values on both computers.

4. Merge the SYSUAF.DAT files from the other computers to the master SYSUAF.DAT by running the Convert utility (CONVERT) on the computer that owns the master SYSUAF.DAT. (See the *OpenVMS Record Management Utilities Reference Manual* for a description of CONVERT.) To use CONVERT to merge the files, each SYSUAF.DAT file must be accessible to the computer that is running CONVERT.

   To merge the UAFs into the master SYSUAF.DAT file, specify the CONVERT command in the following format:

   CONVERT SYSUAF1,SYSUAF2,...SYSUAFn MASTER_SYSUAF

   Note that if a given user name appears in more than one source file, only the first occurrence of that name appears in the merged file.

   The command sequence in the following example adds the SYSUAF.DAT file from two VMScluster computers to the master SYSUAF.DAT in the current default directory:

   ```
   $ SET DEFAULT SYS$SYSTEM
   $ CONVERT [SYS1.SYSEXE]SYSUAF.DAT, [SYS2.SYSEXE]SYSUAF.DAT SYSUAF.DAT
   ```

   The CONVERT command in this example adds the records from the files [SYS1.SYSEXE]SYSUAF.DAT and [SYS2.SYSEXE]SYSUAF.DAT to the file SYSUAF.DAT on the local computer.

   After you run CONVERT, you have a master SYSUAF.DAT that contains records from the other SYSUAF.DAT files.

5. Use AUTHORIZE to modify the accounts in the master SYSUAF.DAT according to the changes you marked on the initial listings of the SYSUAF.DAT files from each computer.

6. Place the master SYSUAF.DAT file in SYS$COMMON:[SYSEXE].

7. Remove all node-specific SYSUAF.DAT files.

Merging RIGHTSLIST.DAT Files

If you need to merge RIGHTSLIST.DAT files, you can use a command sequence like the following:

```
$ SET DEFAULT SYS$SYSTEM
$ ANALYZE/RMS/FDL RIGHTSLIST.DAT
$ CONVERT/STATISTICS/FDL=RIGHTSLIST -
_$ [SYS1.SYSEXE]RIGHTSLIST.DAT, [SYS2.SYSEXE]RIGHTSLIST.DAT RIGHTSLIST.DAT
```

The commands in this example add the RIGHTSLIST.DAT files from two VMScluster computers to the master RIGHTSLIST.DAT file in the current default directory. For detailed information about creating and maintaining RIGHTSLIST.DAT files, see the security guide for your system.

C

Cluster Troubleshooting

This appendix contains information to help you perform troubleshooting operations for the following:

- Failures of computers to boot or to join the cluster
- Cluster hangs
- CLUEXIT bugchecks
- Port device problems

C.1 Diagnosing Failures of Computers to Boot or Join the Cluster

Before you initiate diagnostic procedures, be sure to verify that these conditions are met:

- All cluster hardware components are correctly connected and checked for proper operation.
- VMScluster computers and mass storage devices are configured according to requirements specified in both the VAXcluster Software for OpenVMS VAX *Software Product Description* (SPD 29.78.*xx*) and the VMScluster Software for OpenVMS AXP *Software Product Description* (SPD 42.18.*xx*).

When you attempt to add a new or recently repaired CI computer to the cluster, verify that the CI cables are correctly connected, as described in Section C.4.2.2.

When attempting to add a satellite to a local area or mixed-interconnect cluster, you must verify that the LAN is configured according to requirements specified in the VAXcluster SPD or the VMScluster SPD, and that the machine's memory resources and LAN adapters meet the requirements specified in that document. You must also verify that you have correctly configured and started the DECnet for OpenVMS (DECnet) network, following the procedures described in Section 4.3.

If after performing preliminary checks and taking appropriate corrective action, you find that a computer still fails to boot or to join the cluster, you can follow the procedures in Sections C.1.2 through C.1.4 to attempt recovery.

C.1.1 Events for Computers Booting and Joining the Cluster

To perform diagnostic and recovery procedures effectively, you must understand the events that occur when a computer boots and attempts to join the cluster. This section outlines those events and shows typical messages displayed at the console.

Note that events vary, depending on whether a computer is the first to boot in a new cluster or whether it is booting in an active cluster. Note also that some events (such as loading the cluster security database) occur only in local area or mixed-interconnect clusters.

The normal sequence of events is as follows:

1. 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 VMScluster communications.

   ```
   %VAXcluster-I-SYSLOAD, system loaded from Node X... (XX-XX-XX-XX-XX-XX)
   ```

   For any booting computer, the OpenVMS "banner message" is displayed in the following format:

   operating-system Version *n.n dd-mmm-yyyy hh:mm.ss*

2. The computer attempts to form or join the cluster, and the following message appears:

   ```
   waiting to form or join a VMScluster 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 can be loaded:

   ```
   %VAXcluster-I-LOADSECDB, loading the cluster security database
   %MSCPLOAD-I-LOADMSCP, loading the MSCP disk server
   %TMSCPLOAD-I-LOADTMSCP, loading the TMSCP tape server
   ```

3. 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 VAXcluster membership request to system X...
   ```

   Otherwise, the connection manager forms the cluster when it has enough votes to establish quorum (that is, when enough voting computers have booted).

4. As the booting computer joins the cluster, the connection manager displays a message in the following format:

   ```
   %CNXMAN, now a VAXcluster 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 messages 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 messages 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.

5. Once the computer has joined the cluster, normal startup procedures execute. One of the first functions is to start the OPCOM process:

```
%%%%%%%%%%  OPCOM  15-APR-1990 16:33:55.33  %%%%%%%%%%
Logfile has been initialized by operator _X...$OPA0:
Logfile is SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;17

%%%%%%%%%%  OPCOM  15-APR-1990 16:33:56.43  %%%%%%%%%%
16:32:32.93  Node X... (csid 0002000E) is now a VAXcluster member
```

When other computers join the cluster, OPCOM displays messages like the following:

```
%%%%%%%%%%  OPCOM  15-APR-1990 16:34:25.23  %%%%%%%%%%  (from node X... at 16:34:25.23)
16:34:24.42 Node X... (csid 000100F3) received VAXcluster membership request from node X...
```

As startup procedures continue, various messages report startup events.

Note

For troubleshooting purposes, you can include in your site-specific startup procedures messages announcing each phase of the startup process—for example, mounting disks or starting queues.

C.1.2 CI Computer Fails to Boot

If a CI computer fails to boot, perform the following checks:

- Verify that the computer's SCSNODE and SCSSYSTEMID parameters are unique in the cluster. If they are not, you must either alter *both* values or reboot all other computers.
- Verify that you are using the correct bootstrap command file. This file must specify the internal bus computer number (if applicable), the HSC node number, and the HSC disk from which the computer is to boot. Refer to your processor-specific installation and operations guide for information about setting values in default bootstrap command procedures.
- Verify that the PAMAXPORT system parameter is set to a value greater than or equal to the largest CI port number.
- Verify that the CI port has a unique hardware station address.
- Verify that the HSC subsystem is on line. The ONLINE switch on the HSC operator control panel should be pressed in.
- Verify that the disk is available. The correct port switches on the disk's operator control panel should be pressed in.

- Verify that the computer has access to the HSC subsystem. The SHOW HOSTS command of the HSC SETSHO utility displays status for all computers (hosts) in the cluster. (For complete information about the SETSHO utility, consult the HSC hardware documentation.) If the computer in question appears in the display as DISABLED, use the SETSHO utility to set the computer to the ENABLED state.
- Verify that the HSC subsystem allows access to the boot disk. Invoke the SETSHO utility to ensure that the boot disk is available to the HSC subsystem. The utility's SHOW DISKS command displays the current state of all disks visible to the HSC subsystem and displays all disks in the no-host-access table. If the boot disk appears in the no-host-access table, use the SETSHO utility to set the boot disk to host-access. If the boot disk is available or mounted and host access is enabled, but the disk does not appear in the no-host-access table, contact your Digital Services representative and explain both the problem and the steps you have taken.

C.1.3 Satellite Fails to Boot

To boot successfully, a satellite must communicate with a MOP server over the LAN. You can use DECnet event logging to verify this communication. Proceed as follows:

1. Log in as system manager on the MOP server.
2. If event logging for management layer events is not already enabled, enter the following NCP commands to enable it:

```
NCP> SET LOGGING MONITOR EVENT 0.*
NCP> SET LOGGING MONITOR STATE ON
```

3. Enter the following DCL command:

```
$ REPLY/ENABLE=NETWORK
```

   This command enables the terminal to receive DECnet messages reporting downline load events.

4. Boot the satellite. If the satellite and the MOP server can communicate and all boot parameters are correctly set, messages like the following are displayed at the MOP server's terminal:

```
DECnet event 0.3, automatic line service
From node 2.4 (URANUS), 15-APR-1990 09:42:15.12
Circuit QNA-0, Load, Requested, Node = 2.42 (OBERON)
File = SYS$SYSDEVICE:<SYS10.>, Operating system
Ethernet address = 08-00-2B-07-AC-03

DECnet event 0.3, automatic line service
From node 2.4 (URANUS), 15-APR-1990 09:42:16.76
Circuit QNA-0, Load, Successful, Node = 2.42 (ARIEL)
File = SYS$SYSDEVICE:<SYS11.>, Operating system
Ethernet address = 08-00-2B-07-AC-13
```

   If the satellite cannot communicate with the MOP server (VAX or AXP), no message for that satellite appears. There may be a problem with a LAN cable connection or adapter service.

   If the satellite's data in the DECnet database is incorrectly specified (for example, if the hardware address is incorrect), a message like the following displays the correct address and indicates that a load was requested:

```
DECnet event 0.7, aborted service request
From node 2.4 (URANUS), 15-APR-1990 09:42:09.67
Circuit QNA-0, Line open error, Ethernet address = 08-00-2B-03-29-99
```

Note the absence of the node name, node address, and system root.

Sections C.1.3.1 through Section C.1.3.4 provide more information about satellite boot troubleshooting, and often recommend that you ensure that the system parameters are set correctly.

C.1.3.1 General VMScluster Satellite Boot Troubleshooting

If a satellite fails to boot, use the steps outlined in this section to diagnose and correct problems in VMScluster systems.

1. Verify that the boot device is available. This check is particularly important for clusters in which satellites boot from multiple system disks.
2. Verify that the DECnet for OpenVMS network is up and running.
3. Check the cluster group code and password. The cluster group code and password are set using the CLUSTER_CONFIG.COM procedure.
4. Verify that you have installed the correct OpenVMS AXP and OpenVMS VAX licenses.
5. Verify system parameter values on each satellite node, as follows:

   VAXCLUSTER = 2
   NISCS_LOAD_PEA0 = 1
   NISCS_LAN_OVRHD = 0
   NISCS_MAX_PKTSZ = 1498
   PE3 = 0
   PE4 = 0
   SCSNODE is the name of the computer.
   SCSSYSTEMID is a number that identifies the computer.

   The SCS parameter values are set differently depending on your system configuration. Appendix A describes how to set these SCS parameters.

To check system parameter values on a satellite node that cannot boot, invoke the SYSGEN utility on a running system in the VMScluster that has access to the satellite node's local root. Note that you must invoke the SYSGEN utility from a node that is running the same type of operating system (for example, to troubleshoot an AXP satellite node, you must run the SYSGEN utility on an AXP system). Check system parameters as follows:

1. Enter the following command to invoke NCP and find the local root of the satellite node on the system disk. The following example is run on an AXP system:

```
$ MCR NCP SHOW NODE HOME CHARACTERISTICS

Node Volatile Characteristics as of 10-MAY-1993 09:32:56

Remote node =   63.333 (HOME)

Hardware address          = 08-00-2B-30-96-86
Load file                 = APB.EXE
Load Assist Agent         = SYS$SHARE:NISCS_LAA.EXE
Load Assist Parameter     = ALPHA$SYSD:[SYS17.]
```

   The local root in this example is ALPHA$SYSD:[SYS17.].

2. Enter the SHOW LOGICAL command at the system prompt to translate the logical name for ALPHA$SYSD.

```
$ SHO LOG ALPHA$SYSD

   "ALPHA$SYSD" = "$69$DUA121:" (LNM$SYSTEM_TABLE)
$
```

3. Invoke the SYSGEN utility on the system from which you can access the satellite's local disk. (This example invokes the SYSGEN utility on an AXP system using the AXP parameter file ALPHAVMSSYS.PAR. The SYSGEN utility on VAX systems differs in that it uses the VAX parameter file VAXVMSSYS.PAR). The following example illustrates how to enter the SYSGEN command USE with the system parameter file on the local root for the satellite node and then enter the SHOW command to query the parameters in question.

```
$  MCR SYSGEN

sysgen>  USE $69$DUA121:[SYS17.SYSEXE]ALPHAVMSSYS.PAR
sysgen>  SHOW VOTES
Parameter Name     Current    Default     Min.     Max.     Unit  Dynamic
--------------     -------    -------    -------  -------   ----  -------
VOTES                    0          1          0      127 Votes
SYSGEN>  EXIT
$
```

C.1.3.2 Troubleshooting MOP Servers

To diagnose and correct problems for MOP servers, follow the steps outlined in this section.

1. Perform the steps outlined in Section C.1.3.1.
2. Verify the NCP circuit state is on and the service is enabled. Enter the following commands to run the NCP utility and check the NCP circuit state.

```
$ MCR NCP
NCP> SHOW CIRCUIT ISA-0 CHARACTERISTICS

Circuit Volatile Characteristics as of 12-DEC-1992 10:08:30

Circuit = ISA-0

State                    = on
Service                  = enabled
Designated router        = 63.1021
Cost                     = 10
Maximum routers allowed  = 33
Router priority          = 64
Hello timer              = 15
Type                     = Ethernet
Adjacent node            = 63.1021
Listen timer             = 45
```

If service is not enabled, you can enter NCP commands like the following to enable it:

```
NCP> SET CIRCUIT circuit-id STATE OFF
NCP> DEFINE CIRCUIT circuit-id SERVICE ENABLED
NCP> SET CIRCUIT circuit-id SERVICE ENABLED STATE ON
```

The DEFINE command updates the permanent database and ensures that service is enabled the next time you start the network. Note that DECnet traffic is interrupted while the circuit is off.

3. Verify that the load assist parameter points to the system disk and the system root for the satellite.
4. Verify that the satellite's system disk is mounted on the MOP server node.
5. For AXP systems, verify that the load file is APB.EXE.
6. For MOP booting, the satellite node's parameter file (ALPHAVMSYS.PAR for AXP computers and VAXVMSSYS.PAR for VAX computers) must be located in the [SYSEXE] directory of the satellite system root.
7. Ensure that the file CLUSTER_AUTHORIZE.DAT is located in the [SYSCOMMON.SYSEXE] directory of the satellite system root.

C.1.3.3 Troubleshooting Disk Servers

To diagnose and correct problems for disk servers, follow the steps outlined in this section.

1. Perform the steps in Section C.1.3.1.
2. For each satellite node, verify the following system parameter values:

   MSCP_LOAD = 1
   MSCP_SERVE_ALL = 1

3. The disk servers for the system disk must be connected directly to the disk.

C.1.3.4 Troubleshooting Satellite Booting

To diagnose and correct problems for satellite booting, follow the steps outlined in this section.

1. Perform the steps in Sections C.1.3.1, C.1.3.2, and C.1.3.3.
2. For each satellite node, verify that the VOTES system parameter is set to 0.
3. Verify the DECnet network database on the MOP servers by running the NCP utility and entering the following commands to display node characteristics:

   AXP

   The following example displays information about an AXP node named UTAH:

```
$ MCR NCP
NCP> SHOW NODE UTAH CHARACTERISTICS

Node Volatile Characteristics as of 12-MAY-1993 10:28:09

Remote node =   63.227 (UTAH)

Hardware address          = 08-00-2B-2C-CE-E3
Load file                 = APB.EXE
Load Assist Agent         = SYS$SHARE:NISCS_LAA.EXE
Load Assist Parameter     = $69$DUA100:[SYS17.]
```

   On AXP systems, the load file must be APB.EXE. In addition, when booting AXP nodes, for each LAN adapter specified on the boot command line, the load assist parameter must point to the same system disk and root number.♦

VAX

The following example displays information about a VAX node named ARIEL:

```
$ MCR NCP
NCP> SHOW CHAR NODE ARIEL

Node Volatile Characteristics as of 15-APR-1990 13:15:28

Remote node =    2.41 (ARIEL)

Hardware address          = 08-00-2B-03-27-95
Tertiary loader           = SYS$SYSTEM:TERTIARY_VMB.EXE
Load Assist Agent         = SYS$SHARE:NISCS_LAA.EXE
Load Assist Parameter     = DISK$VAXVMSRL5:<SYS12.>
```

Note that on VAX nodes, the tertiary loader is SYS$SYSTEM:TERTIARY_VMB.EXE.♦

On AXP and VAX nodes, verify the following information in the NCP display:

- Verify the DECnet address for the node.
- Verify the load assist agent is SYS$SHARE:NISCS_LAA.EXE.
- Verify the load assist parameter points to the satellite system disk and correct root.
- Verify that the hardware address matches the satellite's Ethernet address. At the satellite's console prompt, use the information shown in Table 7–2 to obtain the satellite's current LAN hardware address.

  Compare the hardware address values displayed by NCP and at the satellite's console. The values should be identical and should also match the value shown in the SYS$MANAGER:NETNODE_UPDATE.COM file. If the values do not match, you must make appropriate adjustments. For example, if you have recently replaced the satellite's LAN adapter, you must execute CLUSTER_CONFIG.COM CHANGE function to update the network database and NETNODE_UPDATE.COM on the appropriate MOP server.

4. Perform a conversational boot to determine more precisely why the satellite is having trouble booting. The conversational boot procedure displays messages that can help you solve network booting problems. The messages provide information about the state of the network and the communications process between the satellite and the system disk server.

AXP

On AXP systems, the messages displayed are as follows:

- %VMScluster-I-MOPSERVER, MOP server for downline was node UTAH

  This message displays the name of the system providing the DECnet MOP downline load. This message acknowledges that control was properly transferred from the console performing the MOP boot to the image that was loaded.

  If this message is not displayed, either the MOP load failed or the wrong file was MOP downline loaded.

- %VMScluster-I-BUSONLINE, LAN adapter is now running 08-00-2B-2C-CE-E3

  This message displays the LAN address of the Ethernet or FDDI adapter specified in the boot command. Multiple lines can be displayed if multiple LAN devices were specified in the boot command line. The booting satellite can now attempt to locate the system disk by sending a message to the cluster multicast address.

If this message is not displayed, the LAN adapter is not initialized properly. Check the physical network connection. For FDDI, the adapter must be on the ring.

- %VMScluster-I-VOLUNTEER, System disk service volunteered by node EUROPA AA-00-04-00-4C-FD

  This message displays the name of a system claiming to serve the satellite system disk. This system has responded to the multicast message sent by the booting satellite to locate the servers of the system disk.

  If this message is not displayed, one or more of the following situations may be causing the problem:

  - The network path between the satellite and the boot server either is broken or is filtering the local area VMScluster multicast messages.
  - The system disk is not being served.
  - The CLUSTER_AUTHORIZE.DAT file on the system disk does not match the other cluster members.

- %VMScluster-I-CREATECH, Creating channel to node EUROPA 08-00-2B-2C-CE-E2 08-00-2B-12-AE-A2

  This message displays the LAN address of the local LAN adapter (first address) and of the remote LAN adapter (second address) that form a communications path through the network. These adapters can be used to support a NISCA virtual circuit for booting. Multiple messages can be displayed if either multiple LAN adapters were specified on the boot command line or the system serving the system disk has multiple LAN adapters.

  If you do not see as many of these messages as you expect, there may be network problems related to the LAN adapters whose addresses are not displayed. Use the Local Area VMScluster Network Failure Analysis Program for better troubleshooting (see Section E.2).

- %VMScluster-I-OPENVC, Opening virtual circuit to node EUROPA

  This message displays the name of a system that has established an NISCA virtual circuit to be used for communications during the boot process. Booting uses this virtual circuit to connect to the remote MSCP server.

- %VMScluster-I-MSCPCONN, Connected to a MSCP server for the system disk, node EUROPA

  This message displays the name of a system that is actually serving the satellite system disk.

  If this message is not displayed, the system that claimed to serve the system disk could not serve the disk. Check the VMScluster configuration.

- %VMScluster-W-SHUTDOWNCH, Shutting down channel to node EUROPA 08-00-2B-2C-CE-E3 08-00-2B-12-AE-A2

  This message displays the LAN address of the local LAN adapter (first address) and of the remote LAN adapter (second address) that have just lost communications. Depending on the type of failure, multiple messages may be displayed if either the booting system or the system serving the system disk has multiple LAN adapters.

- %VMScluster-W-CLOSEVC, Closing virtual circuit to node EUROPA

  This message indicates that NISCA communications have failed to the system whose name is displayed.

- %VMScluster-I-RETRY, Attempting to reconnect to a system disk server

  This message indicates that an attempt will be made to locate another system serving the system disk. The LAN adapters will be reinitialized and all communications will be restarted.

- %VMScluster-W-PROTOCOL_TIMEOUT, NISCA protocol timeout

  Either the booting node has lost connections to the remote system or the remote system is no longer responding to requests made by the booting system. In either case, the booting system has declared a failure and will reestablish communications to a boot server.♦

C.1.4 Computer Fails to Join the Cluster

If a computer boots but fails to join the cluster, perform the following checks:

- Verify that VMScluster software has been loaded. Look for connection manager (%CNXMAN) messages like those shown in Section C.1.1. If no such messages are displayed, VMScluster software probably was not loaded at boot time. Reboot the computer in conversational mode. At the SYSBOOT> prompt, set the VAXCLUSTER parameter to 2. (In local area or mixed-interconnect clusters, you must also set NISCS_LOAD_PEA0 to 1 and VAXCLUSTER to 2.) Note that these parameters should also be set in the computer's MODPARAMS.DAT file. For more information about booting a computer in conversational mode, consult your installation and operations guide.

  In local area and mixed-interconnect clusters, verify that the cluster security database file (SYS$COMMON:CLUSTER_AUTHORIZE.DAT) exists and that you have specified the correct group number for this cluster (see Section 7.7.8.1).

- Verify that the computer has booted from the correct disk and system root. If %CNXMAN messages are displayed, and if, after the conversational reboot, the computer still does not join the cluster, check the console output on all active computers and look for messages indicating that one or more computers found a remote computer that conflicted with a known or local computer. Such messages suggest that two computers have booted from the same system root.

  Review the boot command files for all CI computers and ensure that all are booting from the correct disks and from unique system roots. If you find it necessary to modify the computer's bootstrap command procedure (console media), you may be able to do so on another processor that is already running in the cluster. Replace the running processor's console media with the media to be modified, and use the Exchange utility and a text editor to make the required changes. Consult the appropriate processor-specific installation and operations guide for information about examining and editing boot command files.

- Verify that the computer's SCSNODE and SCSSYSTEMID parameters are unique in the cluster. To be eligible to join a cluster, a computer must have unique SCSNODE and SCSSYSTEMID parameter values. Check that the current values do not duplicate any values set for existing VMScluster computers. Note that if one or the other value is not unique, you must either

alter *both* values or reboot all other computers. To check or modify values, you can perform a conversational bootstrap operation. However, for reliable future bootstrap operations, you must specify appropriate values for these parameters in the computer's MODPARAMS.DAT file.

Note

Note that if you change the SCSNODE parameter, you must also change the DECnet node name because both names must be the same. In addition, if you change either the SCSNODE parameter or the SCSSYSTEMID parameter on a node that was previously a VMScluster member, you must reboot the entire cluster.

- Verify the cluster group code and password.

C.1.5 Startup Procedures Fail to Complete

If a computer boots and joins the cluster but appears to hang before startup procedures complete—that is, before you are able to log in to the system—be sure that you have allowed sufficient time for the startup procedures to execute.

If the startup procedures fail to complete after a period that is normal for your site, try to access the procedures from another VMScluster computer and make appropriate adjustments. For example, verify that all required devices are configured and available.

One cause of such a failure could be the lack of some system resource such as NPAGEDYN or page file space. If you suspect that the value for the NPAGEDYN parameter is set too low, you can perform a conversational bootstrap operation to increase it. Use SYSBOOT to check the current value, and then double the value. If this procedure is unsuccessful, double the value once more.

If you suspect a shortage of page file space, and if another VMScluster computer is available, you can log in on that computer and use the System Generation utility (SYSGEN) to provide adequate page file space for the problem computer. (Note that insufficent page file space on the booting computer might cause other computers to hang.) If the computer still cannot complete the startup procedures, contact your Digital Services representative.

C.1.6 Diagnosing LAN Component Failures

Troubleshooting LAN component failures (for example, broken LAN bridges) can be performed using the troubleshooting techniques described in Section E.2. This appendix also describes techniques for using the Local Area VMScluster Network Failure Analysis Program.

Intermittent LAN component failures (for example, packet loss) can cause problems in the NISCA transport protocol that delivers SCS messages to other nodes in the VMScluster. Appendix G describes troubleshooting techniques and requirements for LAN analyzer tools.

C.2 Diagnosing Cluster Hangs

Conditions like the following can cause a VMScluster computer to suspend process or system activity (that is, to hang):

- Cluster quorum is lost.
- A shared cluster resource is inaccessible.

Sections C.2.1 and C.2.2 discuss these conditions.

C.2.1 Cluster Quorum Is Lost

The VMScluster quorum scheme coordinates activity among VMScluster computers and ensures the integrity of shared cluster resources. (The quorum scheme is described fully in Section 3.1.1.) Quorum is checked after any change to the cluster configuration—for example, when a voting computer leaves or joins the cluster. If quorum is lost, process creation and I/O activity on all computers in the cluster are blocked.

Information about the loss of quorum and about clusterwide events that cause loss of quorum are sent to the OPCOM process, which broadcasts messages to designated operator terminals. The information is also broadcast to each computer's operator console (OPA0), unless broadcast activity is explicitly disabled on that terminal. However, because quorum may be lost before OPCOM has been able to inform the operator terminals, the messages sent to OPA0 are the most reliable source of information about events that cause loss of quorum.

If quorum is lost, you might add or reboot a node with additional votes.

See also the information about cluster quorum in Section 7.7.5.

C.2.2 Shared Cluster Resource Is Inaccessible

Access to shared cluster resources is coordinated by the distributed lock manager. If a particular process is granted a lock on a resource (for example, a shared data file), other processes in the cluster that request incompatible locks on that resource must wait until the original lock is released. If the original process retains its lock for an extended period, other processes waiting for the lock to be released may appear to hang.

Occasionally, a system activity must acquire a restrictive lock on a resource for an extended period. For example, to perform a volume rebuild, system software takes out an exclusive lock on the volume being rebuilt. While this lock is held, no processes can allocate space on the disk volume. If they attempt to do so, they may appear to hang.

Access to files that contain data necessary for the operation of the system itself is coordinated by the distributed lock manager. For this reason, a process that acquires a lock on one of these resources and is then unable to proceed may cause the cluster to appear to hang.

For example, this condition may occur if a process locks a portion of the system authorization file (SYS$SYSTEM:SYSUAF.DAT) for write access. Any activity that requires access to that portion of the file, such as logging in to an account with the same or similar user name or sending mail to that user name, is blocked until the original lock is released. Normally, this lock is released quickly, and users do not notice the locking operation.

However, if the process holding the lock is unable to proceed, other processes could enter a wait state. Because the authorization file is used during login and for most process creation operations (for example, batch and network jobs), blocked processes could rapidly accumulate in the cluster. Because the distributed lock manager is functioning normally under these conditions, users are not notified by broadcast messages or other means that a problem has occurred.

C.3 Diagnosing CLUEXIT Bugchecks

The operating system performs **bugcheck** operations only when it detects conditions that could compromise normal system activity or endanger data integrity. A **CLUEXIT bugcheck** is a type of bugcheck initiated by the connection manager, the VMScluster software component that manages the interaction of cooperating VMScluster computers. Most such bugchecks are triggered by conditions resulting from hardware failures (particularly failures in communications paths), configuration errors, or system management errors.

The most common conditions that 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.

  This condition can occur upon recovery with battery backup after a power failure, after the repair of an SCS communication link, or after the computer was halted for a period longer than the number of seconds specified for the RECNXINTERVAL parameter 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 SCSMAXMSG system parameter on a computer is too small. Verify that the value of SCSMAXMSG on all VMScluster computers is set to a value that is at the least the default value.

C.4 Diagnosing Port Device Problems

The following sections present information on the CI and LAN port devices. Information is also provided on entries in the system error log and on corrective actions to take when errors occur. Topics include the following:

- Port communication mechanisms
- Port failures
- VMScluster error log entries
- OPA0 error messages

C.4.1 Port Communication Mechanisms

This section describes CI and LAN port communication mechanisms and System Communications Services (SCS) connections.

Port Polling

Shortly after a CI computer boots, the CI port driver (PADRIVER) begins configuration polling to discover other active ports on the CI. Normally, the poller runs every 5 seconds (the default value of the system parameter PAPOLLINTERVAL). In the first polling pass, all addresses are probed over cable path A; on the second pass, all addresses are probed over path B; on the third pass, path A is probed again; and so on.

The poller probes by sending request ID (REQID) packets to all possible port numbers, including itself. Active ports receiving the REQIDs return ID packets (IDREC) to the port issuing the REQID. A port might respond to a REQID even if the computer attached to the port is not running.

In any CI only, local area, or mixed-interconnect cluster, the port drivers perform a start handshake when a pair of ports and port drivers has successfully exchanged ID packets. The port drivers exchange datagrams containing information about the computers, such as the type of computer and the operating system version. If this exchange is successful, each computer declares a virtual circuit open. An open virtual circuit is prerequisite to all other activity.

LAN Communications

In local area and mixed-interconnect clusters, a multicast scheme is used to locate computers on the LAN. Approximately every 3 seconds, the port emulator driver (PEDRIVER) sends a HELLO datagram message via each LAN adapter to a cluster-specific multicast address that is derived from the cluster group number. The driver also enables the reception of these messages from other computers. When the driver receives a HELLO datagram message from a computer with which it does not currently share an open virtual circuit, it attempts to create a circuit. HELLO datagram messages received from a computer with a currently open virtual circuit indicate that the remote computer is operational.

A standard three-message exchange handshake is used to create a virtual circuit. The handshake messages contain information about the transmitting computer and its record of the cluster password. These parameters are verified at the receiving computer, which continues the handshake only if its verification is successful. Thus, each computer authenticates the other. After the final message, the virtual circuit is opened for use by both computers.

System Communications Services (SCS) Connections

System services such as the disk class driver, connection manager, and the MSCP and TMSCP servers communicate between computers with a protocol called System Communications Services (SCS). SCS is responsible primarily for forming and breaking intersystem process connections and for controlling flow of message traffic over those connections. SCS is implemented in the port driver (for example, PADRIVER, PBDRIVER, PEDRIVER, PIDRIVER), and in a loadable piece of the operating system called SCSLOA.EXE (loaded automatically during system initialization).

When a virtual circuit has been opened, a computer periodically probes a remote computer for system services that the remote computer may be offering. The SCS directory service, which makes known services that a computer is offering, is always present both on computers and HSC subsystems. As system services discover their counterparts on other computers and HSC subsystems, they

establish SCS connections to each other. These connections are full duplex and are associated with a particular virtual circuit. Multiple connections are typically associated with a virtual circuit.

C.4.2 Port Failures

Taken together, SCS, the port drivers, and the port itself support a hierarchy of communication paths. Starting with the most fundamental level, these are as follows:

- The physical wires. The Ethernet is a single coaxial cable. FDDI has a pair of fiber-optic cables for transmit and receive. The CI has two pairs of transmitting and receiving cables (path A transmit and receive and path B transmit and receive). For the CI, the operating system software normally sends traffic in automatic path select mode. The port chooses the free path or, if both are free, an arbitrary path (implemented in the cables and star coupler and managed by the port).
- The virtual circuit (implemented partly in the CI port or LAN port emulator driver (PEDRIVER) and partly in SCS software).
- The SCS connections (implemented in system software).

Failures can occur at each communication level and in each component. Failures at one level translate into failures at other levels as follows:

- **Wires**. If the LAN fails or is disconnected, LAN traffic stops or is interrupted, depending on the nature of the failure. For the CI, either path A or B can fail while the virtual circuit remains intact. All traffic is directed over the remaining good path. When the wire is repaired, the repair is detected automatically by port polling, and normal operations resume on all ports.
- **Virtual circuit**. If no path works between a pair of ports, the virtual circuit fails and is closed. A path failure is discovered as follows:
  - For the CI, when polling fails, or when attempts are made to send normal traffic, and the port reports that neither path yielded transmit success.
  - For the LAN, when no multicast HELLO datagram message or incoming traffic is received from another computer.

  When a virtual circuit fails, every SCS connection on it fails. The software automatically reestablishes connections when the virtual circuit is reestablished. Normally, reestablishing a virtual circuit takes several seconds after the problem is corrected.
- **CI port**. If a port fails, all virtual circuits to that port fail, and all SCS connections on those virtual circuits fail. If the port is successfully reinitialized, virtual circuits and connections are reestablished automatically. Normally, port reinitialization and reestablishment of connections take several seconds.
- **LAN adapter**. If a LAN adapter device fails, attempts are made to restart it. If repeated attempts fail, all channels using that adapter are broken. A channel is a pair of LAN addresses, one local and one remote. If the last open channel for a virtual circuit fails, the virtual circuit is closed and the connections are broken.

- **SCS connection**. When the software protocols fail or, in some instances, when the software detects a hardware malfunction, a connection is terminated. Other connections are usually unaffected, as is the virtual circuit. Breaking of connections is also used under certain conditions as an error recovery mechanism—most commonly when there is insufficient nonpaged pool available on the computer.
- **Computer**. If a computer fails because of operator shutdown, bugcheck, or halt and reboot, all other computers in the cluster record the failure as failures of their virtual circuits to the port on the failed computer.

C.4.2.1 Verifying CI Port Functions

Before you boot in a cluster a CI connected computer that is new, just repaired, or suspected of having a problem, you should have Digital Services verify that the computer runs correctly on its own.

To diagnose communication problems, you can invoke the Show Cluster utility and tailor the SHOW CLUSTER report by entering the SHOW CLUSTER command ADD CIRCUIT CABLE_ST. This command adds a class of information about all the virtual circuits as seen from the computer on which you are running SHOW CLUSTER. Primarily, you are checking whether there is a virtual circuit in the OPEN state to the failing computer. Common causes of failure to open a virtual circuit and keep it open are the following:

- Port errors on one side or the other
- Cabling errors
- A port set off line because of software problems
- Insufficient nonpaged pool on both sides
- Failure to set correct values for the system parameters SCSNODE, SCSSYSTEMID, PAMAXPORT, PANOPOLL, PASTIMOUT, and PAPOLLINTERVAL

Run SHOW CLUSTER from each active computer in the cluster to verify whether each computer's view of the failing computer is consistent with every other computer's view. If all the active computers have a consistent view of the failing computer, the problem may be in the failing computer. If, on the other hand, only one of several active computers detects that the newcomer is failing, that particular computer may have a problem.

If no virtual circuit is open to the failing computer, check the bottom of the SHOW CLUSTER display for information on circuits to the port of the failing computer. Virtual circuits in partially open states are shown at the bottom of the display. If the circuit is shown in a state other than OPEN, communications between the local and remote ports are taking place, and the failure is probably at a higher level than in port or cable hardware. Next, check that both paths A and B are good to the failing port. The loss of one path should not prevent a computer from participating in a cluster.

C.4.2.2 Verifying CI Cable Connections

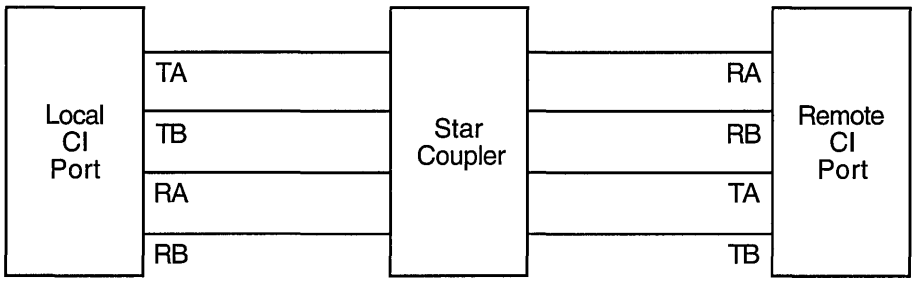
Whenever the configuration poller finds that no virtual circuits are open and that no handshake procedures are currently opening virtual circuits, the poller analyzes its environment. It does so by using the send-loopback-datagram facility of the CI port.

The send-loopback-datagram facility tests the connections between the CI port and the star coupler by routing messages across them. The messages are called loopback datagrams. (The port processes other self-directed messages without using the star coupler or external cables.)

The configuration poller makes entries in the error log whenever it detects a change in the state of a circuit. Note, however, that it is possible two changed-to-failed-state messages can be entered in the log without an intervening changed-to-succeeded-state message. Such a series of entries means that the circuit state continues to be faulty.

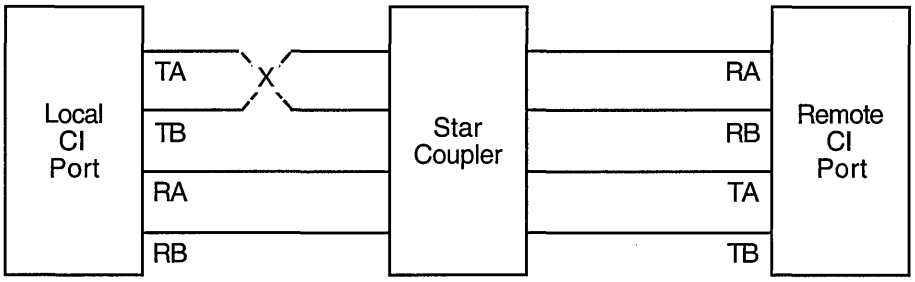
The following paragraphs discuss various incorrect CI cabling configurations and the entries made in the error log when these configurations exist. Figure C–1 shows a two-computer configuration with all cables correctly connected. Figure C–2 shows a CI cluster with a pair of crossed cables.

Figure C–1 Correctly Connected Two-Computer CI Cluster



ZK–1924–GE

Figure C–2 Crossed CI Cable Pair



ZK–1925–GE

If a pair of transmitting cables or a pair of receiving cables is crossed, a message sent on TA is received on RB, and a message sent on TB is received on RA. This is a hardware error condition from which the port cannot recover. An entry is made in the error log indicating that a single pair of crossed cables exists. The entry contains the following lines:

```
DATA CABLE(S) CHANGE OF STATE
PATH  1.  LOOPBACK HAS GONE FROM GOOD TO BAD
```

If this situation exists, you can correct it by reconnecting the cables properly. The cables could be misconnected in several places. The coaxial cables that connect the port boards to the bulkhead cable connectors can be crossed, or the cables can be misconnected to the bulkhead or the star coupler.

The information illustrated in Figure C–2 is represented more simply in Configuration 1. It shows the cables positioned as in Figure C–2, but it does not show the star coupler or the computers. The labels LOC and REM indicate the pairs of transmitting (T) and receiving (R) cables on the local and remote computers, respectively.

Configuration 1

```
T x   = R

R =   = T

LOC   REM
```

The pair of crossed cables causes loopback datagrams to fail on the local computer but to succeed on the remote computer. Crossed pairs of transmitting cables and crossed pairs of receiving cables cause the same behavior.

Note that only an odd number of crossed cable pairs causes these problems. If an even number of cable pairs is crossed, communications succeed. An error log entry is made in some cases, however, and the contents of the entry depends on which pairs of cables are crossed.

Configuration 2 shows two-computer clusters with the combinations of two crossed cable pairs. These crossed pairs cause the following entry to be made in the error log of the computer that has the cables crossed:

```
DATA CABLE(S) CHANGE OF STATE
CABLES HAVE GONE FROM UNCROSSED TO CROSSED
```

Loopback datagrams succeed on both computers, and communications are possible.

Configuration 2

```
T x   = R        T =   x R

R x   = T        R =   x T

LOC   REM        LOC   REM
```

Configuration 3 shows the possible combinations of two pairs of crossed cables that cause loopback datagrams to fail on both computers in the cluster. Communications can still take place between the computers. An entry stating that cables are crossed is made in the error log of each computer.

Configuration 3

```
T x   = R        T =   x R

R =   x T        R x   = T

LOC   REM        LOC   REM
```

Configuration 4 shows the possible combinations of two pairs of crossed cables that cause loopback datagrams to fail on both computers in the cluster, but allow communications. No entry stating that cables are crossed is made in the error log of either computer.

Configuration 4

```
T x   x R        T =   = R

R =   = T        R x   x T

LOC   REM        LOC   REM
```

Configuration 5 shows the possible combinations of four pairs of crossed cables. In each case, loopback datagrams fail on the computer that has only one crossed pair of cables. Loopback datagrams succeed on the computer with both pairs crossed. No communications are possible.

Configuration 5

```
T x   x R        T x   = R        T =   x R        T x   x R

R x   = T        R x   x T        R x   x T        R =   x T

LOC   REM        LOC   REM        LOC   REM        LOC   REM
```

If all four cable pairs between two computers are crossed, communications succeed, loopback datagrams succeed, and no crossed-cable message entries are made in the error log. You might detect such a condition by noting error log entries made by a third computer in the cluster, but only if the third computer has one of the crossed-cable cases described.

C.4.2.3 Repairing CI Cables

This section describes some ways in which Digital Services can make repairs on a running computer. This information is provided to aid system managers in scheduling repairs.

For cluster software to survive cable-checking activities or cable-replacement activities, you must be sure that either path A or path B is intact at all times between each port and between every other port in the cluster.

For example, you can remove path A and path B in turn from a particular port to the star coupler. To make sure that the configuration poller finds a path that was previously faulty but is now operational, follow these steps:

1. Remove path B.
2. After the poller has discovered that path B is faulty, reconnect path B.
3. Wait two poller intervals, and then enter the DCL command SHOW CLUSTER to make sure that the poller has reestablished path B. Or, enter the DCL command SHOW CLUSTER/CONTINUOUS followed by the SHOW CLUSTER command ADD CIRCUITS, CABLE_ST. Wait until SHOW CLUSTER tells you that path B has been reestablished.
4. Remove path A.
5. After the poller has discovered that path A is faulty, reconnect path A.
6. Wait two poller intervals to make sure that the poller has reestablished path A.

If both paths are lost at the same time, the virtual circuits are lost between the port with the broken cables and all other ports in the cluster. This condition will in turn result in loss of SCS connections over the broken virtual circuits. However, recovery from this situation is automatic after an interruption in service on the affected computer. The length of the interruption varies, but it is usually approximately two poller intervals (or 10 seconds) at the default system parameter settings.

C.4.2.4 Verifying LAN Connections

The Local Area VMScluster Network Failure Analysis program described in Section E.1.3 uses the HELLO datagram messages to verify continuously the network paths (channels) used by PEDRIVER. This verification process, combined with physical description of the network, can isolate failing network components. Failing channels are grouped together and mapped onto the physical network description. The failure analysis calls out the common components related to the channel failures.

C.4.3 Analyzing Error Log Entries for Port Devices

To anticipate and avoid potential problems, you must monitor events recorded in the error log. From the total error count, displayed by the DCL command SHOW DEVICES *device-name*, you can determine whether errors are increasing. If so, you should examine the error log.

The DCL command ANALYZE/ERROR_LOG invokes the Error Log utility to report the contents of an error log file. (For more information on the Error Log utility, see the *OpenVMS System Management Utilities Reference Manual*.)

Note that some error log entries are informational only and require no action. For example, if you shut down a computer in the cluster, all other active computers that have open virtual circuits between themselves and the computer that has been shut down make entries in their error logs. Such computers record up to three errors for the event:

1. Path A received no response.
2. Path B received no response.
3. The virtual circuit is being closed.

These messages are normal and reflect the change of state in the circuits to the computer that has been shut down.

On the other hand, some error log entries are made for problems that degrade operation or for nonfatal hardware problems. The operating system might continue to run satisfactorily under these conditions. The purpose of detecting these problems early is to prevent nonfatal problems (such as loss of a single CI path) from becoming serious problems (such as loss of both paths).

C.4.3.1 Error Log Entry Formats

Errors and other events on the CI or LAN cause port drivers to enter information in the system error log. The two formats used for error log entries are the **device-attention** format and the **logged-message** format. Sections C.4.3.2 and C.4.3.3 describe those formats.

Device-attention entries for the CI record events that, in general, are indicated by the setting of a bit in a hardware register. For the LAN, device-attention entries typically record errors on an LAN adapter device. Logged-message entries record the receipt of a message packet that contains erroneous data or that signals an error condition.

C.4.3.2 Device-Attention Entries

Example C–1 shows device-attention entries for the CI. The left column gives the name of a device register or a memory location. The center column gives the value contained in that register or location, and the right column gives an interpretation of that value.

Example C–1 CI Device-Attention Entry

```
************************ ENTRY    83. **************************** ❶
ERROR SEQUENCE 10.                    LOGGED ON:      SID 0150400A
DATE/TIME 15-APR-1990 11:45:27.61                 SYS_TYPE 01010000 ❷
DEVICE ATTENTION    KA780                                           ❸
                    SCS NODE: MARS

CI SUB-SYSTEM, MARS$PAA0: - PORT POWER DOWN                         ❹

      CNFGR           00800038
                                    ADAPTER IS CI
                                    ADAPTER POWER-DOWN
      PMCSR           000000CE
                                    MAINTENANCE TIMER DISABLE
                                    MAINTENANCE INTERRUPT ENABLE
                                    MAINTENANCE INTERRUPT FLAG
                                    PROGRAMMABLE STARTING ADDRESS
                                    UNINITIALIZED STATE
      PSR             80000001
                                    RESPONSE QUEUE AVAILABLE
                                    MAINTENANCE ERROR
      PFAR            00000000
      PESR            00000000
      PPR             03F80001
      UCB$B_ERTCNT          32                                        ❺
                                    50. RETRIES REMAINING
      UCB$B_ERTMAX          32                                        ❻
                                    50. RETRIES ALLOWABLE
      UCB$L_CHAR      0C450000
                                    SHAREABLE
                                    AVAILABLE
                                    ERROR LOGGING
                                    CAPABLE OF INPUT
                                    CAPABLE OF OUTPUT
      UCB$W_STS             0010
                                    ONLINE
      UCB$W_ERRCNT          000B                                      ❼
                                    11. ERRORS THIS UNIT
```

The following are descriptions of device-attention entries in Example C–1:

❶ The first two lines are the entry heading. These lines contain the number of the entry in this error log file, the sequence number of this error, and the identification number (SID) of this computer. Each entry in the log file contains such a heading.

❷ This line contains the date and time and the computer type.

❸ The next two lines contain the entry type, the processor type (KA780), and the computer's SCS node name.

❹ This line shows the name of the subsystem and the device that caused the entry and the reason for the entry. The CI subsystem's device PAA0 on MARS was powered down.

The next 15 lines contain the names of hardware registers in the port, their contents, and interpretations of those contents. See the appropriate CI hardware manual for a description of all the CI port registers.

The CI port can recover from many errors, but not all. When an error occurs from which the CI cannot recover, the port notifies the port driver. The port driver logs the error and attempts to reinitialize the port. If the port fails after 50 such initialization attempts, the driver takes it off line, unless the system disk is connected to the failing port or unless this computer is supposed to be a cluster member. If the CI port is required for system disk access or cluster participation and all 50 reinitialization attempts have been used, then the computer bugchecks with a CIPORT-type bugcheck. Once a CI port is off line, you can put the port back on line only by rebooting the computer.

❺ The UCB$B_ERTCNT field contains the number of reinitializations that the port driver can still attempt. The difference between this value and UCB$B_ERTMAX is the number of reinitializations already attempted.

❻ The UCB$B_ERTMAX field contains the maximum number of times the port can be reinitialized by the port driver.

❼ The UCB$W_ERRCNT field contains the total number of errors that have occurred on this port since it was booted. This total includes both errors that caused reinitialization of the port and errors that did not.

Example C–2 shows device-attention entries for the LAN. The left column gives the name of a device register or a memory location. The center column gives the value contained in that register or location, and the right column gives an interpretation of that value.

Example C–2 LAN Device-Attention Entry

```
************************ ENTRY   80. ***************************** ❶
ERROR SEQUENCE 26.                    LOGGED ON:      SID 08000000
DATE/TIME 15-APR-1990 11:30:53.07                SYS_TYPE 01010000 ❷
DEVICE ATTENTION  KA630                                            ❸
                  SCS NODE: PHOBOS
NI-SCS SUB-SYSTEM, PHOBOS$PEA0:                                    ❹
       FATAL ERROR DETECTED BY DATALINK                            ❺

       STATUS1          0000002C                                  ❻
       STATUS2          00000000
       DATALINK UNIT       0001                                   ❼
       DATALINK NAME    41515803                                  ❽
                        00000000
                        00000000
                        00000000
                                      DATALINK NAME = XQA1:
```

(continued on next page)

Example C–2 (Cont.) LAN Device-Attention Entry

```
REMOTE NODE     00000000                                        ❾
                00000000
                00000000
                00000000
REMOTE ADDR     00000000                                        ❿
                    0000
LOCAL ADDR      000400AA                                        ⓫
                    4C07
                                ETHERNET ADDR = AA-00-04-00-07-4C
ERROR CNT           0001                                        ⓬
                                1. ERROR OCCURRENCES THIS ENTRY
UCB$W_ERRCNT        0007
                                7. ERRORS THIS UNIT
```

The following are descriptions of LAN device-attention entries in Example C–2:

❶ The first two lines are the entry heading. These lines contain the number of the entry in this error log file, the sequence number of this error, and the identification number (SID) of this computer. Each entry in the log file contains such a heading.

❷ This line contains the date and time and the computer type.

❸ The next two lines contain the entry type, the processor type (KA630), and the computer's SCS node name.

❹ This line shows the name of the subsystem and component that caused the entry.

❺ This line shows the reason for the entry. The LAN driver has shut down the data link because of a fatal error. The data link will be restarted automatically, if possible.

❻ STATUS1 shows the I/O completion status returned by the LAN driver. STATUS2 is the VCI event code delivered to PEDRIVER by the LAN driver. The event values and meanings are described in the following table:

Event Code	Meaning
1200	Port usable
1201	Port unusable
1202	Change address

If a message transmit was involved, the status applies to that transmit.

❼ DATALINK UNIT shows the unit number of the LAN device on which the error occurred.

❽ DATALINK NAME is the name of the LAN device on which the error occurred.

❾ REMOTE NODE is the name of the remote node to which the packet was being sent. If zeros are displayed, either no remote node was available or no packet was associated with the error.

❿ REMOTE ADDR is the LAN address of the remote node to which the packet was being sent. If zeros are displayed, no packet was associated with the error.

⓫ LOCAL ADDR is the LAN address of the local node.

⓬ ERROR CNT. Because some errors can occur at extremely high rates, some error log entries represent more than one occurrence of an error. This field indicates how many. The errors counted occurred in the 3 seconds preceding the timestamp on the entry.

C.4.3.3 Logged-Message Entries

Logged-message entries are made when the CI or LAN port receives a response that contains either data that the port driver cannot interpret or an error code in the status field of the response.

Example C–3 shows a CI logged-message entry with an error code in the status field PPD$B_STATUS.

Example C–3 CI Logged-Message Entry

```
************************ ENTRY     3. ***************************** ❶
ERROR SEQUENCE 3.                          LOGGED ON SID 01188542

ERL$LOGMESSAGE, 15-APR-1993 13:40:25.13                              ❷
               KA780 REV #3. SERIAL #1346.    MFG PLANT 15.          ❸

CI SUB-SYSTEM, MARS$PAA0:                                            ❹
DATA CABLE(S) STATE CHANGE - PATH #0. WENT FROM GOOD TO BAD          ❺

     LOCAL STATION ADDRESS, 000000000002 (HEX)                       ❻

     LOCAL SYSTEM ID, 000000000001 (HEX)                             ❼

     REMOTE STATION ADDRESS, 000000000004 (HEX)                      ❽

     REMOTE SYSTEM ID, 0000000000A9 (HEX)                            ❾

     UCB$B_ERTCNT          32                                        ❿
                                  50. RETRIES REMAINING
     UCB$B_ERTMAX          32
                                  50. RETRIES ALLOWABLE
     UCB$W_ERRCNT        0001
                                  1. ERRORS THIS UNIT
     PPD$B_PORT            04                                        ⓫
                                  REMOTE NODE #4.
     PPD$B_STATUS          A5                                        ⓬
                                  FAIL
                                  PATH #0., NO RESPONSE
                                  PATH #1., "ACK" OR NOT USED
                                  NO PATH
     PPD$B_OPC             05                                        ⓭
                                  IDREQ
     PPD$B_FLAGS           03                                        ⓮
                                  RESPONSE QUEUE BIT
                                  SELECT PATH #0.
```

(continued on next page)

Example C–3 (Cont.) CI Logged-Message Entry

```
"CI" MESSAGE                                                    ⓯

                    00000000
                    00000000
                    80000004
                    0000FE15
                    4F503000
                    00000507
                    00000000
                    00000000
                    00000000
                    00000000
                    00000000
                    00000000
                    00000000
                    00000000
                    00000000
                    00000000
                    00000000
                    00000000
```

The following are descriptions of CI logged-message entries in Example C–3:

❶ The first two lines are the entry heading. These lines contain the number of the entry in this error log file, the sequence number of the error, and the identification number (SID) of the computer. Each entry in the log file contains a heading.

❷ This line contains the entry type, the date, and the time.

❸ This line contains the processor type (KA780), the hardware revision number of the computer (REV #3), the serial number of the computer (SERIAL #1346), and the plant number (15).

❹ This line shows the name of the subsystem and the device that caused the entry.

❺ This line gives the reason for the entry (one or more data cables have changed state) and a more detailed reason for the entry. Path 0, which the port used successfully before, cannot be used now.

Note

ANALYZE/ERROR_LOG uses the notation "path 0" and "path 1"; cable labels use the notation "path A (=0)" and "path B (=1)".

❻ The local (❻) and remote (❽) station addresses are the port numbers (range 0 to 15) of the local and remote ports. The port numbers are set in hardware switches by Digital Services.

❼ The local (❼) and remote (❾) system IDs are the SCS system IDs set by the system parameter SCSSYSTEMID for the local and remote systems. For HSC subsystems, the system ID is set with the HSC console.

❽ See ❻.

❾ See ❼.

⑩ The next three lines consist of the entry fields that begin with UCB$. These fields give information on the contents of the unit control block (UCB) for this CI device.

⑪ The lines that begin with PPD$ are fields in the message packet that the local port has received. PPD$B_PORT contains the station address of the remote port. In a loopback datagram, however, this field contains the local station address.

⑫ The PPD$B_STATUS field contains information about the nature of the failure that occurred during the current operation. When the operation completes without error, ERF prints the word NORMAL beside this field; otherwise, ERF decodes the error information contained in PPD$B_STATUS. Here a NO PATH error occurred because of a lack of response on path 0, the selected path.

⑬ The PPD$B_OPC field contains the code for the operation that the port was attempting when the error occurred. The port was trying to send a request-for-ID message.

⑭ The PPD$B_FLAGS field contains bits that indicate, among other things, the path that was selected for the operation.

⑮ "CI" MESSAGE is a hexadecimal listing of bytes 16 through 83 (decimal) of the response (message or datagram). Because responses are of variable length, depending on the port opcode, bytes 16 through 83 may contain either more or fewer bytes than actually belong to the message.

C.4.3.4 Error Log Entry Descriptions

This section describes error log entries for the CI and LAN ports. Each entry shown is followed by a brief description of what the associated port driver (PADRIVER, PBDRIVER, PEDRIVER) does, and the suggested action a system manager should take. In cases where Software Performance Reports (SPRs) with crash dumps are requested, it is important to capture the crash dumps as soon as possible after the error. For CI entries, note that path A and path 0 are the same path, and that path B and path 1 are the same path.

BIIC FAILURE

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services.

CI PORT TIMEOUT

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Increase the PAPOLLINTERVAL system parameter. If the problem disappears and you are not running privileged user-written software, submit an SPR. Otherwise, contact Digital Services.

11/750 CPU MICROCODE NOT ADEQUATE FOR PORT

Explanation: The port driver sets the port off line with no retries attempted. In addition, if this port is needed because the computer is booted from an HSC subsystem or is participating in a cluster, the computer bugchecks with a UCODEREV code bugcheck.

User Action: Read the appropriate section in the current VAXcluster SPD or VMScluster SPD for information on required computer microcode revisions. Contact Digital Services if necessary.

PORT MICROCODE REV NOT CURRENT, BUT SUPPORTED

Explanation: The port driver detected that the microcode is not at the current level, but the port driver will continue normally. This error is logged as a warning only.

User Action: Contact Digital Services when it is convenient to have the microcode updated.

PORT MICROCODE REV NOT SUPPORTED

Explanation: The port driver sets the port off line without attempting any retries.

User Action: Read the VAXcluster or VMScluster SPD for information on the required CI port microcode revisions. Contact Digital Services if necessary.

DATA CABLE(S) STATE CHANGE
CABLES HAVE GONE FROM CROSSED TO UNCROSSED

Explanation: The port driver logs this event.

User Action: No action needed.

DATA CABLE(S) STATE CHANGE
CABLES HAVE GONE FROM UNCROSSED TO CROSSED

Explanation: The port driver logs this event.

User Action: Check for crossed cable pairs. (See Section C.4.2.2.)

DATA CABLE(S) STATE CHANGE
PATH 0. WENT FROM BAD TO GOOD

Explanation: The port driver logs this event.

User Action: No action needed.

DATA CABLE(S) STATE CHANGE
PATH 0. WENT FROM GOOD TO BAD

Explanation: The port driver logs this event.

User Action: Check path A cables to see that they are not broken or improperly connected.

DATA CABLE(S) STATE CHANGE
PATH 0. LOOPBACK IS NOW GOOD, UNCROSSED

Explanation: The port driver logs this event.

User Action: No action needed.

DATA CABLE(S) STATE CHANGE
PATH 0. LOOPBACK WENT FROM GOOD TO BAD

Explanation: The port driver logs this event.

User Action: Check for crossed cable pairs or faulty CI hardware. (See Sections C.4.2.1 and C.4.2.2.)

DATA CABLE(S) STATE CHANGE
PATH 1. WENT FROM BAD TO GOOD

Explanation: The port driver logs this event.

User Action: No action needed.

DATA CABLE(S) STATE CHANGE
PATH 1. WENT FROM GOOD TO BAD

Explanation: The port driver logs this event.

User Action: Check path B cables to see that they are not broken or improperly connected.

DATA CABLE(S) STATE CHANGE
PATH 1. LOOPBACK IS NOW GOOD, UNCROSSED

Explanation: The port driver logs this event.

User Action: No action needed.

DATA CABLE(S) STATE CHANGE
PATH 1. LOOPBACK WENT FROM GOOD TO BAD

Explanation: The port driver logs this event.

User Action: Check for crossed cable pairs or faulty CI hardware. (See Sections C.4.2.1 and C.4.2.2.)

DATAGRAM FREE QUEUE INSERT FAILURE

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention.

DATAGRAM FREE QUEUE REMOVE FAILURE

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention.

FAILED TO LOCATE PORT MICRO-CODE IMAGE

Explanation: The port driver marks device off line and makes no retries.

User Action: Make sure console volume contains the microcode file CI780.BIN (for the CI780, CI750, or CIBCI) or the microcode file CIBCA.BIN for the CIBCA-AA. Then reboot the computer.

HIGH PRIORITY COMMAND QUEUE INSERT FAILURE

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, 8800) contention.

MSCP ERROR LOGGING DATAGRAM RECEIVED

Explanation: On receipt of an error message from the HSC subsystem, the port driver logs the error and takes no other action. You should disable the sending of HSC informational error log datagrams with the appropriate HSC console command because such datagrams take considerable space in the error log data file.

User Action: Error log datagrams are useful to read only if they are not captured on the HSC console for some reason (for example, if the HSC console ran out of paper.) This logged information duplicates messages logged on the HSC console.

INAPPROPRIATE SCA CONTROL MESSAGE

Explanation: The port driver closes the port-to-port virtual circuit to the remote port.

User Action: Submit an SPR to Digital. Include the error logs and the crash dumps from the local and remote computers.

INSUFFICIENT NON-PAGED POOL FOR INITIALIZATION

Explanation: The port driver marks device off line and makes no retries.

User Action: Reboot the computer with a larger value for NPAGEDYN or NPAGEVIR.

LOW PRIORITY CMD QUEUE INSERT FAILURE

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention.

MESSAGE FREE QUEUE INSERT FAILURE

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention.

MESSAGE FREE QUEUE REMOVE FAILURE

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention.

MICRO-CODE VERIFICATION ERROR

Explanation: The port driver detected an error while reading the microcode that it just loaded into the port. The driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services.

NO PATH-BLOCK DURING VIRTUAL CIRCUIT CLOSE

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Submit an SPR to Digital. Include the error log and a crash dump from the local computer.

NO TRANSITION FROM UNINITIALIZED TO DISABLED

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services.

PORT ERROR BIT(S) SET

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: A **maintenance timer expiration** bit may mean that the PASTIMOUT system parameter is set too low and should be increased, especially if the local computer is running privileged user-written software. For all other bits, call Digital Services.

PORT HAS CLOSED VIRTUAL CIRCUIT

Explanation: The port driver closed the virtual circuit that the local port opened to the remote port.

User Action: Check the PPD$B_STATUS field of the error log entry for the reason the virtual circuit was closed. This error is normal if the remote computer crashed or was shut down. For PEDRIVER, ignore the PPD$B_ OPC field value; it is an unknown opcode.

If PEDRIVER logs a large number of these errors, there may be a problem either with the LAN or with a remote system, or nonpaged pool may be insufficient on the local system.

PORT POWER DOWN

Explanation: The port driver halts port operations and then waits for power to return to the port hardware.

User Action: Restore power to the port hardware.

PORT POWER UP

Explanation: The port driver reinitializes the port and restarts port operations.

User Action: No action needed.

RECEIVED CONNECT WITHOUT PATH-BLOCK

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Submit an SPR to Digital. Include the error log and a crash dump from the local computer.

REMOTE SYSTEM CONFLICTS WITH KNOWN SYSTEM

Explanation: The configuration poller discovered a remote computer with SCSSYSTEMID and/or SCSNODE equal to that of another computer to which a virtual circuit is already open.

User Action: Shut down the new computer as soon as possible. Reboot it with a unique SCSYSTEMID and SCSNODE. Do not leave the new computer up any longer than necessary. If you are running a cluster, and two computers with conflicting identity are polling when any other virtual circuit failure takes place in the cluster, then computers in the cluster may crash with a CLUEXIT bugcheck.

RESPONSE QUEUE REMOVE FAILURE

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention.

SCSSYSTEMID MUST BE SET TO NON-ZERO VALUE

Explanation: The port driver sets the port off line without attempting any retries.

User Action: Reboot the computer with a conversational boot and set the SCSSYSTEMID to the correct value. At the same time, check that SCSNODE has been set to the correct nonblank value.

SOFTWARE IS CLOSING VIRTUAL CIRCUIT

Explanation: The port driver closes the virtual circuit to the remote port.

User Action: Check error log entries for the cause of the virtual circuit closure. Faulty transmission or reception on both paths, for example, causes this error and may be detected from the one or two previous error log entries noting bad paths to this remote computer.

SOFTWARE SHUTTING DOWN PORT

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Check other error log entries for the possible cause of the port reinitialization failure.

UNEXPECTED INTERRUPT

Explanation: The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line.

User Action: Contact Digital Services.

UNRECOGNIZED SCA PACKET

Explanation: The port driver closes the virtual circuit to the remote port. If the virtual circuit is already closed, the port driver inhibits datagram reception from the remote port.

User Action: Submit an SPR to Digital. Include the error log file that contains this entry and the crash dumps from both the local and remote computers.

VIRTUAL CIRCUIT TIMEOUT

Explanation: The port driver closes the virtual circuit that the local CI port opened to the remote port. This closure occurs if the remote computer is running CI microcode Version 7 or later, and if the remote computer has failed to respond to any messages sent by the local computer.

User Action: This error is normal if the remote computer has halted, crashed, or was shut down. This error may mean that the local computer's TIMVCFAIL system parameter is set too low, especially if the remote computer is running privileged user-written software.

INSUFFICIENT NON-PAGED POOL FOR VIRTUAL CIRCUITS

Explanation: The port driver closes virtual circuits because of insufficient pool.

User Action: Enter the DCL command SHOW MEMORY to determine pool requirements, and then adjust the appropriate system parameter requirements.

Note

The following descriptions apply only to LAN devices.

FATAL ERROR DETECTED BY DATALINK

Completion status: First longword SS$_NORMAL (00000001), second longword (00001201)

Explanation: The LAN driver stopped the local area VMScluster protocol on the device. This completion status is returned when the SYS$LAVC_STOP_BUS routine completes successfully. The SYS$LAVC_STOP_BUS routine is called either from within the LAVC$STOP_BUS.MAR program found in SYS$EXAMPLES or from a user-written program. The local area VMScluster protocol remains stopped on the specified device until the SYS$LAVC_START_BUS routine executes successfully. The SYS$LAVC_START_BUS routine is called from within the LAVC$START_BUS.MAR program found in SYS$EXAMPLES or from a user-written program.

User Action: If the protocol on the device was stopped inadvertently, then restart the protocol by assembling and executing the LAVC$START_BUS program found in SYS$EXAMPLES (see Appendix E for an explanation of the

local area VMScluster sample programs). Otherwise, this error message can be safely ignored.

Completion status: First longword is any value other than (00000001), second longword (00001201)

Explanation: The LAN driver has shut down the device because of a fatal error and is returning all outstanding transmits with SS$_OPINCOMPL. The LAN device is restarted automatically.

User Action: Infrequent occurrences of this error are typically not a problem. If the error occurs frequently or is accompanied by loss or reestablishment of connections to remote computers, there may be a hardware problem. Check for the proper LAN adapter revision level or contact Digital Services.

Completion status: First longword (undefined), second longword (00001200)

Explanation: The LAN driver has restarted the device successfully after a fatal error. This error log message is usually preceded by a FATAL ERROR DETECTED BY DATALINK error log message whose first completion status longword is anything other than 00000001 and whose second completion status longword is 00001201.

User Action: No action needed.

TRANSMIT ERROR FROM DATALINK

Completion status: SS$_OPINCOMPL (000002D4)

Explanation: The LAN driver is in the process of restarting the data link because an error forced the driver to shut down the controller and all users (see FATAL ERROR DETECTED BY DATALINK).

Completion status: SS$_DEVREQERR (00000334)

Explanation: The LAN controller tried to transmit the packet 16 times and failed because of defers and collisions. This condition indicates that LAN traffic is heavy.

Completion status: SS$_DISCONNECT (0000204C)

Explanation: There was a loss of carrier during or after the transmit.

User Action: The port emulator automatically recovers from any of these errors, but many such errors indicate either that the LAN controller is faulty or that the LAN is overloaded. If you suspect either of these conditions, contact Digital Services.

INVALID CLUSTER PASSWORD RECEIVED

Explanation: A computer is trying to join the cluster using the correct cluster group number for this cluster but an invalid password. The port emulator discards the message. The probable cause is that another cluster on the LAN is using the same cluster group number.

User Action: Provide all clusters on the same LAN with unique cluster group numbers.

NISCS PROTOCOL VERSION MISMATCH RECEIVED

Explanation: A computer is trying to join the cluster using a version of the cluster LAN protocol that is incompatible with the one in use on this cluster.

User Action: Install a version of the operating system that uses a compatible protocol, or change the cluster group number so that the computer joins a different cluster.

C.4.4 OPA0 Error Messages

Port drivers detect certain error conditions and attempt to log them. Under some circumstances, attempts to log errors to the error logging device can fail. Such failures can occur because the error logging device is not accessible when attempts are made to log the error condition. Because of the central role that the port device plays in clusters, the loss of error-logged information in such cases makes it difficult to diagnose and fix problems.

A second, redundant method of error logging captures at least some of the information about port device error conditions that would otherwise be lost. This method consists of broadcasting selected information about the error condition to OPA0 in addition to the port driver's attempt to log the error condition to the error logging device. The port driver attempts both OPA0 error broadcasting and standard error logging under any of the following circumstances:

- The system disk has not yet been mounted.
- The system disk is undergoing mount verification.
- During mount verification, the system disk drive contains the wrong volume.
- Mount verification for the system disk has timed out.
- The local computer is participating in a cluster, and quorum has been lost.

Note the implicit assumption that the system and error logging devices are one and the same.

This method of reporting errors is not entirely reliable. Because of the way OPA0 error broadcasting is performed, some error conditions may not be reported. This situation occurs whenever a second error condition is detected before the port driver has been able to broadcast the first error condition to OPA0. In such a case, only the first error condition is reported to OPA0, because that condition is deemed to be the more important one.

Certain error conditions are always broadcast to OPA0, regardless of whether the error logging device is accessible. In general, these are errors that cause the port to shut down either permanently or temporarily.

One OPA0 error message for each error condition is always logged. The text of each error message is similar to the text in the summary displayed by formatting the corresponding standard error log entry using the Error Log utility. (See Section C.4.3.4 for a list of Error Log utility summary messages and their explanations.)

Many of the OPA0 error messages contain some optional information, such as the remote port number, CI packet information (flags, port operation code, response status, and port number fields), or specific CI port registers.

Following is a list of OPA0 error messages, divided by error type. See the CI hardware documentation for a detailed description of the CI port registers (CNF = configuration register; PMC = port maintenance and control register; PSR = port status register), which are optionally displayed for certain error conditions. The codes, always file accessible, specify whether the message is always logged on OPA0 or is logged only when the system device is inaccessible.

Software Errors During Initialization (Always Logged on OPA0)

%Pxxn, Insufficient Non-Paged Pool for Initialization

%Pxxn, Failed to Locate Port Micro-code Image

%Pxxn, SCSSYSTEMID has NOT been set to a Non-Zero Value

Hardware Errors (Always Logged on OPA0)

%Pxxn, BIIC failure - BICSR/BER/CNF xxxxxx/xxxxxx/xxxxxx

%Pxxn, Micro-code Verification Error

%Pxxn, Port Transition Failure - CNF/PMC/PSR xxxxxx/xxxxxx/xxxxxx

%Pxxn, Port Error Bit(s) Set - CNF/PMC/PSR xxxxxx/xxxxxx/xxxxxx

%Pxxn, Port Power Down

%Pxxn, Port Power Up

%Pxxn, Unexpected Interrupt - CNF/PMC/PSR xxxxxx/xxxxxx/xxxxxx

%Pxxn, CI Port Timeout

%Pxxn, CI port ucode not at required rev level. RAM/PROM rev is xxxx/xxxx

%Pxxn, CI port ucode not at current rev level. RAM/PROM rev is xxxx/xxxx

%Pxxn, CPU ucode not at required rev level for CI activity

Queue Interlock Failures (Always Logged on OPA0)

%Pxxn, Message Free Queue Remove Failure

%Pxxn, Datagram Free Queue Remove Failure

%Pxxn, Response Queue Remove Failure

%Pxxn, High Priority Command Queue Insert Failure

%Pxxn, Low Priority Command Queue Insert Failure

%Pxxn, Message Free Queue Insert Failure

%Pxxn, Datagram Free Queue Insert Failure

Errors Signaled with a CI Packet

%Pxxn, Unrecognized SCA Packet - FLAGS/OPC/STATUS/PORT xx/xx/xx/xx
(ALWAYS)

%Pxxn, Port has Closed Virtual Circuit - REMOTE PORT xxx
(ALWAYS)

%Pxxn, Software Shutting Down Port
(ALWAYS)

%Pxxn, Software is Closing Virtual Circuit - REMOTE PORT xxx
(ALWAYS)

%Pxxn, Received Connect Without Path-Block - FLAGS/OPC/STATUS/PORT xx/xx/xx/xx
(ALWAYS)

%Pxxn, Inappropriate SCA Control Message - FLAGS/OPC/STATUS/PORT xx/xx/xx/xx
(ALWAYS)

%Pxxn, No Path-Block During Virtual Circuit Close - REMOTE PORT xxx
(ALWAYS)

%Pxxn, HSC Error Logging Datagram Received - REMOTE PORT xxx
(INACCESSIBLE)

%Pxxn, Remote System Conflicts with Known System - REMOTE PORT xxx
(ALWAYS)

%Pxxn, Virtual Circuit Timeout - REMOTE PORT xxx
(ALWAYS)

```
%Pxxn, Parallel Path is Closing Virtual Circuit - REMOTE PORT xxx
    (ALWAYS)

%Pxxn, Insufficient Non-paged Pool for Virtual Circuits
    (ALWAYS)
```

Cable Change-of-State Notification

```
%Pxxn, Path #0. Has gone from GOOD to BAD - REMOTE PORT xxx
    (INACCESSIBLE)

%Pxxn, Path #1. Has gone from GOOD to BAD - REMOTE PORT xxx
    (INACCESSIBLE)

%Pxxn, Path #0. Has gone from BAD to GOOD - REMOTE PORT xxx
    (INACCESSIBLE)

%Pxxn, Path #1. Has gone from BAD to GOOD - REMOTE PORT xxx
    (INACCESSIBLE)

%Pxxn, Cables have gone from UNCROSSED to CROSSED - REMOTE PORT xxx
    (INACCESSIBLE)

%Pxxn, Cables have gone from CROSSED to UNCROSSED - REMOTE PORT xxx
    (INACCESSIBLE)

%Pxxn, Path #0. Loopback has gone from GOOD to BAD - REMOTE PORT xxx
    (ALWAYS)

%Pxxn, Path #1. Loopback has gone from GOOD to BAD - REMOTE PORT xxx
    (ALWAYS)

%Pxxn, Path #0. Loopback has gone from BAD to GOOD - REMOTE PORT xxx
    (ALWAYS)

%Pxxn, Path #1. Loopback has gone from BAD to GOOD - REMOTE PORT xxx
    (ALWAYS)

%Pxxn, Path #0. Has become working but CROSSED to Path #1. - REMOTE PORT xxx
    (INACCESSIBLE)

%Pxxn, Path #1. Has become working but CROSSED to Path #0. - REMOTE PORT xxx
    (INACCESSIBLE)
```

Note that if the port driver can identify the remote SCS node name of the affected computer, the driver replaces the "REMOTE PORT *xxx*" text with "REMOTE SYSTEM X...", where X... is the value of the system parameter SCSNODE on the remote computer. If the remote SCS node name is not available, the port driver uses the existing message format.

Two other messages concerning the CI port appear on OPA0:

```
%Pxxn, CI port is reinitializing (xxx retries left.)

%Pxxn, CI port is going off line.
```

The first message indicates that a previous error requiring the port to shut down is recoverable, and that the port will be reinitialized. The "*xxx* retries left" specifies how many more reinitializations are allowed before the port must be left permanently off line. Each reinitialization of the port (for reasons other than power fail recovery) causes approximately 2 KB of nonpaged pool to be lost.

The second message indicates that a previous error is not recoverable, and that the port will be left off line. In this case, the only way to recover the port is to reboot the computer.

D

Local Area VMScluster Network Connections

This appendix describes the sample local area network connections in a local area VMScluster.

Each network connection requires a LAN adapter in the host system. Thick wire connections require a transceiver cable connected to one of the following network components:

- Transceiver, H4000, H4005, or equivalent (see Figure D–1).
- DELNI network interconnect or equivalent (see Figure D–2).
- DESTA network adapter or equivalent, plus the components required to make a ThinWire connection (see Figure D–3).

Figure D–1 Sample Transceiver/Thick Wire Ethernet Connection

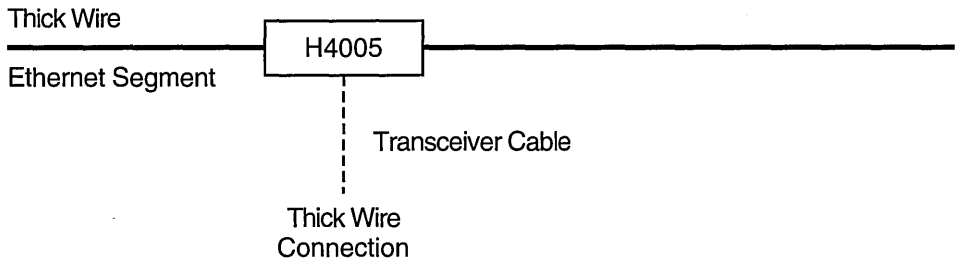


Figure D–2 Sample DELNI Connection

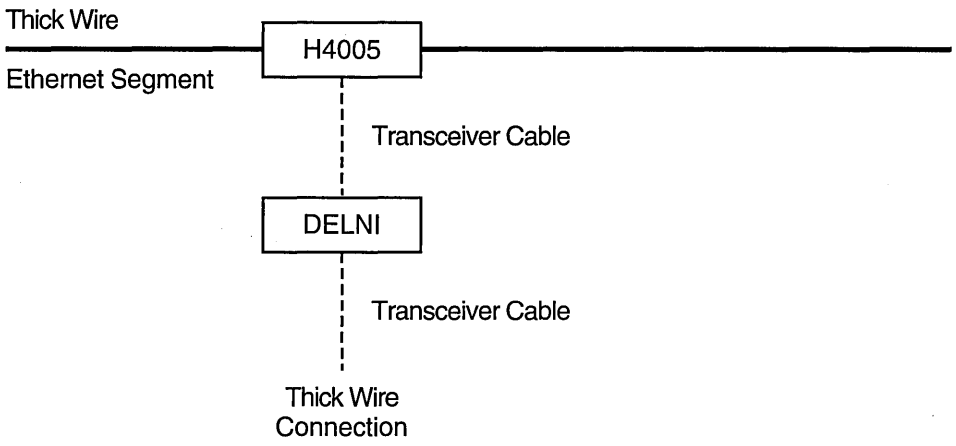
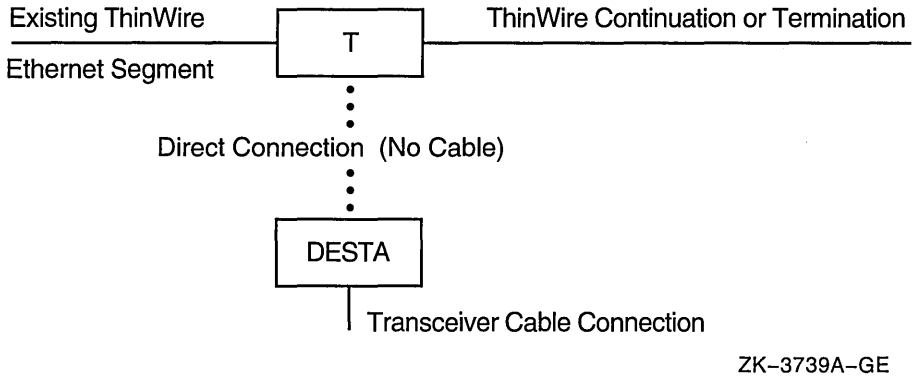


Figure D–3 Sample DESTA Connection



ThinWire connections are typically created to:

- Extend an existing ThinWire segment. This connection requires a segment of ThinWire cable and a T-connector. Figure D–4 illustrates this type of connection.
- Make a new connection to an existing DEMPR repeater. This connection requires a segment of ThinWire cable, a T-connector, and a terminator. Figure D–5 illustrates this type of connection.

Figure D–4 Sample Connection to an Existing ThinWire Segment

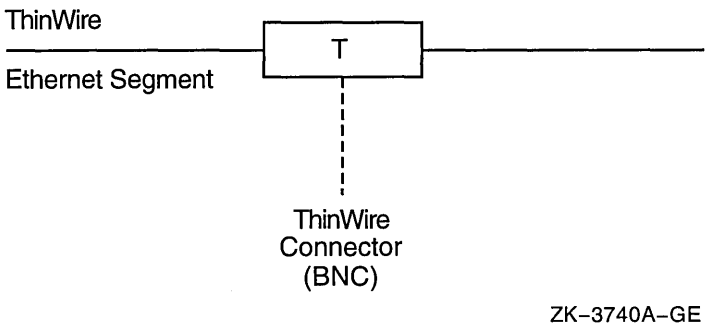
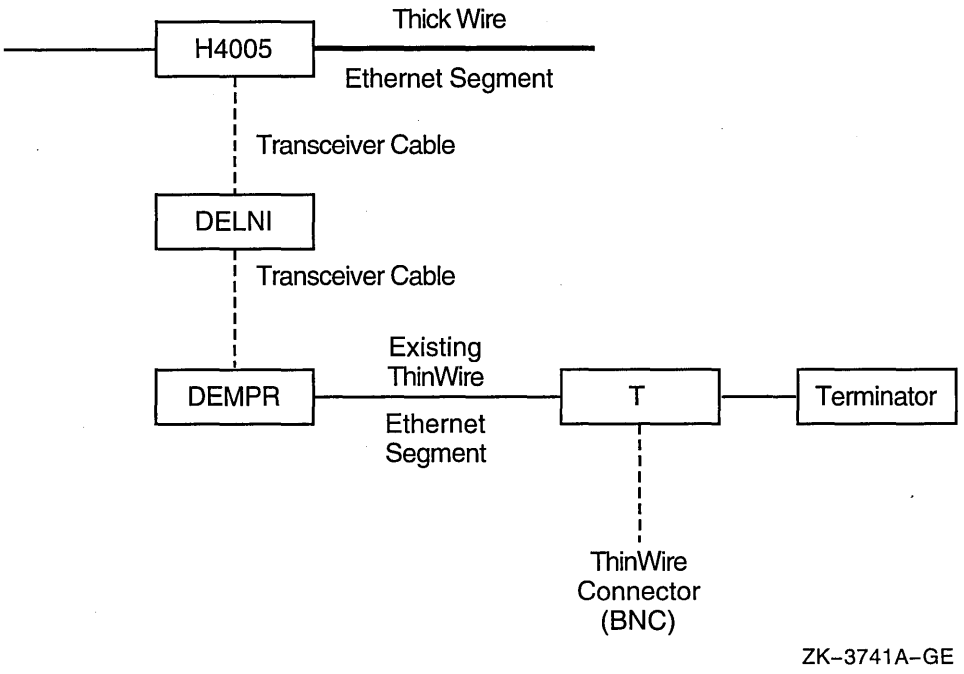


Figure D–5 Sample New Connection to an Existing DEMPR



E

Local Area VMScluster Sample Programs

Sample programs are provided in SYS$EXAMPLES to start and stop the local area VMScluster protocol on a LAN adapter, and to enable local area VMScluster network failure analysis.

For descriptions of the programs, see Section E.1. For information about editing and using the Local Area VMScluster Network Failure Analysis program, refer to Section E.2.

E.1 Sample Programs for Local Area VMSclusters

You can find three sample programs for local area VMSclusters in SYS$EXAMPLES:

- LAVC$START_BUS.MAR—Program to start the local area VMScluster protocol on a specified LAN adapter
- LAVC$STOP_BUS.MAR—Program to stop the local area VMScluster protocol on a specified LAN adapter
- LAVC$FAILURE_ANALYSIS.MAR—Template program to enable the Local Area VMScluster Network Failure Analysis

PEDRIVER starts the protocol on all of the LAN adapters in the cluster. LAVC$START_BUS.MAR and LAVC$STOP_BUS.MAR are provided for cluster managers who want to split the network load according to protocol type and therefore do not want the local area VMScluster protocol running on all of the LAN adapters.

In the SYS$EXAMPLES directory, you can also find the command file, LAVC$BUILD.COM, which assembles and links these sample programs.

The following sections describe these programs in more detail.

E.1.1 Starting the Local Area VMScluster Protocol on a LAN Adapter

The sample program LAVC$START_BUS.MAR, provided in SYS$EXAMPLES, starts the local area VMScluster protocol on a specified LAN adapter.

To build this program, first copy the files LAVC$START_BUS.MAR and LAVC$BUILD.COM from SYS$EXAMPLES to your local directory. Then assemble and link the sample program using the following command:

```
$ @LAVC$BUILD.COM LAVC$START_BUS.MAR
```

To start the local area VMScluster protocol on a LAN adapter, perform the following steps:

1. Define the foreign command (DCL symbol).
2. Execute the foreign command (LAVC$START_BUS.EXE), followed by the name of the LAN adapter on which you want to start the protocol.

Note

You must have the PHY_IO privilege in order to execute LAVC$START_BUS.EXE.

The following example starts the cluster protocol on LAN adapter ETA0:

```
$ START_BUS:==$SYS$DISK:[ ]LAVC$START_BUS.EXE
$ START_BUS ETA
```

E.1.2 Stopping the Local Area VMScluster Protocol on a LAN Adapter

The sample program LAVC$STOP_BUS.MAR, provided in SYS$EXAMPLES, stops the local area VMScluster protocol on a specified LAN adapter.

Caution

Stopping the local area VMScluster protocol on all LAN adapters causes the local area VMScluster satellites to hang and could cause cluster systems to fail with a CLUEXIT bugcheck.

Note

When the LAVC$STOP_BUS module executes successfully, the following device-attention entry is written to the system error log:

```
DEVICE ATTENTION . . .

NI-SCS SUB-SYSTEM . . .

FATAL ERROR DETECTED BY DATALINK . . .
```

In addition, the following hexidecimal values are written to the STATUS field of the entry:

First longword (00000001)
Second longword (00001201)

This error log entry indicates expected behavior and can be ignored. However, if the first longword of the STATUS field contains a value other than hexidecimal value 00000001, an error has occurred and further investigation may be necessary.

To build this program, first copy the files LAVC$STOP_BUS.MAR and LAVC$BUILD.COM from SYS$EXAMPLES to your local directory. Then assemble and link the sample program using the following command:

```
$ @LAVC$BUILD.COM LAVC$STOP_BUS.MAR
```

To stop the local area VMScluster protocol on a LAN adapter, perform the following steps:

1. Define the foreign command (DCL symbol).
2. Execute the foreign command (LAVC$STOP_BUS.EXE), followed by the name of the LAN adapter on which you want to stop the protocol.

Note

You must have the PHY_IO privilege to execute LAVC$STOP_BUS.EXE.

The following example stops the cluster protocol on LAN adapter ETA0:.

```
$ STOP_BUS:==$SYS$DISK[ ]LAVC$STOP_BUS.EXE
$ STOP_BUS ETA
```

E.1.3 Enabling VMScluster Network Failure Analysis

LAVC$FAILURE_ANALYSIS.MAR is a sample program that you can edit and use to help detect and isolate a failed network component. LAVC$FAILURE_ANALYSIS.MAR is located in SYS$EXAMPLES. Edit this sample program to include in it a physical description of the communications network for your cluster. Then assemble and link the program and execute it on one or more cluster systems that will perform the failure analysis. When the program executes, it provides the physical description of your cluster communications network to the set of routines that perform the failure analysis.

Using the network failure analysis program can help reduce the time necessary for detection and isolation of a failing network component and, therefore, significantly increase cluster availability. The program groups channels that fail and compares them with the physical description of the cluster network. The program then develops a list of nonworking network components related to the failed channels and uses OPCOM messages to display the names of components with a probability of causing one or more channel failures. If the network failure analysis cannot verify that a portion of a path (containing multiple components) works, the program calls out the first component in the path as the primary suspect (%LAVC-W-PSUSPECT). The other components are listed as secondary or additional suspects (%LAVC-I-ASUSPECT). When the component works again, OPCOM displays the message %LAVC-S-WORKING.

For information about how to edit and use LAVC$FAILURE_ANALYSIS.MAR, refer to Section E.2.

E.2 Using the Local Area VMScluster Network Failure Analysis Program

LAVC$FAILURE_ANALYSIS.MAR is a sample macro program that you must edit, assemble, link, and execute in order to use. This sample program is located in SYS$EXAMPLES.

To use the network failure analysis program, perform these steps:

1. Collect information specific to your cluster communications network. See Section E.2.1 for more information about this step.
2. Edit a copy of LAVC$FAILURE_ANALYSIS.MAR to include the information you collected. See Section E.2.2.

3. Assemble, link, and debug the program. See Section E.2.3.
4. Execute the program on one or more of the nodes where you plan to perform the network failure analysis. See Section E.2.4.
5. Test the Local Area VMScluster Network Failure Analysis subsystem. See Section E.2.5.

E.2.1 Collecting Information for the Network Failure Analysis Program

To collect the information you will need to edit the network failure analysis program, perform the following steps:

1. Draw a diagram of your VMScluster communications network. When you edit LAVC$FAILURE_ANALYSIS.MAR, you include this drawing (in electronic form) in the program. Your drawing should show the physical layout of the cluster and include the following components:
   - LAN segments or rings
   - LAN bridges
   - Wiring concentrators, DELNI interconnects, or DEMPR repeaters
   - LAN adapters
   - VAX and AXP systems

   For large clusters, you may need to verify the configuration by tracing the cables.
2. Give each component in the drawing a unique label. If your VMScluster contains a large number of nodes, you may want to replace each node name with a shorter abbreviation. Abbreviating node names can help save space in the electronic form of the drawing when you include it in LAVC$FAILURE_ANALYSIS.MAR. For example, you can replace the node name ASTRA with A and call node ASTRA's two LAN adapters A1 and A2.
3. List the following information for each component:
   - Unique label.
   - Type [SYSTEM, LAN_ADP, DELNI].
   - Location (the physical location of the component).
   - LAN address or addresses (if applicable). (Devices such as DELNI interconnects, DEMPR repeaters, and cables do not have LAN addresses.)
4. Classify each component into one of the following categories:
   - Node: VAX or AXP system in the VMScluster configuration.
   - Adapter: LAN adapter on the system that is normally used for VMScluster communications.
   - Component: Generic component in the network. Components in this category can usually be shown to be working if at least one path through them is working. Wiring concentrators, DELNI interconnects, DEMPR repeaters, LAN bridges, and LAN segments and rings typically fall into this category.

- Cloud: Generic component in the network. Components in this category cannot be shown to be working even if one or more paths are shown to be working. This type of component is necessary only when multiple paths exist between two points within the network, such as with redundant bridging between LAN segments. At a high level, multiple paths can exist; however, during operation, this bridge configuration allows only one path to exist at one time. In general, this bridge example is probably better handled by representing the active bridge in the description as a Component and ignoring the standby bridge. (You can identify the active bridge with such network monitoring software as RBMS or DECelms.) With the default bridge parameters, failure of the active bridge will be called out.

5. Use the component labels from step 3 to describe each of the connections in the VMScluster communications network.
6. Choose a node or group of nodes to run the network failure analysis program. You should run the network failure analysis program only on a node that you included in the physical description when you edited LAVC$FAILURE_ANALYSIS.MAR. The network failure analysis program on one node operates independently from other systems in the VMScluster. So, for executing the network failure analysis program, you should choose systems that are not normally shut down. Other good candidates for running the program are systems with the following characteristics:
   - Faster CPU speed
   - Larger amounts of memory
   - More LAN adapters (running local area VMScluster protocol)

Note

The physical description is loaded into nonpaged pool, and all processing is performed at IPL 8. CPU use increases as the average number of network components in the network path increases. CPU use also increases as the total number of network paths increases.

E.2.2 Editing the Network Failure Analysis Program

To edit the network failure analysis program, first copy the files LAVC$FAILURE_ANALYSIS.MAR and LAVC$BUILD.COM from SYS$EXAMPLES to your local directory. Then use the VMScluster network map and the other information you collected to edit the copy of the LAVC$FAILURE_ANALYSIS.MAR.

Example E–1 shows the portion of LAVC$FAILURE_ANALYSIS.MAR that you edit.

Example E–1 Portion of LAVC$FAILURE_ANALYSIS.MAR to Edit

```
;       *** Start edits here ***
;       Edit 1.❶
;
;               Define the hardware components needed to describe
;               the physical configuration.
;

        NEW_COMPONENT   SYSTEM          NODE
        NEW_COMPONENT   LAN_ADP         ADAPTER
        NEW_COMPONENT   DEMPR           COMPONENT
        NEW_COMPONENT   DELNI           COMPONENT
        NEW_COMPONENT   SEGMENT         COMPONENT
        NEW_COMPONENT   NET_CLOUD       CLOUD

;       Edit 2.❷
;
;                       Diagram of a multi-adapter local area VMScluster.
;
;
;        Sa   -------+---------------+---------------+---------------+-------
;                    |               |               |               |
;                    |             MPR_A             |               |
;                    |            .----+----.        |               |
;                    |          1|       1|       1|               |
;                   BrA        ALPHA     BETA      DELTA            BrB
;                    |          2|       2|       2|               |
;                    |            '----+----'        |               |
;                    |               LNI_A           |               |
;                    |               |               |               |
;        Sb   -------+---------------+---------------+---------------+-------
;
;
;       Edit 3.❸
;
;               Label    Node                    Description
;               -----    ------  ---------------------------------------------

        SYSTEM  A,      ALPHA,  < - MicroVAX II; In the Computer room>
        LAN_ADP A1,     ,       <XQA; ALPHA - MicroVAX II; Computer room>,
        LAN_ADP A2,     ,       <XQB; ALPHA - MicroVAX II; Computer room>,

        SYSTEM  B,      BETA,   < - MicroVAX 3500; In the Computer room>
        LAN_ADP B1,     ,       <XQA; BETA - MicroVAX 3500; Computer room>,
        LAN_ADP B2,     ,       <XQB; BETA - MicroVAX 3500; Computer room>,

        SYSTEM  D,      DELTA, < - VAXstation II; In Dan's office>
        LAN_ADP D1,     ,       <XQA; DELTA - VAXstation II; Dan's office>,
        LAN_ADP D2,     ,       <XQB; DELTA - VAXstation II; Dan's office>,

;       Edit 4.❹
;
;               Label each of the other network components.
;

        DEMPR   MPR_A, , <Connected to segment A; In the Computer room>
        DELNI   LNI_A, , <Connected to segment B; In the Computer room>

        SEGMENT Sa,  , <Ethernet segment A>
        SEGMENT Sb,  , <Ethernet segment B>

        NET_CLOUD       BRIDGES, , <Bridging between ethernet segments A and B>
```

(continued on next page)

Example E–1 (Cont.) Portion of LAVC$FAILURE_ANALYSIS.MAR to Edit

```
;       Edit 5.❺
;
;               Describe the network connections.
;
        CONNECTION      Sa,     MPR_A
        CONNECTION              MPR_A,  A1
        CONNECTION                      A1,     A
        CONNECTION              MPR_A,  B1
        CONNECTION                      B1,     B

        CONNECTION      Sa,     D1
        CONNECTION              D1,     D

        CONNECTION      Sa,     BRIDGES
        CONNECTION      Sb,     BRIDGES

        CONNECTION      Sb,     LNI_A
        CONNECTION              LNI_A,  A2
        CONNECTION                      A2,     A
        CONNECTION              LNI_A,  B2
        CONNECTION                      B2,     B

        CONNECTION      Sb,     D2
        CONNECTION              D2,     D

        .PAGE

;       *** End of edits ***
```

In the program, Edit *number* identifies a place where you must edit the program to incorporate information about your network. Make the following edits to the program:

❶ At Edit 1 in the template, define a category for each component in the configuration. Use the information from step 5 in Section E.2.1. Use the following format:

NEW_COMPONENT component_type category

In the following example, a DEMPR repeater is defined as part of the Component category:

```
NEW_COMPONENT     DEMPR     COMPONENT
```

❷ At Edit 2, draw the network map you drew for step 1 of Section E.2.1. Including the map here in LAVC$FAILURE_ANALYSIS.MAR gives you an electronic record of the map that you can locate and update more easily than a drawing on paper.

❸ At Edit 3, list each VMScluster node and its LAN adapters. Use one line for each node. Each line should include the following information. Separate the items of information with commas to create a table of the information.

- Component type, followed by a comma.
- Label from the network map, followed by a comma.
- Node name (for SYSTEM components only). If there is no node name, enter a comma.
- Descriptive text that the network failure analysis program displays if it detects a failure with this component. Put this text within angle brackets (< >). This text should include the component's physical location.

- LAN hardware address (for LAN adapters).
- DECnet LAN address for the LAN adapter that DECnet uses.

❹ At Edit 4, list each of the other network components. Use one line for each component. Each line should include the following information:

- Component name and category you defined with NEW_COMPONENT.
- Label from the network map.
- Descriptive text that the network failure analysis program displays if it detects a failure with this component. Include a description of the physical location of the component.
- LAN hardware address (optional).
- Alternate LAN address (optional).

❺ At Edit 5, define the connections between the network components. Use the CONNECTION macro and the labels for the two components that are connected. Include the following information:

- CONNECTION macro name
- First component label
- Second component label

E.2.3 Assembling and Linking the Failure Analysis Program

Use the following command procedure to assemble and link the program:

```
$ @LAVC$BUILD.COM LAVC$FAILURE_ANALYSIS.MAR
```

Make the edits necessary to fix the assembly or link errors, such as errors caused by mistyping component labels in the path description. Assemble the program again.

E.2.4 Executing the Network Failure Analysis Program

Before you execute LAVC$FAILURE_ANALYSIS.EXE, modify the startup files in SYS$COMMON:[SYSMGR] to add a conditional statement to run LAVC$FAILURE_ANALYSIS.EXE only on the node for which you supplied data. The following is an example of such a conditional statement:

```
$ If F$GETSYI ("nodename").EQS."OMEGA"
$ THEN
$   RUN SYS$MANAGER:LAVC$FAILURE_ANALYSIS.EXE
$ ENDIF
```

Note

You must have the PHY_IO privilege in order to execute LAVC$FAILURE_ANALYSIS.EXE.

Execute the linked program on one of the nodes that will perform the network failure analysis. Use an account that has the PHY_IO privilege. Then execute the program on each of the nodes that will perform the network failure analysis.

After it executes, the program displays the approximate amount of nonpaged pool required for the network description. The display is similar to the following:

```
Non-paged Pool Usage:  ~ 10004 bytes
```

On each system running the network failure analysis, modify the file SYS$SPECIFIC:[SYSEXE]MODPARAMS.DAT to include the following lines, where *value* is the value displayed for nonpaged pool usage:

```
ADD_NPAGEDYN = value
ADD_NPAGEVIR = value
```

Run AUTOGEN on each system for which you modified MODPARAMS.DAT.

E.2.5 Testing the Network Failure Analysis Subsystem

Test the program by causing a failure. For example, disconnect a transceiver cable or ThinWire segment, or cause a power failure on a bridge, a DELNI interconnect, or a DEMPR repeater. Then check the OPCOM messages to see whether LAVC$FAILURE_ANALYSIS reports the failed component correctly. If it does not report the failure, check your edits to the network failure analysis program.

E.2.6 PEDRIVER Suspect Network Component Display

When a VMScluster network component failure occurs, OPCOM displays a list of suspected components. Displaying the list through OPCOM allows the system manager to enable and disable selectively the display of these messages.

The following are sample displays:

```
%%%%%%%%%%  OPCOM   1-AUG-1991 14:16:13.30  %%%%%%%%%%    (from node BETA at
1-AUG -1991 14:15:55.38) Message from user SYSTEM on BETA
LAVC-W-PSUSPECT, component_name

%%%%%%%%%%  OPCOM   1-AUG-1991 14:16:13.41  %%%%%%%%%%    (from node BETA at
1-AUG-1991 14:15:55.49) Message from user SYSTEM on BETA
%LAVC-W-PSUSPECT, component_name

%%%%%%%%%%  OPCOM   1-AUG-1991 14:16:13.50  %%%%%%%%%%    (from node BETA at
1-AUG-1991 14:15:55.58) Message from user SYSTEM on BETA
%LAVC-I-ASUSPECT, component_name
```

The OPCOM display of suspected failures uses the following prefixes to list suspected failures:

- %LAVC-W-PSUSPECT—Primary suspects
- %LAVC-I-ASUSPECT—Secondary or additional suspects
- %LAVC-S-WORKING—Suspect component is now working

The text following the message prefix is the description of the network component you supplied when you edited LAVC$FAILURE_ANALYSIS.MAR.

F

Local Area VMScluster Subroutine Package

The subroutines (described in Appendix E) are provided to control the features of the local area VMScluster. The sample programs LAVC$FAILURE_ANALYSIS.MAR, LAVC$START_BUS.MAR, and LAVC$STOP_BUS.MAR use this subroutine package, and these programs may be sufficient for your needs. The subroutine package provides a way of extending the sample programs as your needs require.

This subroutine package offers the following function calls to manage LAN adapters:

- SYS$LAVC_START_BUS—Directs PEDRIVER to start the local area VMScluster protocol on a specified LAN adapter
- SYS$LAVC_STOP_BUS—Directs PEDRIVER to stop the local area VMScluster protocol on a specified LAN adapter

The subroutine package also offers the following function calls to control the network failure analysis subsystem:

- SYS$LAVC_DEFINE_NET_COMPONENT—Creates a representation of a physical network component
- SYS$LAVC_DEFINE_NET_PATH—Creates a directed list of network components between two network nodes
- SYS$LAVC_ENABLE_ANALYSIS—Enables the network failure analysis, which makes it possible to analyze future channel failures
- SYS$LAVC_DISABLE_ANALYSIS—Stops the network failure analysis and deallocates the memory used for the physical network description

F.1 Subroutine Package to Start the Protocol on a LAN Adapter

The SYS$LAVC_START_BUS routine starts the local area VMScluster protocol on a specified LAN adapter.

To use the routine SYS$LAVC_START_BUS, specify the following parameter:

- BUS_NAME: String descriptor representing the LAN adapter name buffer, passed by reference. The LAN adapter name must consist of 15 characters or fewer.

The following Fortran sample program uses the SYS$LAVC_START_BUS call to start the local area VMScluster protocol on the LAN adapter XQA:

```
PROGRAM START_BUS

EXTERNAL SYS$LAVC_START_BUS
INTEGER*4 SYS$LAVC_START_BUS
INTEGER*4 STATUS

STATUS = SYS$LAVC_START_BUS ( 'XQA0:' )

CALL SYS$EXIT ( %VAL ( STATUS ))

END
```

The SYS$LAVC_START_BUS call returns a status value in register R0. A success status indicates that PEDRIVER is attempting to start the local area VMScluster protocol on the specified adapter. A failure status indicates that PEDRIVER cannot start the protocol on the specified LAN adapter.

SYS$LAVC_START_BUS can return the following errors:

- SS$_ACCVIO: This status is returned for the following conditions:
  - No access to the argument list
  - No access to the LAN adapter name buffer descriptor
  - No access to the LAN adapter name buffer
- SS$_DEVACTIVE: Bus already exists. PEDRIVER is already trying to use this LAN adapter for the local area VMScluster protocol.
- SS$_INSFARG: Not enough arguments supplied
- SS$_INSFMEM: Insufficient nonpaged pool to create the bus data structure
- SS$_INVBUSNAM: Invalid bus name specified. The device specified does not represent a LAN adapter that can be used for the protocol.
- SS$_IVBUFLEN: This status value is returned under the following conditions:
  - The LAN adapter name contains no characters (length = 0).
  - The LAN adapter name contains more than 15 characters.
- SS$_NOSUCHDEV: This status value is returned under the following conditions:
  - The LAN adapter name specified does not correspond to a LAN device available to PEDRIVER on this system.
  - No LAN drivers are loaded in this system; the value for NET$AR_LAN_VECTOR is 0.
  - PEDRIVER is not initialized; PEDRIVER's PORT structure is not available.

  Note that by calling this routine, an error log message is generated.
- SS$_NOTNETDEV: PEDRIVER does not support the specified LAN device.
- SS$_SYSVERDIF: The specified LAN device's driver does not support the VCI interface version required by PEDRIVER.
- PEDRIVER can return additional errors that indicate it has failed to create the connection to the specified LAN adapter.

F.2 Subroutine Package to Stop the Protocol on a LAN Adapter

The SYS$LAVC_STOP_BUS routine stops the local area VMScluster protocol on a specified LAN adapter.

Caution

Stopping the local area VMScluster protocol on all LAN adapters will cause the local area VMScluster satellites to hang and could cause cluster systems to crash with a CLUEXIT bugcheck.

Note

When the LAVC$STOP_BUS module executes successfully, the following device-attention entry is written to the system error log:

```
DEVICE ATTENTION . . .
NI-SCS SUB-SYSTEM . . .
FATAL ERROR DETECTED BY DATALINK . . .
```

In addition, the following hexidecimal values are written to the STATUS field of the entry:

First longword (00000001)
Second longword (00001201)

This error log entry indicates expected behavior and can be ignored. However, if the first longword of the STATUS field contains a value other than hexidecimal value 00000001, an error has occurred and further investigation may be necessary.

To use this routine, specify the following parameter:

- BUS_NAME: String descriptor representing the LAN adapter name buffer, passed by reference. The LAN adapter name must contain 15 characters or fewer.

The following Fortran sample program shows the SYS$LAVC_STOP_BUS call used to stop the local area VMScluster protocol on the LAN adapter XQB:

```
PROGRAM STOP_BUS

EXTERNAL SYS$LAVC_STOP_BUS
INTEGER*4 SYS$LAVC_STOP_BUS
INTEGER*4 STATUS

STATUS = SYS$LAVC_STOP_BUS ( 'XQB' )

CALL SYS$EXIT ( %VAL ( STATUS ))

END
```

The SYS$LAVC_STOP_BUS call returns a status value in register R0. A success status indicates that PEDRIVER is attempting to shut down the local area VMScluster protocol on the specified adapter. A failure status indicates that PEDRIVER cannot shut down the protocol on the specified LAN adapter. However, PEDRIVER performs the shutdown asynchronously, and there could be other reasons why PEDRIVER is unable to complete the shutdown.

SYS$LAVC_STOP_BUS can return the following errors:

- SS$_ACCVIO: This status is returned for the following conditions:
  - No access to the argument list
  - No access to the LAN adapter name buffer descriptor
  - No access to the LAN adapter name buffer
- SS$_INVBUSNAM: Invalid bus name specified. The device specified does not represent a LAN adapter that can be used for the local area VMScluster protocol.
- SS$_IVBUFLEN: This status value is returned under the following conditions:
  - The LAN adapter name contains no characters (length = 0).
  - The LAN adapter name has more than 15 characters.
- SS$_NOSUCHDEV: This status value is returned under the following conditions:
  - The LAN adapter name specified does not correspond to a LAN device that is available to PEDRIVER on this system.
  - No LAN drivers are loaded in this system. NET$AR_LAN_VECTOR is zero.
  - PEDRIVER is not initialized. PEDRIVER's PORT structure is not available.

F.3 Subroutine Package to Create a Network Component

The SYS$LAVC_DEFINE_NET_COMPONENT subroutine call creates a representation for a physical network component.

Specify the following parameters for SYS$LAVC_DEFINE_NET_COMPONENT:

- component_description: Address of a string descriptor representing network component name buffer. The length of the network component name must be less than or equal to the number of COMP$C_MAX_NAME_LEN characters.
- nodename_length: Address of the length of the node name. This address is located at the beginning of the network component name buffer for COMP$C_NODE types. You should use zero for other component types.
- component_type: Address of the component type. These values are defined by $PEMCOMPDEF found in SYS$LIBRARY:LIB.MLB.
- lan_hardware_addr: String descriptor of a buffer containing the component's LAN hardware address (6 bytes). You must specify this value for COMP$C_ADAPTER types. For other component types, this value is optional.
- lan_decnet_addr: String descriptor of a buffer containing the component's LAN DECnet address (6 bytes). This is an optional parameter for all component types.
- component_id_value: Address of a longword that is written with the component ID value.

Use the following format to specify the parameters:

```
STATUS = SYS$LAVC_DEFINE_NET_COMPONENT (
                    component_description,
                    nodename_length,
                    component_type,
                    lan_hardware_addr,
                    lan_decnet_addr,
                    component_id_value )
```

This subroutine call defines a network component. If successful, this call creates a COMP data structure and returns its ID value. This call copies user-specified parameters into the data structure and sets the reference count to zero.

This call also returns a status as well as the component ID value. The component ID value is a 32-bit value that has a one-to-one association with a network component. Lists of these component IDs are passed to SYS$LAVC_DEFINE_NET_PATH to specify the components used when a packet travels from one node to another.

SYS$LAVC_DEFINE_NET_COMPONENT can return the following errors:

- SS$_ACCVIO—This status is returned under the following conditions:
  - No access to the network component name buffer descriptor
  - No access to the network component name buffer
  - No access to the component's LAN hardware address if a nonzero value was specified
  - No access to the component's LAN DECnet address if a nonzero value was specified
  - No access to the lan_hardware_addr string descriptor
  - No access to the lan_decnet_addr string descriptor
  - No write access to the component_id_value address
  - No access to the component_type address
  - No access to the nodename_length address
  - No access to the argument list
- SS$_DEVACTIVE: Analysis program already running. You must stop the analysis by calling the SYS$LAVC_DISABLE_ANALYSIS before you define the network components and the network component lists.
- SS$_INSFARG: Not enough arguments supplied.
- SS$_INVCOMPTYPE: The component type is either 0 or greater than or equal to COMP$C_INVALID.
- SS$_IVBUFLEN: This status value is returned under the following conditions:
  - The component name has no characters (length = 0).
  - Length of the component name is greater than COMP$C_MAX_NAME_LEN.
  - The node name has no characters (length = 0) and the component type is COMP$C_NODE.

- The node name has more than 8 characters and the component type is COMP$C_NODE.
- The lan_hardware_addr string descriptor has fewer than 6 characters.
- The lan_decnet_addr has fewer than 6 characters.

F.4 Subroutine Package to Create a Network Component List

The SYS$LAVC_DEFINE_NET_PATH subroutine call creates a directed list of network components between two network nodes. A **directed list** is a list of all the components through which a packet passes as it travels from the failure analysis node to other nodes in the cluster network.

Specify the following parameters for SYS$LAVC_DEFINE_NET_PATH:

- network_component_list: Address of a string descriptor for a buffer containing the component ID values for each of the components in the path. List the component ID values in the order in which a network message travels through them. Specify components in the following order:
  1. The local node
  2. The local LAN adapter
  3. Intermediate network components
  4. The remote network LAN adapter
  5. The remote node

  You must list two nodes and two LAN adapters in the network path. The buffer length must be greater than 15 bytes and less than 509 bytes.
- used_for_analysis_status: Address of a longword status value that is written. This status indicates whether this network path has any value for the network failure analysis.
- bad_component_id: Address of a longword value that contains the component ID that is in error if an error is detected while processing the component list.

Use the following format to specify the parameters:

```
STATUS = SYS$LAVC_DEFINE_NET_PATH (
                    network_component_list,
                    used_for_analysis_status,
                    bad_component_id )
```

This subroutine call creates a directed list of network components that describe a specific network path. This call, if successful, creates a CLST data structure. If one node is the local node, then this data structure is associated with a PEDRIVER channel. In addition, the reference count for each network component in the list is incremented. If neither node is the local node, then the used_for_analysis_status address contains an error status.

This call returns a status in register R0 indicating that the network component list has the correct construction. If this value is successful, the used_for_analysis_status value indicates whether the network path is useful for network analysis performed on the local node.

If a failure status returned in R0 is SS$_INVCOMPID, the bad_component_id address contains the value of the bad_component_id found in the buffer.

The following errors can be returned:

- SS$_ACCVIO: This status value can be returned under the following conditions:
  - No access to the descriptor or the network component ID value buffer
  - No access to the argument list
  - No write access to used_for_analysis_status address
  - No write access to bad_component_id address
- SS$_DEVACTIVE: Analysis already running. You must stop the analysis by calling the SYS$LAVC_DISABLE_ANALYSIS function before defining the network components and the network component lists.
- SS$_INSFARG: Not enough arguments supplied
- SS$_INVCOMPID: Invalid network component ID specified in the buffer. The bad_component_id address contains the failed component ID.
- SS$_INVCOMPLIST: This status value can be returned under the following conditions:
  - Fewer than two nodes were specified in the node list.
  - More than two nodes were specified in the list.
  - The first network component ID was not a COMP$C_NODE type.
  - The last network component ID was not a COMP$C_NODE type.
  - Fewer than two adapters were specified in the list.
  - More than two adapters were specified in the list.
- SS$_IVBUFLEN: Length of the network component ID buffer is less than 16, is not a multiple of 4, or is greater than 508.
- SS$_RMTPATH: Network path is not associated with the local node. This status is returned only to indicate whether this path was needed for network failure analysis on the local node.

F.5 Subroutine Package to Start the Network Component Failure Analysis

The SYS$LAVC_ENABLE_ANALYSIS subroutine call starts the network component failure analysis. The following is an example of using the call:

```
STATUS = SYS$LAVC_ENABLE_ANALYSIS ( )
```

This subroutine call attempts to enable the network component failure analysis code. The attempt will succeed if at least one component list is defined.

This call returns a status in register R0.

The following errors can be returned:

- SS$_DEVOFFLINE: PEDRIVER is not properly initialized. ROOT or PORT block is not available.
- SS$_NOCOMPLSTS: No network connection lists exist. Network analysis is not possible.
- SS$_WASSET: Network component analysis is already running.

F.6 Subroutine Package to Stop the Network Component Failure Analysis

The SYS$LAVC_DISABLE_ANALYSIS subroutine call stops the network component failure analysis. The following is an example of using the call:

```
STATUS = SYS$LAVC_DISABLE_ANALYSIS ( )
```

This subroutine call disables the network component failure analysis code and, if analysis was enabled, deletes all the network component definitions and network component list data structures from nonpaged pool.

SYS$LAVC_DISABLE_ANALYSIS returns a status in register R0.

The following errors can be returned:

- SS$_DEVOFFLINE: PEDRIVER is not properly initialized. ROOT or PORT block is not available.
- SS$_WASCLR: Network component analysis already stopped.

G

Troubleshooting the NISCA Protocol

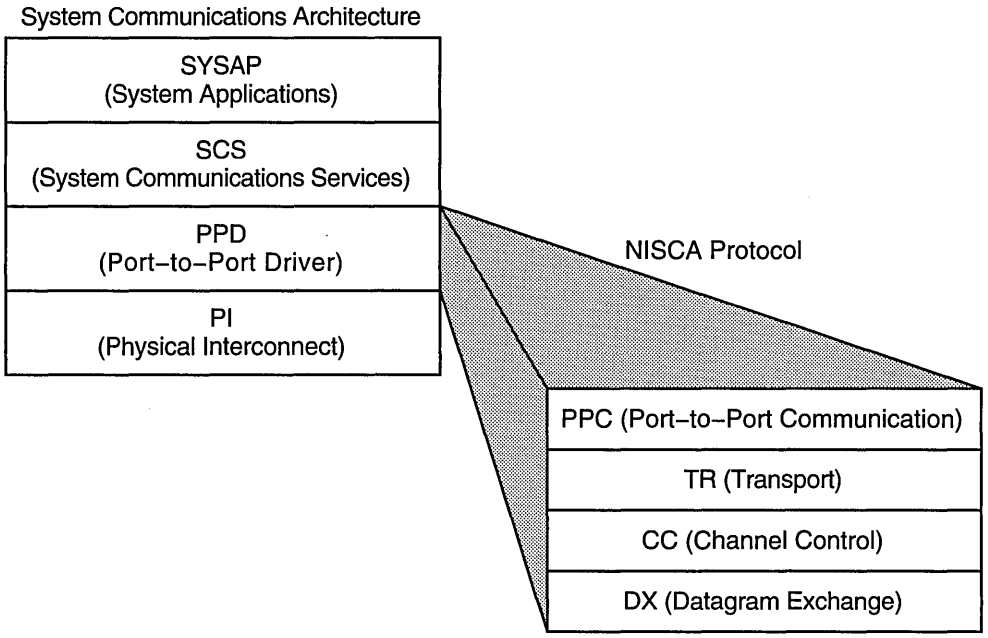
NISCA is the transport protocol responsible for carrying messages, such as disk I/Os and lock messages, across Ethernet and FDDI LANs to other nodes in the cluster. In this appendix, the acronym NISCA refers to the protocol that implements an Ethernet or FDDI network interconnect (NI) according to the System Communications Architecture (SCA).

Problems with VMScluster LAN communications can be difficult to troubleshoot. A number of hardware, software, and network factors come into play so that diagnosis based on problem symptoms alone might not be adequate. This appendix provides a strategy to help you analyze and troubleshoot VMScluster LAN communication problems that are specific to the NISCA protocol.

G.1 Overview of the NISCA Protocol

The SCA specifies a number of protocols for VMScluster systems, including System Applications (SYSAP), System Communications Services (SCS), the Port-to-Port Driver (PPD), and the Physical Interconnect (PI) of the device driver and LAN adapter. Figure G–1 shows these protocols as interdependent levels that make up the SCA architecture. Figure G–1 shows how the PPD level, which contains the NISCA protocol, fits into the SCA architecture.

Figure G–1 How the NISCA Protocol Fits Into the SCA Architecture



The following list describes the levels of the SCA protocol shown in Figure G–1:

- SYSAP represents clusterwide system applications that execute on each node. These system applications share communication paths in order to send messages between nodes. Examples of system applications are disk class drivers (such as DUDRIVER), the MSCP server, and the connection manager.
- SCS manages connections around the VMScluster and multiplexes messages between system applications over a common transport called a **virtual circuit**. This level also notifies individual system applications when a connection fails so that they can respond appropriately. For example, an SCS notification might trigger DUDRIVER to fail over a disk, trigger a cluster state transition, or notify the connection manager to start timing reconnect (RECNXINTERVAL) intervals.
- PPD executes the NISCA protocol. It consists of four interdependent levels:
  - Port-to-Port Communication (PPC) level—Provides port-to-port communication, datagrams, sequenced messages, and block transfers. "Segmentation" also occurs at the PPC level. During segmentation of large blocks of data, the data transfers are mapped differently on a LAN than they are on a CI or a DSSI bus. LAN data packets are fragmented according to the size allowed by the particular LAN communications path, as follows:
    - Ethernet-to-Ethernet communications allow 1498 bytes per packet.
    - FDDI-to-Ethernet communications allow 1498 bytes per packet.
    - FDDI-to-Ethernet-to-FDDI communications allow 1498 bytes per packet.
    - FDDI-to-FDDI communications allow 4468 bytes per packet.

  - Transport (TR) level—Provides an error-free path, called a virtual circuit, between nodes. The PPC level uses a virtual circuit for transporting sequenced messages and datagrams between two nodes in the cluster.
  - Channel Control (CC) level—Manages network paths, called channels, between nodes in a VMScluster. The CC level maintains channels by sending HELLO datagram messages between nodes. A node sends a HELLO datagram messages to indicate it is still functioning and an active member of the VMScluster. The TR level uses channels for sequenced message and datagram traffic.
  - Datagram Exchange (DX) level—Interfaces to the LAN driver.
- PI provides connections to LAN devices. PI consists of:
  - LAN drivers—Multiplex NISCA and many other clients (such as DECnet, TCP/IP, LAT, LAD/LAST) and provide them with datagram services on Ethernet and FDDI network interfaces.
  - LAN adapters—Consist of the LAN network driver and adapter hardware.

The port emulator driver, PEDRIVER, implements the NISCA protocol and establishes and controls communication paths between local and remote LAN ports. The communication paths, called **channels**, are determined by the pairs of adapters and the connecting network. For example, two nodes, each having two adapters, could establish four channels.

The difference between a channel and a virtual circuit is that channels provide a path for datagram service. Virtual circuits, layered on channels, provide an error-free path between nodes. Multiple channels can exist between nodes in a VMScluster, but only one virtual circuit can exist between any two nodes at a time.

PEDRIVER implements a packet delivery service (at the TR level of the NISCA protocol) that guarantees the sequential delivery of messages. The messages carried by a particular virtual circuit can be sent over any of the channels connecting the two nodes. The choice of channel is determined by the sender (PEDRIVER) of the message. Because a node sending a message can choose any channel, PEDRIVER, as a receiver, must be prepared to receive messages over any channel.

At any point in time the TR level makes use of a single "preferred channel" to carry the traffic for a particular virtual circuit. The TR level can modify its choice of a prefered channel at any time, based on the following:

- Minimum measured delay

  The NISCA protocol routinely measures response time to messages and uses these measurements to pick the most lightly loaded channel on which to send messages.

- Maximum packet size

  PEDRIVER favors channels with large packet sizes. For example, an FDDI-to-FDDI channel is favored over an FDDI-to-Ethernet channel or an Ethernet-to-Ethernet channel. if your configuration uses FDDI to Ethernet bridges, the PPC level of the NISCA protocol segments messages into the smaller packet sizes allowed by Ethernet before transmitting them.

The remainder of this appendix describes the NISCA transport protocol as it is implemented by PEDRIVER and provides techniques for troubleshooting communication problems.

G.2 Addressing LAN Problems Specific to the Local Node

Communication trouble in VMScluster systems may be indicated by symptoms such as the following:

- Poor performance
- Console messages
  - "Virtual circuit closed" messages on the console
  - "Connection loss" OPCOM messages on the console
  - CLUEXIT bugcheck messages
- Repeated loss of a virtual circuit or multiple virtual circuits over a short period of time (fewer than 10 minutes)

Before you initiate complex diagnostic procedures, do not overlook the obvious. Always make sure the hardware is configured and connected properly and that the network is started. Also, make sure system parameters are set correctly on all nodes in the VMScluster.

Identifying Problems Related to LAN Traffic

Keep in mind that a VMScluster system generates substantially heavier traffic than other LAN protocols. In many cases, cluster behavior problems that appear to be related to the network might actually be related to software, hardware, or user errors. For example, a large amount of traffic does not necessarily indicate a problem with the VMScluster network. The amount of traffic generated depends on how the users utilize the system and the way that the VMScluster is configured with additional interconnects (such as DSSI and CI).

If the amount of traffic generated by the VMScluster exceeds the expected or desirable levels, then you might be able to reduce the level of traffic by:

- Adding DSSI or CI interconnects
- Shifting the user load between machines
- Adding LAN segments and reconfiguring the LAN connections across the VMScluster system

Preliminary Diagnosis of LAN Communication Failures

If the symptoms and preliminary diagnosis indicate that you might have a network problem, troubleshooting LAN communication failures should start with the step-by-step procedures described in Appendix C. Appendix C helps you diagnose and solve common Ethernet and FDDI LAN communication failures during the following stages of VMScluster activity:

- When a computer or a satellite fails to boot
- When a computer fails to join the VMScluster
- During run time when startup procedures fail to complete
- When a VMScluster hangs

The procedures in Appendix C require that you verify a number of parameters during the diagnostic process. Because system parameter settings play a key role in effective VMScluster communications, Section G.2.1 describes several system parameters that are especially important to the timing of LAN bridges, disk failover, and channel availability.

Diagnosing Problems Related to PEDRIVER

Because PEDRIVER communication is based on channels, LAN network problems typically fall into these areas:

- Channel formation and maintenance

  Channels are formed when HELLO datagram messages are received from a remote system. A failure can occur when the HELLO datagram messages are not received or when the channel control message contains the wrong data.

- Retransmission

  A well-configured VMScluster system should not perform excessive retransmissions between nodes. Retransmissions between any nodes that occur more frequently than once every few seconds deserve network investigation.

Diagnosing failures at this level becomes more complex because the errors are usually intermittent. Moreover, even though PEDRIVER is aware when a channel is unavailable and performs error recovery based on this information, it does not provide notification when a channel failure occurs; PEDRIVER provides notification only for virtual circuit failures.

However, the Local Area VMScluster Network Failure Analysis program, available in SYS$EXAMPLES, can help you use PEDRIVER information about channel status. You can use the Local Area VMScluster Network Failure Analysis Program that is documented in Appendix E to analyze long-term channel outages, such as hard failures in LAN network components that occur during run time.

This program uses tables in which you describe your LAN hardware configuration. During a channel failure, PEDRIVER uses the hardware configuration represented in the table to isolate which component might be causing the failure. PEDRIVER reports the suspected component through an OPCOM display. You can then isolate the LAN component for repair or replacement.

Section G.4 addresses the kinds of problems you might find in the NISCA level of LAN communications and provides methods for diagnosing and solving them.

G.2.1 Checking System Timing Parameters

Several system parameters are relevant to the recovery and failover time limits for LANs in a VMScluster:

- The RECNXINTERVAL system parameter defines the amount of time to wait before removing a node from the VMScluster after detection of a virtual circuit failure, which could result from a LAN bridge failure. If your network uses multiple paths and you want the VMScluster to survive failover between LAN bridges, make sure the value of RECNXINTERVAL is greater than the time it takes to fail over those paths. The formula for calculating this parameter is discussed in Section 2.9.4.

- The MVTIMEOUT system parameter defines the amount of time the OpenVMS operating system tries to recover a path to a disk before returning failure messages to the application. This parameter is relevant when the VMScluster configuration is set up to serve disks over either the Ethernet or FDDI.
- The SHADOW_MBR_TIMEOUT system parameter defines the amount of time that the Volume Shadowing for OpenVMS tries to recover from a transient disk error on a single member of a multiple-member shadow set. This parameter differs from MVTIMEOUT because you should set SHADOW_MBR_TIMEOUT to remove a failing shadow set member quickly. This is because the remaining members can recover more rapidly once the failing member is removed.

In addition, channel timeouts are detected by PEDRIVER. PEDRIVER listens for HELLO datagram messages, which are sent over channels every 1.6 to 3 seconds. Every node in the VMScluster multicasts HELLO datagram messages on each LAN adapter to notify other nodes that the channel is still available and that the node is still a functioning member of the cluster.

PEDRIVER closes a channel when a HELLO datagram message has not been received for a period of 8 to 9 seconds. Because HELLO datagram messages are transmitted every 1.6 to 3 seconds, PEDRIVER times out a channel only if at least two HELLO datagram messages are lost.

PEDRIVER closes a virtual circuit if there are no channels available or if the packet size of an available channel is insufficient. The virtual circuit is not closed if any other channels to the node are still available except when the packet size of an available channel is smaller than the channel being closed. For example, if a channel fail over from FDDI to Ethernet, PEDRIVER closes the virtual circuit and then reopens it after negotiating the smaller packet size that is necessary for Ethernet segmentation.

Errors are not reported when a channel is closed. OPCOM "Connection loss" errors or SYSAP messages are not sent to users or other system applications until after the virtual circuit shuts down. This fact is significant, especially if there are multiple paths to a node and a LAN hardware failure occurs. In this case, you might not receive an error message; PEDRIVER continues to use the virtual circuit over another available channel.

A closed virtual circuit is reestablished when a channel becomes available again. PEDRIVER reopens a channel when HELLO datagram messages are received.

Note

The TIMVCFAIL system parameter, which optimizes the amount of time needed to detect a communication failure, is not recommended for use with LAN communications. This parameter is intended for CI and DSSI connections. PEDRIVER (which is for Ethernet and FDDI) usually supersedes the value set in TIMVCFAIL with the listen timeout of 8 to 9 seconds.

G.2.2 Using SDA to Monitor LAN Communications

If your system shows symptoms of intermittent failures during run time, you need to determine whether there is a network problem or the symptoms are caused by some other activity in the system.

Generally, you can diagnose problems with NISCA or the network using the OpenVMS System Dump Analyzer (SDA). SDA is an effective tool for isolating problems on specific nodes running in the VMScluster system. The following sections describe the use of some SDA commands and qualifiers. You should also refer to the *OpenVMS AXP System Dump Analyzer Utility Manual* or the *OpenVMS VAX System Dump Analyzer Utility Manual* for complete information about SDA for your system.

Monitoring Virtual Circuits

The SDA command SHOW PORT provides relevant information that is useful in troubleshooting PEDRIVER and LAN adapters in particular. You must enter the SHOW PORT command to allow SDA to define cluster symbols. Example G–1 illustrates how the SHOW PORT command provides a summary of VMScluster data structures.

Example G–1 SDA Command SHOW PORT Display

```
$ ANALYZE/SYSTEM
SDA> SHOW PORT

VAXcluster data structures
--------------------------

                --- PDT Summary Page ---

 PDT Address          Type         Device          Driver Name
 -----------          ----         -------         -----------

  80C3DBA0             pa          PAA0            PADRIVER
  80C6F7A0             pe          PEA0            PEDRIVER
```

To examine information about the virtual circuit (VC) that carries messages between the local node (where you are running SDA) and another remote node, enter the SDA command SHOW PORT/VC=VC_*remote-node-name*. Example G–2 shows how to examine information about the virtual unit running between nodes UPNVMS (the local node) and NODE11 (the remote node).

Example G–2 SDA Command SHOW PORT/VC Display

```
SDA> SHOW PORT/VC=VC_NODE11

VAXcluster data structures
--------------------------
                --- Virtual Circuit (VC) 98625380 ---
Remote System Name:  NODE11  (0:VAX)     Remote SCSSYSTEMID:  19583
Local System ID:  217 (D9)              Status: 0005 open,path
------ Transmit -------  ----- VC Closures -----  ❼--- Congestion Control ----
Msg Xmt❶      46193196  SeqMsg TMO            0  Pipe Quota/Slo/Max❽ 31/ 7/31
  Unsequence         3  CC DFQ Empty          0  Pipe Quota Reached❾   213481
  Sequence    41973703  Topology Change❺      0  Xmt C/T❿             0/1984
  ReXmt❷        128/106  NPAGEDYN Low❻        0  RndTrp uS⓫       18540+7764
  Lone ACK      4219362                          UnAcked Msgs              0
Bytes Xmt    137312089                          CMD Queue Len/Max       0/21
------- Receive -------  - Messages  Discarded -  ----- Channel Selection -----
Msg Rcv❸      47612604  No Xmt Chan           0  Preferred Channel    9867F400
  Unsequence         3  Rcv Short Msg         0  Delay Time           FAAD63E0
  Sequence    37877271  Illegal Seq Msg       0  Buffer Size              1424
  ReRcv❹          13987  Bad Checksum          0  Channel Count              18
  Lone ACK     9721030  TR DFQ Empty          0  Channel Selections      32138
  Cache            314  TR MFQ Empty          0  Protocol                1.3.0
  Ill ACK            0  CC MFQ Empty          0  Open⓬ 8-FEB-1993 17:00:05.12
Bytes Rcv   3821742649  Cache Miss            0  Cls⓭ 17-NOV-1858 00:00:00.00
```

The SHOW PORT/VC=VC_*remote-node-name* command displays a number of performance statistics about the virtual circuit for the target node. The display groups the statistics into general categories that summarize such things as packet transmissions to the remote node, packets received from the remote node, and congestion control behavior. The statistics most useful for problem isolation are called out in Example G–2 and described in the following list.

Note

The counters shown in Example G–2 represent fixed-size fields that are automatically reset to 0 when a field reaches its maximum size (or when the system is rebooted). Because each field has a different maximum size and a different rate of growth, the field counters are likely to reset at different times. Thus, for a system that has been running for a long time some field values may seem illogical and contradictory with others.

❶ The Msg Xmt (messages transmitted) field shows the total number of packets transmitted over the virtual circuit to the remote node, including both sequenced and unsequenced (channel control) messages, and lone acknowledgments. (All application data is carried in sequenced messages.) The counters for sequenced messages and lone acknowledgments grow more quickly than most other fields.

❷ The ReXmt (retransmission) field indicates the number of retransmissions for the virtual circuit. A retransmission occurs when the local node does not receive an acknowledgment for a transmitted packet within a predetermined timeout interval. A timeout indicates one of the following problems:

- The remote system NODE11 did not receive the sequenced message sent by UPNVMS.
- The sequenced message arrived but was delayed in transit to NODE11.

- The local system UPNVMS did not receive the acknowledgment to the message sent to remote node NODE11.
- The acknowledgment arrived but was delayed in transit from NODE11.

Congestion either in the network or at one of the nodes can cause the following problems:

- Congestion in the network can result in delay or lost packets. Network hardware problems can also result in lost packets.
- Congestion in UPNVMS or NODE11 can result either in packet delay because of queuing in the adapter or in packet discard because of insufficient buffer space.

The rightmost number (106) in the ReXmt field indicates the number of times a timeout occured. The leftmost number (128) indicates the number of packets actually retransmitted. For example, if the network loses two packets at the same time, one timeout is counted but two packets are retransmitted.

Although you should expect to see a certain number of retransmissions (especially in heavily loaded networks), an excessive number of retransmissions wastes network bandwidth and indicate excessive load or intermittent hardware failure. If the leftmost value in the ReXmt field is greater than about 0.01% to 0.05% of the total number of the transmitted messages shown in the Msg Xmt field, the VMScluster system probably is experiencing excessive network problems or local loss from congestion.

❸ The Msg Rcv (messages received) field indicates the total number of messages received by local node UPNVMS over this virtual circuit. The values for sequenced messages and lone acknowledgments usually increase at a rapid rate.

❹ The ReRcv (rereceive) field displays the number of packets redundantly received by this system. A remote system may retransmit packets even though the local node has already successfully received them. This happens when the cumulative delay of the packet and its acknowledgment is longer than the estimated round trip time being used as a timeout value by the remote node. Therefore, the remote node retransmits the packet even though it is unnnecessary.

Underestimation of the round-trip delay by the remote node is not directly harmful, but the retransmission and subsequent congestion-control behavior on the remote node have a detrimental effect on data throughput. Large numbers indicate frequent bursts of congestion in the network or adapters leading to excessive delays. If the value in the ReRcv field is greater than approximately 0.01% to 0.05% of the total messages received, there may be a problem with congestion or network delays.

❺ The Topology Change field is displayed in the "VC Closures" section. (VC stands for virtual circuit.) The Topology Change field indicates the number of times PEDRIVER has performed a failover from FDDI to Ethernet, which necessitates closing and reopening the virtual circuit. In Example G–2, there have been no failovers. However, if the field indicates a number of failovers, a problem may exist on the FDDI ring.

❻ The NPAGEDYN (nonpaged dynamic pool) field displays the number of times the virtual circuit was lost because of a pool allocation failure on the local node. You probably need to increase the value of the NPAGEDYN system parameter on the local node.

❼ The Congestion Control counters display information about the virtual circuit to control the pipe quota (the number of messages that can be sent to the remote node [put into the "pipe"] before receving an acknowledgment and the retransmission timeout). PEDRIVER varies the pipe quota and the timeout value to control the amount of network congestion.

❽ The Pipe Quota/Slo/Max field in the third column indicates the current thresholds governing the pipe quota. The leftmost number (31) is the current value of the pipe quota (transmit window). After a timeout, the pipe quota is reset to 1 to decrease congestion and is allowed to increase quickly as acknowledgments are received. The middle number (7) is the slow-growth threshold (the size at which the rate of increase is slowed) to avoid congestion on the network again. The rightmost number (31) is the maximum value currently allowed for the VC based on channel limitations.

❾ The Pipe Quota Reached field indicates the number of times the entire transmit window was full. If this number is small as compared with the number of sequenced messages transmitted, it indicates that the local node is not sending large bursts of data to the remote node.

❿ The Xmt C/T (transmission count/target) field shows both the number of successful transmissions since the last time the pipe quota was increased and the target value at which the pipe quota is allowed to increase. In the example, the count is 0 because the pipe quota is already at its maximum value (31), so successful transmissions are not being counted.

⓫ The RndTrp uS (round trip in microseconds) field displays values that are used to calculate the retransmission timeout in microseconds. The leftmost number (18540) is the average round-trip time, and the rightmost number (7764) is the average variation in round-trip time. In the example, the values indicate that the round trip is about 19 milliseconds plus or minus about 8 milliseconds.

⓬ The Channel Selection window displays open (Open) and closed (Cls) timestamps for the last significant changes in the virtual circuit. The repeated loss of one or more virtual circuits over a short period of time (fewer than 10 minutes) indicates network problems.

⓭ If you are analyzing a crash dump, you should check whether the crash dump time corresponds to the timestamp for channel closures (Cls).

G.2.2.1 Monitoring PEDRIVER Buses

The SDA command SHOW PORT/BUS=BUS_*LAN-device* command is useful for displaying the PEDRIVER representation of a LAN adapter. To PEDRIVER, a bus is the logical representation of the LAN adapter. (To list the names and addresses of buses, enter the SDA command SHOW PORT/ADDR=PE_PDT and then press the Return key twice.) Example G–3 shows a display for the LAN adapter named EXA.

Example G–3 SDA Command SHOW PORT/BUS Display

```
SDA> SHOW PORT/BUS=BUS_EXA
VAXcluster data structures
--------------------------
--- BUS: 817E02C0  (EXA)  Device: EX_DEMNA  LAN Address: AA-00-04-00-64-4F ---
                                  LAN Hardware Address: 08-00-2B-2C-20-B5
Status: 00000803 run,online❶,restart
------- Transmit ------  ------- Receive -------  ---- Structure Addresses ---
Msg Xmt        20290620  Msg Rcv        67321527  PORT Address        817E1140
  Mcast Msgs    1318437    Mcast Msgs   39773666  VCIB Addr           817E0478
  Mcast Bytes 168759936    Mcast Bytes 159660184  HELLO Message Addr  817E0508
Bytes Xmt     2821823510  Bytes Rcv    3313602089  BYE Message Addr    817E0698
Outstand I/Os          0  Buffer Size        1424  Delete BUS Rtn Adr  80C6DA46
Xmt Errors❷        15896  Rcv Ring Size        31
Last Xmt Error 0000005C         Time of Last Xmt Error❸21-JAN-1993 15:33:38.96
--- Receive Errors ----  ------ BUS Timer ------  ----- Datalink Events ------
TR Mcast Rcv           0  Handshake TMO  80C6F070  Last  7-DEC-1992 17:15:42.18
Rcv Bad SCSID          0  Listen TMO     80C6F074  Last Event          00001202
Rcv Short Msg          0  HELLO timer           3  Port Usable                1
Fail CH Alloc          0  HELLO Xmt err❹     1623  Port Unusable              0
Fail VC Alloc          0                          Address Change             1
Wrong PORT             0                          Port Restart Fail          0
```

❶ The Status line should always display a status of "online" to indicate that PEDRIVER can access its LAN adapter.

❷ The Xmt Errors (transmission errors) field indicates the number of times PEDRIVER has been unable to transmit a packet using this LAN adapter.

❸ You can compare the time shown in the Time of Last Xmt Error field with the Open and Cls times shown in the VC display in Example G–2 to determine whether the time of the LAN adapter failure is close to the time of a virtual circuit failure. Note that transmission errors at the LAN adapter bus level cause a virtual circuit breakage.

❹ The HELLO Xmt err (HELLO transmission error) field indicates how many times a message transmission failure has "dropped" a PEDRIVER HELLO datagram message. (The channel control [CC] description in Section G.1 briefly describes the purpose of HELLO datagram messages.) If many HELLO transmission errors occur, PEDRIVER on other nodes probably is timing out a channel, which could eventually result in closure of the virtual circuit.

In Example G–3, the 1623 HELLO transmission failures contributed to the high number of transmission errors (15896). Note that it is impossible to have a low number of transmission errors and a high number of HELLO transmission errors.

G.2.2.2 Monitoring LAN Adapters

Use the SDA command SHOW LAN/COUNT to display information about the LAN adapter maintained by the device driver (for example, for all protocols and not only the PEDRIVER representation of events occurring on the LAN). Example G–4 shows a sample display from the SHOW LAN/COUNT command.

Example G–4 SDA Command SHOW LAN/COUNTERS Display

```
$ ANALYZE/SYSTEM
SDA> SHOW LAN/COUNTERS

LAN Data Structures
-------------------
            -- EXA Counters Information 22-JAN-1993 11:21:19 --

Seconds since zeroed          3953329    Station failures                 0
Octets received           13962888501    Octets sent              11978817384
PDUs received               121899287    PDUs sent                   76872280
Mcast octets received      7494809802    Mcast octets sent          183142023
Mcast PDUs received          58046934    Mcast PDUs sent              1658028
Unrec indiv dest PDUs               0    PDUs sent, deferred          4608431
Unrec mcast dest PDUs               0    PDUs sent, one coll          3099649
Data overruns                       2    PDUs sent, mul coll          2439257
Unavail station buffs❶              0    Excessive collisions❷           5059
Unavail user buffers                0    Carrier check failure              0
Frame check errors                483    Short circuit failure              0
Alignment errors                10215    Open circuit failure               0
Frames too long                   142    Transmits too long                 0
Rcv data length error               0    Late collisions                14931
802E PDUs received              28546    Coll detect chk fail               0
802 PDUs received                   0    Send data length err               0
Eth PDUs received           122691742    Frame size errors                  0

LAN Data Structures
-------------------
          -- EXA Internal Counters Information 22-JAN-1993 11:22:28 --

Internal counters address    80C58257    Internal counters size            24
Number of ports                     0    Global page transmits              0
No work transmits             3303771    SVAPTE/BOFF transmits              0
Bad PTE transmits                   0    Buffer_Adr transmits               0

Fatal error count                   0    RDL errors                         0
Transmit timeouts                   0    Last fatal error                None
Restart failures                    0    Prev fatal error                None
Power failures                      0    Last error CSR              00000000
Hardware errors                     0    Fatal error code                None
Control timeouts                    0    Prev fatal error                None

Loopback sent                       0    Loopback failures                  0
System ID sent                      0    System ID failures                 0
ReqCounters sent                    0    ReqCounters failures               0

       -- EXA1 60-07 (SCA) Counters Information 22-JAN-1993 11:22:31 --

Last receive❸         22-JAN 11:22:31    Last transmit❸     22-JAN 11:22:31
Octets received           7616615830    Octets sent              2828248622
PDUs received               67375315    PDUs sent                  20331888
Mcast octets received               0    Mcast octets sent                  0
Mcast PDUs received                 0    Mcast PDUs sent                    0
Unavail user buffer                 0    Last start attempt              None
Last start done        7-DEC 17:12:29    Last start failed               None
    .
    .
    .
```

The SHOW LAN/COUNTERS display usually includes device counter information about several LAN adapters. However, for purposes of example, only one device is shown in Example G–4.

❶ The Unavail station buffs (unavailable station buffers) field records the number of times that fixed station buffers in the LAN driver were unavailable for incoming packets. The node receiving a message can lose packets when

the node does not enough LAN station buffers. (LAN buffers are used by a number of other consumers besides PEDRIVER, such as DECnet, TCP/IP, and LAT.) Packet loss because of insufficient LAN station buffers is a symptom of either LAN adapter congestion or the system's inability to reuse the existing buffers fast enough.

❷ The Excessive collisions field indicates the number of unsuccessful attempts to transmit messages on the adapter. This problem is often caused by:

- A LAN loading problem resulting from heavy traffic (70% to 80% utilization) on the specific LAN segment.
- A component called a screamer. A **screamer** is an adapter whose protocol does not adhere to Ethernet or FDDI hardware protocols. A screamer does not wait for permission to transmit packets on the adapter, thereby causing collision errors to register in this field.

If a significant number of transmissions with multiple collisions have occurred, then VMScluster performance is degraded. You might be able to improve performance either by removing some nodes from the LAN segment or by adding another LAN segment to the cluster. The overall goal is to reduce traffic on the LAN segment, thereby making more bandwidth available to the VMScluster system.

❸ The difference in the times shown in the Last receive and Last transmit message fields should not be large. Minimally, the timestamps in these fields should reflect that HELLO datagram messages are being sent across channels every 3 seconds. Large time differences might indicate:

- A hardware failure
- Whether or not the LAN driver sees the NISCA protocol as being active on a specific LAN adapter

G.3 Troubleshooting NISCA Communications

The following sections describe two likely areas of trouble for LAN networks: channel formation and retransmission. The discussions of these two problems often include references to the use of a LAN analyzer tool to isolate information in the NISCA protocol. Therefore, as you read about how to diagnose NISCA problems, you may also find it helpful to refer to Section G.4, which describes the NISCA protocol packet, and Section G.5, which describes how to choose and use a LAN network failure analyzer.

G.3.1 Channel Formation and Maintenance Problems

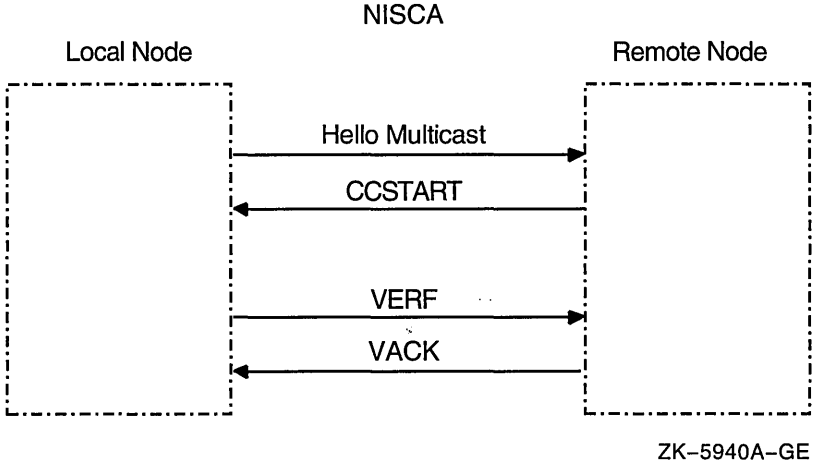
Channel formation problems occur when two nodes cannot communicate properly between LAN adapters. Channels are formed when a node sends a HELLO datagram from its LAN adapter to a LAN adapter on another cluster node. If this is a new remote LAN adapter address, or if the corresponding channel is closed, the remote node receiving the HELLO datagram sends a CCSTART datagram to the originating node after a delay of up to 2 seconds.

Upon receiving a CCSTART datagram, the originating node verifies the cluster password and, if the password is correct, the node responds with a VERF datagram and waits for up to 5 seconds for the remote node to send a VACK datagram. (VERF, VACK, CCSTART, and HELLO datagrams are described in Section G.4.3.)

Upon receiving a VERF datagram, the remote node verifies the cluster password; if the password is correct, the node responds with a VACK datagram and marks the channel as open.

Figure G–2 shows a message exchange during a successful channel formation handshake.

Figure G–2 Channel Formation Handshake



If the local node does not receive the VACK datagram within 5 seconds, the channel state goes back to closed and the handshake timeout counter is incremented. If the VACK datagram is received within 5 seconds and the cluster password is correct, the channel is opened. Once a channel has been formed, it is maintained (kept open) by the regular multicast of HELLO datagram messages. Each node multicasts a HELLO datagram message every 1.6 to 3.0 seconds over each LAN adapter. Either of the nodes sharing a channel closes the channel with a listen timeout if it does not receive a HELLO datagram message from the other node within 8 to 9 seconds. If you receive a "Port closed virtual circuit" message, it indicates a channel was formed but there is a problem receiving HELLO datagram messages on time. When this happens, look for HELLO datagram messages getting lost.

Channel formation problems occur when there is a break in communications between two nodes. When you troubleshoot channel formation problems, first check the obvious:

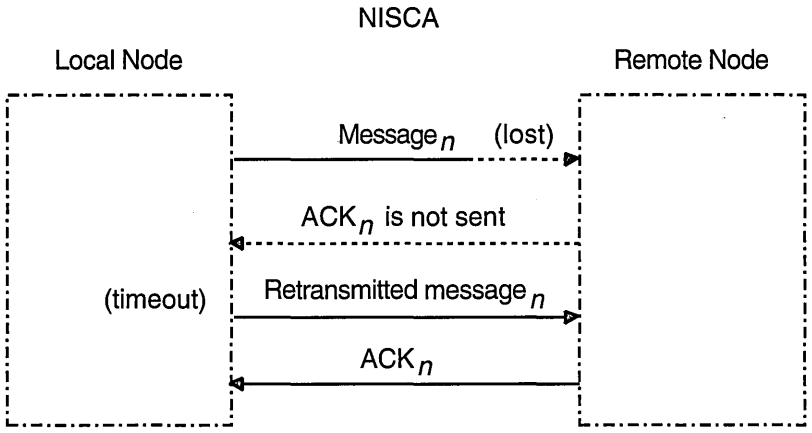
- Is the remote node powered on?
- Is the remote node booted?
- Are the required network connections connected?
- Do the cluster multicast datagrams pass through all of the required bridges in both directions?
- Are the cluster group code and password the same on all nodes?

Next, check for dead channels by using SDA. The SDA command SHOW PORT can help you determine whether a channel ever existed; the command displays the channel's state. (Refer to Section G.2.2 for examples of the SHOW PORT command.) Section G.6.1 describes how to use a LAN analyzer to troubleshoot channel formation problems. Also, see Appendix D for information about using the LAVC$FAILURE_ANALYSIS program to troubleshoot channel problems.

G.3.2 Retransmission Problems

Retransmissions occur when the local node does not receive acknowledgment of a message in a timely manner. This occurs typically if the node runs out of a critical resource, such as large request packets (LRPs) or nonpaged pool, and a message is lost after it reaches the remote node. Other potential causes of retransmissions include overloaded LAN bridges, slow LAN adapters (such as the DELQA), and heavily loaded systems, which delay packet transmission or reception. Figure G–3 shows an unsuccessful transmission followed by a successful retransmission.

Figure G–3 Lost Messages Cause Retransmissions



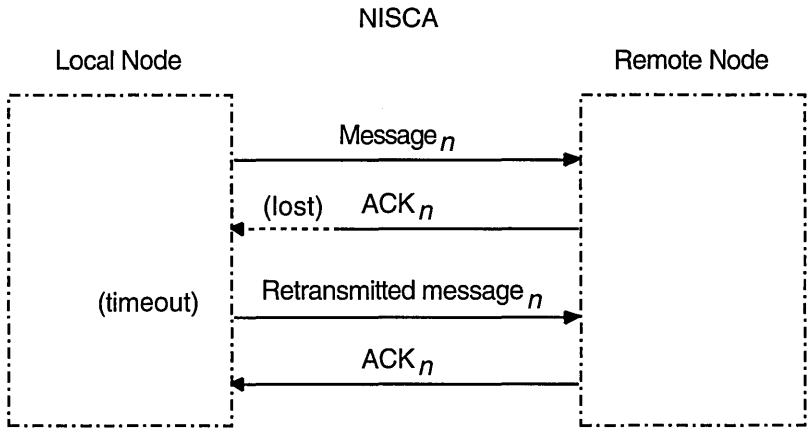
Note: *n* represents a 16–bit number that identifies each sequenced message.

ZK–5941A–GE

Because the first message was lost, the local node does not receive acknowledgment (ACK) from the remote node. The remote node acknowledged the second (successful) transmission of the message.

Retransmission can also occur if the cables are seated improperly, if the network is too busy and the datagram cannot be sent, or if the datagram is corrupted or lost during transmission either by the originating LAN adapter or by any bridges or repeaters. Figure G–4 illustrates another type of retransmission.

Figure G–4 Lost ACKs Cause Retransmissions



Note: *n* represents a 16–bit number that identifies each sequenced message.

ZK–5942A–GE

In Figure G–4, the remote node receives the message and transmits an acknowledgment (ACK) to the sending node. However, because the ACK from the receiving node is lost, the sending node retransmits the message.

The first time the sending node transmits the datagram containing the sequenced message data, PEDRIVER sets the value of the REXMT flag bit in the TR header to 0. If the datagram requires retransmission, PEDRIVER sets the REXMT flag bit to 1 and resends the datagram. PEDRIVER retransmits the datagram until either the datagram is received or the virtual circuit is closed. If multiple channels are available, PEDRIVER attempts to retransmit the message on a different channel in an attempt to avoid the problem that caused the retransmission.

You can troubleshoot cluster retransmissions using a LAN protocol analyzer for each LAN segment. If multiple segments are used for cluster communications, then the LAN analyzers need to support a distributed enable and trigger mechanism (see Section G.5). See also Appendix I for more information about how PEDRIVER chooses channels on which to transmit datagrams.

Techniques for isolating the retransmitted datagram using a LAN analyzer are discussed in Section G.6.2. See also Appendix H for more information about congestion control and PEDRIVER message retransmission.

G.4 Understanding the Format of NISCA Datagrams

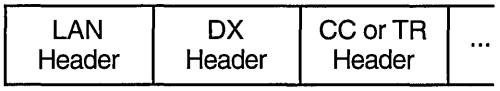
Troubleshooting NISCA protocol communication problems requires an understanding of the NISCA protocol packet that is exchanged across the VMScluster system.

The format of packet on the NISCA protocol is defined by the $NISCADEF macro, which is located in [DRIVER.LIS] on VAX systems and in [LIB.LIS] for AXP systems.

Figure G–5 shows the general form of NISCA datagrams. A NISCA datagram consists of the following headers, which are usually followed by user data:

- LAN headers, including an Ethernet or an FDDI header
- Datagram exchange (DX) header
- Channel control (CC) or transport (TR) header

Figure G–5 NISCA Headers



ZK-5920A-GE

Caution

The NISCA protocol is subject to change without notice.

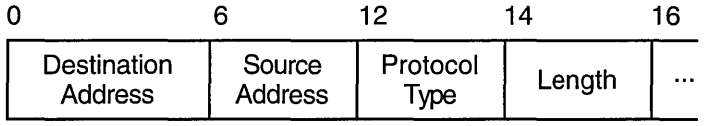
G.4.1 LAN Headers

The NISCA protocol is supported on LANs consisting of Ethernet and FDDI. Sections G.4.1.1 and G.4.1.2 describe the Ethernet and FDDI headers. These headers contain information that is useful for diagnosing problems that occur between LAN adapters. See Section G.5.2.4 for methods of isolating information in LAN headers.

G.4.1.1 Ethernet Header

Each datagram that is transmitted or received on the Ethernet is prefixed with an Ethernet header. The Ethernet header, shown in Figure G–6, is 16 bytes long.

Figure G–6 Ethernet Header



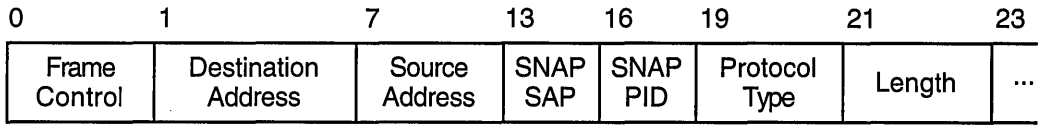
ZK-5921A-GE

- Destination address—LAN address of the adapter that should receive the datagram
- Source address—LAN address of the adapter sending the datagram
- Protocol type—NISCA protocol (60–07) hexadecimal
- Length—Number of data bytes in the datagram following the length field

G.4.1.2 FDDI Header

Each datagram that is transmitted or received on the FDDI is prefixed with an FDDI header. The NISCA protocol uses mapped Ethernet format datagrams on the FDDI. The FDDI header, shown in Figure G–7, is 23 bytes long.

Figure G–7 FDDI Header



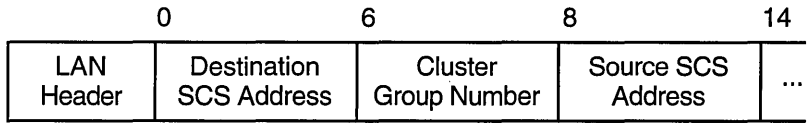
ZK–5922A–GE

- Frame control—NISCA datagrams are logical link control (LLC) frames with a priority value (5*x*). The low-order 3 bits of the frame-control byte contain the priority value. All NISCA frames are transmitted with a nonzero priority field. Frames received with a zero priority field are assumed to have traveled over an Ethernet segment because Ethernet packets do not have a priority value and because Ethernet-to-FDDI bridges generate a priority value of 0.
- Destination address—LAN address of the adapter that should receive the datagram.
- Source address—LAN address of the adapter sending the datagram.
- SNAP SAP—Subnetwork access protocol service access point. The value of the access point is AA–AA–03 hexadecimal.
- SNAP PID—Subnetwork access protocol protocol identifier. The value of the identifier is 00–00–00 hexadecimal.
- Protocol type—NISCA protocol (60–07) hexadecimal.
- Length—Number of data bytes in the datagram following the length field.

G.4.2 Datagram Exchange (DX) Header

The datagram exchange (DX) header for the VMScluster protocol is used to address the data to the correct VMScluster node. The DX header, shown in Figure G–8, is 14 bytes long. It contains information that describes the VMScluster connection between two nodes. See Section G.5.2.3 about methods of isolating data for the DX header.

Figure G–8 DX Header



ZK–5923A–GE

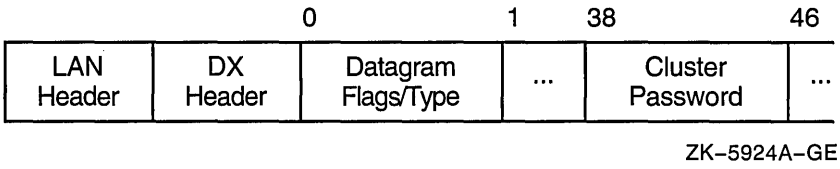
- Destination SCS address—This address is manufactured using the address AA–00–04–00–*local-node*-SCSSYSTEMID. Append the remote node's SCSSYSTEMID system parameter value for the low-order 16 bits. This address represents the destination SCS transport address or the VMScluster multicast address.

- Cluster group number—This value is the cluster group number specified by the system manager. See Section 3.2 for more information about cluster group numbers.
- Source SCS address—This address represents the source SCS transport address and is manufactured using the address AA–00–04–00–*remote-node*-SCSSYSTEMID. Append the local node's SCSSYSTEMID system parameter value as the low-order 16 bits.

G.4.3 Channel Control (CC) Header

The channel control (CC) message is used to form and maintain working network paths between nodes in the VMScluster system. The important fields for network troubleshooting are the datagram flags and type and the cluster password. Note that because the CC and TR headers occupy the same space, there is a TR/CC flag that identifies the type of message being transmitted over the channel. Figure G–9 shows the portions of the CC header needed for network troubleshooting.

Figure G–9 CC Header



- Datagram type (bits <3:0>)—Identifies the type of message on the Channel Control level. Table G–1 shows the datagrams and their functions.

Table G–1 CC Datagrams

Value	Abbreviated Datagram Type	Expanded Datagram Type	Function
0	HELLO	HELLO datagram message	Multicast datagram that initiates the formation of a channel between cluster nodes and tests and maintains the existing channels. This datagram does not contain a valid cluster password.
1	BYE	Node stop notification	Datagram that signals the departure of a cluster node.
2	CCSTART	Channel start	Datagram that starts the channel formation handshake between two cluster nodes. This datagram is sent in response to receiving a HELLO datagram from an unknown LAN adapter address.
3	VERF	Verify	Datagram that acknowledges the CCSTART datagram and continues the channel formation handshake. The datagram is sent in response to receiving a CCSTART or SOLICIT_SRV datagram.
4	VACK	Verify acknowledge	Datagram that completes the channel formation handshake. The datagram is sent in response to receiving a VERF datagram.
5	Reserved		
6	SOLICIT_ SERVER	Solicit	Datagram sent by a booting node to form a channel to its disk server. The server responds by sending a VERF which forms the channel.
7–15	Reserved		

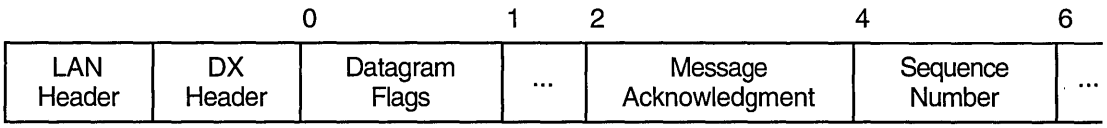
- Datagram flags (bits <7:4>)—Provide additional information about the control datagram. The following bits are defined:
  - Bit <4> (AUTHORIZE)—Set to 1 if the cluster password field is valid.
  - Bit <5> (Reserved)—Set to 1.
  - Bit <6> (Reserved)—Set to 0.
  - Bit <7> (TR/CC flag)—Set to 1 to indicate the CC datagram.
- Cluster password—Contains the cluster password. See Section 3.2 for more information about cluster passwords.

G.4.4 Transport (TR) Header

The transport (TR) header is used to pass SCS datagrams and sequenced messages between cluster nodes. The important fields for network troubleshooting are the TR datagram flags, message acknowledgment, and sequence numbers. Note that because the CC and TR headers occupy the same space, a TR/CC flag identifies the type of message being transmitted over the channel.

Figure G–10 shows the portions of the TR header that are needed for network troubleshooting.

Figure G–10 TR Header



ZK–5925A–GE

- Datagram flags (bits <7:0>)—Provide additional information about the transport datagram. Table G–2 shows the datagrams and their functions.

Table G–2 TR Datagrams

Value	Datagram Type	Expanded Datagram Type	Function
0	DATA	Packet data	Contains data to be delivered to the upper levels of software.
1	SEQ	Sequence flag	Set to 1 if this is a sequenced message and the sequence number is valid.
2	Reserved		Set to 0.
3	ACK	Acknowledgment	Acknowledges the field is valid.
4	RSVP	Reply flag	Set when an ACK datagram is needed immediately.
5	REXMT	Retransmission	Set for all retransmissions of a sequenced message.
6	Reserved		Set to 0.
7	TR/CC flag	Transport flag	Set to 0; indicates a TR datagram.

- Message acknowledgment—An increasing value that specifies the last sequenced message segment received by the local node. All messages prior to this value are also acknowledged.
- Sequence number—An increasing value that specifies the order of datagram transmission from the local node. This number is used to provide guaranteed delivery of this sequenced message segment to the remote node.

G.5 Using a LAN Protocol Analysis Program

Some failures, such as packet loss resulting from congestion, intermittent network interruptions of less than 20 seconds, problems with backup bridges, and intermittent performance problems, can be difficult to diagnose. Intermittent failures may require the use of a LAN analysis tool to isolate and troubleshoot the NISCA protocol levels described in Section G.1.

As you evaluate the various network analysis tools currently available, you should look for certain capabilities when comparing LAN analyzers. The following sections describe the required capabilities according to whether you are troubleshooting problems on a single LAN segment or on multiple LAN segments.

G.5.1 Troubleshooting Single and Multiple LAN Segments

Whether you need to troubleshoot problems on a single LAN segment or on multiple LAN segments, a LAN analyzer should help you isolate specific patterns of data. Choose a LAN analyzer that can isolate data matching unique patterns that you define. You should be able to define data patterns located in the data regions following the LAN header (described in Section G.4.1). In order to troubleshoot the NISCA protocol properly, a LAN analyzer should be able to match multiple data patterns simultaneously.

To troubleshoot single or multiple LAN segments, you must minimally define and isolate transmitted and retransmitted data in the TR header (see Section G.4.4). Additionally, for effective network troubleshooting across multiple LAN segments, a LAN analysis tool should include the following functions:

- A **distributed enable** function that allows you to synchronize multiple LAN analyzers that are set up at different locations so that they can capture information about the same event as it travels through the LAN configuration
- A **distributed combination trigger** function that automatically triggers multiple LAN analyzers at different locations so that they can capture information about the same event

The purpose of distributed enable and distributed combination trigger functions are to capture packets as they travel across multiple LAN segments. The implementation of these functions discussed in the following sections use multicast messages to reach all LAN segments of the extended LAN in the system configuration. By providing the ability to synchronize several LAN analyzers at different locations across multiple LAN segments, the distributed enable and combination trigger functions allow you to troubleshoot LAN configurations that span multiple sites over several miles.

To troubleshoot multiple LAN segments, LAN analyzers must be able to capture the multicast packets and dynamically enable the trigger function of the LAN analyzer, as follows:

1. Start capturing the data according to the rules specific to your LAN analyzer. Digital recommends that only one LAN analyzer transmit a distributed enable multicast packet on the LAN. The packet must be transmitted according to the media access control rules.
2. Wait for the distributed enable multicast packet. When the packet is received, enable the distributed combination trigger function. Prior to receiving the distributed enable packet, all LAN analyzers must be able to ignore the trigger condition. This feature is required in order to set up multiple LAN analyzers capable of capturing the same event. Note that the LAN analyzer transmitting the distributed enable should not wait to receive it.
3. Wait for an explicit (user-defined) trigger event or a distributed trigger packet. When the LAN analyzer receives either of these triggers, the LAN analyzer should stop the data capture.

   Prior to receiving either trigger, the LAN analyzer should continue to capture the requested data. This feature is required in order to allow multiple LAN analyzers to capture the same event.
4. Once triggered, the LAN analyzer completes the distributed trigger function to stop the other LAN analyzers from capturing data related to the event that has already occurred.

The HP 4972A LAN Protocol Analyzer, available from the Hewlett-Packard Company, is one example of a network failure analysis tool that provides the required functions described in this section. Section G.6 provides examples that use the HP 4972A LAN Protocol Analyzer.

G.5.2 Data Isolation Techniques

The following sections describe the types of data you should isolate when you use a LAN analysis tool to capture VMScluster data between nodes and LAN adapters.

G.5.2.1 Isolating All VMScluster Traffic

To isolate all VMScluster traffic on a specific LAN segment, capture all the packets whose LAN header contains the protocol type 60–07. See also Section G.4.1 for a description of the LAN headers.

G.5.2.2 Isolating Specific VMScluster Traffic

To isolate VMScluster traffic for a specific cluster on a specific LAN segment, capture packets in which:

- The LAN header contains the the protocol type 60–07.
- The DX header contains the cluster group number specific to that VMScluster.

See Sections G.4.1 and G.4.2 for descriptions of the LAN and DX headers.

G.5.2.3 Isolating Virtual Circuit (Node-to-Node) Traffic

To isolate virtual circuit traffic between a specific pair of nodes, capture packets in which the LAN header contains:

- The protocol type 60–07
- The destination SCS address
- The source SCS address

You can further isolate virtual circuit traffic between a specific pair of nodes to a specific LAN segment by capturing the following additional information from the DX header:

- The cluster group code specific to that VMScluster
- The destination SCS transport address
- The source SCS transport address

See Sections G.4.1 and G.4.2 for LAN and DX header information.

G.5.2.4 Isolating Channel (LAN Adapter–to–LAN Adapter) Traffic

To isolate channel information, capture all packet information on every channel between LAN adapters. The DX header contains information useful for diagnosing heavy communication traffic between a pair of LAN adapters. Capture packets in which the LAN header contains:

- The destination LAN adapter address
- The source LAN adapter address

Because nodes can use multiple LAN adapters, specifying the source and destination LAN addresses may not capture all of the traffic for the node. Therefore, you must specify a channel as the source LAN address and the destination LAN address in order to isolate traffic on a specific channel. See Section G.4.1 for information about the LAN header.

G.5.2.5 Isolating Channel Control Traffic

To isolate channel control traffic, capture packets in which:

- The LAN header contains the the protocol type 60–07.
- The CC header datagram flags byte (the TR/CC flag, bit <7>) is set to 1.

See Sections G.4.1 and G.4.3 for a description of the LAN and CC headers.

G.5.2.6 Isolating Transport Data

To isolate transport data, capture packets in which:

- The LAN header contains the the protocol type 60–07.
- The TR header datagram flags byte (the TR/CC flag, bit <7>) is set to 0.

See Sections G.4.1 and G.4.4 for a description of the LAN and TR headers.

G.6 Setting Up an HP 4972A LAN Protocol Analyzer

The HP 4972A LAN Protocol Analyzer, available from the Hewlett-Packard Company, is highlighted here because it meets all of the requirements listed in Section G.5. However, the HP 4972A LAN Protocol Analyzer is merely representative of the type of product useful for LAN network troubleshooting. Use of this particular product as an example here should not be construed as a specific purchase requirement or endorsement.

This section provides some examples of how to set up the HP 4972A LAN Protocol Analyzer to troubleshoot the local area VMScluster system protocol for channel formation and retransmission problems.

G.6.1 Setting Up the LAN Analyzer to Troubleshoot Channel Formation Problems

If you have a LAN protocol analyzer, you can set up filters to capture data related to the channel control header (described in Section G.4.3).

You can trigger the LAN analyzer by using the following datagram fields:

- Protocol type set to 60–07 hexadecimal
- Correct cluster group number
- TR/CC flag set to 1

Then look for the HELLO, CCSTART, VERF, and VACK datagrams in the captured data. The CCSTART, VERF, VACK, and SOLICIT_SRV datagrams should have the AUTHORIZE bit (bit <4>) set in the CC flags byte. Additionally, these messages should contain the encrypted cluster password (nonzero authorization field). You can find the encrypted cluster password and the cluster group number in the first four longwords of SYS$SYSTEM:CLUSTER_AUTHORIZE.DAT file.

See Sections G.5.2.3 through G.5.2.5 for additional data isolation techniques.

G.6.2 Setting Up the LAN Analyzer to Troubleshoot Retransmission Problems

Using a LAN analyzer, you can trace datagrams as they travel across a VMScluster system. Trigger the analyzer using the following datagram fields:

- Protocol type set to 60–07
- Correct cluster group number

- TR/CC flag set to 0
- REXMT flag set to 1

Use the distributed enable function to allow the same event to be captured by several LAN analyzers at different locations. The LAN analyzers should start the data capture, wait for the distributed enable message, and then wait for the explicit trigger event or the distributed trigger message. Once triggered, the analyzer should complete the distributed trigger function to stop the other LAN analyzers capturing data.

Once all the data is captured, locate the sequence number for the datagram being retransmitted (the datagram with the REXMT flag set). Then, search through the previously captured data for another datagram between the same two nodes (not necessarily the same LAN adapters) with the following characteristics:

- Protocol type set to 60–07
- Same DX header as the datagram with the REXMT flag set
- TR/CC flag set to 0
- REXMT flag set to 0
- Same sequence number as the datagram with the REXMT flag set

This technique is a way of searching for the problem's origin. If the datagram appears to be corrupt, then use the LAN analyzer to search in the direction of the source node for the corruption cause. If the datagram appears to be correct, search in the direction of the destination node to ensure that the datagram gets to its destination. If the datagram arrives successfully at its LAN segment destination, look for a TR packet from the destination node containing the sequence number in the message acknowledgment field. ACK datagrams have the following fields set:

- Protocol type set to 60–07
- Same DX header as the datagram with the REXMT flag set
- TR/CC flag set to 0
- ACK flag set to 1

If the acknowledgment was not sent or if a significant delay occurred between the reception of the message and the transmission of the acknowledgment, then look for a problem with the destination node and LAN adapter. Then follow the ACK packet through the network. If the ACK arrives back at the node that sent the retransmission packet, either of the following conditions may exist:

- The retransmitting node is having trouble receiving LAN data.
- The round-trip delay of the original datagram exceeded the estimated timeout value.

You can verify the second possibility by using SDA and looking at the ReRcv field of the virtual circuit display of the system receiving the retransmitted datagram. (See Example G–2 for an example of this type of SDA display.) Refer to Appendix H for more information about congestion control and PEDRIVER message retransmission.

G.6.3 Filters

This section describes how to use the HP 4972A LAN Protocol Analyzer filters to isolate packets that have been retransmitted or that are specific to a particular VMScluster. This section also describes how to enable the distributed enable and trigger functions.

G.6.3.1 Capturing All Local Area VMScluster Retransmissions for a Specific Cluster

Use the values shown in Table G–3 to set up a filter, named LAVc_TR_ReXMT, for all of the local area VMScluster retransmissions for a specific cluster. Fill in the value for the local area VMScluster group code (*nn–nn*) to isolate a specific VMScluster on the LAN.

Table G–3 Capturing Local Area VMScluster Retransmissions (LAVc_TR_ReXMT)

Byte Number	Field	Value
1	DESTINATION	*XX–XX–XX–XX–XX–XX*
7	SOURCE	*XX–XX–XX–XX–XX–XX*
13	TYPE	60–07
23	LAVC_GROUP_CODE	*nn–nn*
31	TR FLAGS	$0X1XXXXX_2$
33	ACKING MESSAGE	*XX–XX*
35	SENDING MESSAGE	*xx–xx*

G.6.3.2 Capturing All Local Area VMScluster Packets for a Specific Cluster

Use the values shown in Table G–4 to filter all of the local area VMScluster packets for a specific cluster. Fill in the value for local area VMScluster group code (*nn–nn*) to isolate a specific VMScluster on the LAN. The filter is named LAVc_all.

Table G–4 Capturing All Local Area VMScluster Packets (LAVc_all)

Byte Number	Field	Value
1	DESTINATION	*XX–XX–XX–XX–XX–XX*
7	SOURCE	*XX–XX–XX–XX–XX–XX*
13	TYPE	60–07
23	LAVC_GROUP_CODE	*nn–nn*
33	ACKING MESSAGE	*XX–XX*
35	SENDING MESSAGE	*XX–XX*

G.6.3.3 Setting Up the Distributed Enable Filter

Use the values shown in Table G–5 to set up a filter, named Distrib_Enable, for the distributed enable packet received event. Use this filter to troubleshoot multiple LAN segments.

Table G–5 Setting Up a Distributed Enable Filter (Distrib_Enable)

Byte Number	Field	Value	ASCII
1	DESTINATION	01–4C–41–56–63–45	.LAVcE
7	SOURCE	*XX–XX–XX–XX–XX–XX*	
13	TYPE	60–07	‘.
15	TEXT	*XX*	

G.6.3.4 Setting Up the Distributed Trigger Filter

Use the values shown in Table G–6 to set up a filter, named Distrib_Trigger, for the distributed trigger packet received event. Use this filter to troubleshoot multiple LAN segments.

Table G–6 Setting Up the Distributed Trigger Filter (Distrib_Trigger)

Byte Number	Field	Value	ASCII
1	DESTINATION	01–4C–41–56–63–54	.LAVcT
7	SOURCE	*XX–XX–XX–XX–XX–XX*	
13	TYPE	60–07	‘.
15	TEXT	*XX*	

G.6.4 Messages

This section describes how to set up the distributed enable and distributed trigger messages.

G.6.4.1 Distributed Enable Message

Table G–7 shows how to define the distributed enable message (Distrib_Enable) by creating a new message. You must replace the source address (*nn nn nn nn nn nn*) with the LAN address of the LAN analyzer.

Table G–7 Setting Up the Distributed Enable Message (Distrib_Enable)

Field	Byte Number	Value	ASCII
Destination	1	01 4C 41 56 63 45	.LAVcE
Source	7	*nn nn nn nn nn nn*	
Protocol	13	60 07	‘.
Text	15	44 69 73 74 72 69 62 75 74 65	Distribute
	25	64 20 65 6E 61 62 6C 65 20 66	d enable f
	35	6F 72 20 74 72 6F 75 62 6C 65	or trouble
	45	73 68 6F 6F 74 69 6E 67 20 74	shooting t
	55	68 65 20 4C 6F 63 61 6C 20 41	he Local A
	65	72 65 61 20 56 4D 53 63 6C 75	rea VMSclu
	75	73 74 65 72 20 50 72 6F 74 6F	ster Proto
	85	63 6F 6C 3A 20 4E 49 53 43 41	col: NISCA

G.6.4.2 Distributed Trigger Message

Table G–8 shows how to define the distributed trigger message (Distrib_Trigger) by creating a new message. You must replace the source address (*nn nn nn nn nn nn*) with the LAN address of the LAN analyzer.

Table G–8 Setting Up the Distributed Trigger Message (Distrib_Trigger)

Field	Byte Number	Value	ASCII
Destination	1	01 4C 41 56 63 54	.LAVcT
Source	7	*nn nn nn nn nn nn*	
Protocol	13	60 07	`.
Text	15	44 69 73 74 72 69 62 75 74 65	Distribute
	25	64 20 74 72 69 67 67 65 72 20	d trigger
	35	66 6F 72 20 74 72 6F 75 62 6C	for troubl
	45	65 73 68 6F 6F 74 69 6E 67 20	eshooting
	55	74 68 65 20 4C 6F 63 61 6C 20	the Local
	65	41 72 65 61 20 56 4D 53 63 6C	Area VMScl
	75	75 73 74 65 72 20 50 72 6F 74	uster Prot
	85	6F 63 6F 6C 3A 20 4E 49 53 43	ocol: NISC
	95	41	A

G.6.5 Programs That Capture Retransmission Errors

You can program the HP 4972 LAN Protocol Analyzer, as shown in the following source code, to capture retransmission errors. The starter program initiates the capture across all of the LAN analyzers. Only one LAN analyzer should run a copy of the starter program. Other LAN analyzers should run either the partner program or the scribe program. The partner program is used when the initial location of the error is unknown and when all analyzers should cooperate in the detection of the error. Use the scribe program to trigger on a specific LAN segment as well as to capture data from other LAN segments.

G.6.5.1 Starter Program

The starter program initially sends the distributed enable signal to the other LAN analyzers. Next, this program captures all of the local area VMScluster traffic, and terminates as a result of either a retransmitted packet detected by this LAN analyzer or after receiving the distributed trigger sent from another LAN analyzer running the partner program.

The starter program shown in the following example is used to initiate data capture on multiple LAN segments using multiple LAN analyzers. The goal is to capture the data during the same time interval on all of the LAN segments so that the reason for the retransmission can be located.

```
Store: frames matching LAVc_all
 or Distrib_Enable
 or Distrib_Trigger
       ending with LAVc_TR_ReXMT
        or Distrib_Trigger

Log file: not used
```

```
Block 1:   Enable_the_other_analyzers
     Send message Distrib_Enable
       and then
     Go to block 2

Block 2:   Wait_for_the_event
     When frame matches LAVc_TR_ReXMT then go to block 3

Block 3:   Send the distributed trigger
     Mark frame
       and then
     Send message Distrib_Trigger
```

G.6.5.2 Partner Program

The partner program waits for the distributed enable; then it captures all of the local area VMScluster traffic and terminates as a result of either a retransmission or the distributed trigger. Upon termination, this program transmits the distributed trigger to make sure that other LAN analyzers also capture the data at about the same time as when the retransmitted packet was detected on this segment or another segment. After the data capture completes, the data from multiple LAN segments can be reviewed to locate the initial copy of the data that was retransmitted. The partner program is shown in the following example:

```
Store: frames matching LAVc_all
        or Distrib_Enable
        or Distrib_Trigger
       ending with Distrib_Trigger

Log file: not used

Block 1:   Wait_for_distributed_enable
     When frame matches Distrib_Enable then go to block 2

Block 2:   Wait_for_the_event
     When frame matches LAVc_TR_ReXMT then go to block 3

Block 3:   Send the distributed trigger
     Mark frame
       and then
     Send message Distrib_Trigger
```

G.6.5.3 Scribe Program

The scribe program waits for the distributed enable and then captures all of the local area VMScluster traffic and terminates as a result of the distributed trigger. The scribe program allows a network manager to capture data at about the same time as when the retransmitted packet was detected on another segment. After the data capture has completed, the data from multiple LAN segments can be reviewed to locate the initial copy of the data that was retransmitted. The scribe program is shown in the following example:

```
Store: frames matching LAVc_all
        or Distrib_Enable
        or Distrib_Trigger
       ending with Distrib_Trigger

Log file: not used

Block 1:   Wait_for_distributed_enable
     When frame matches Distrib_Enable then go to block 2

Block 2:   Wait_for_the_event
     When frame matches LAVc_TR_ReXMT then go to block 3
```

```
Block 3:   Mark_the_frames
     Mark frame
       and then
     Go to block 2
```

H

PEDRIVER Congestion Control

Network congestion can have a negative impact on cluster performance in several ways. Moderate levels of congestion can lead to increased queue lengths in network components (such as adapters and bridges) that in turn can lead to increased latency and slower response. Higher levels of congestion can result in the discarding of packets because of queue overflow. Packet loss can lead to packet retransmissions and, potentially, even more congestion. In extreme cases, packet loss can result in the loss of VMScluster connections.

Network congestion occurs as the result of complex interactions of workload distribution and network topology, including the speed and buffer capacity of individual hardware components. Thus, while a particular network component or protocol cannot guarantee the absence of congestion, PEDRIVER now incorporates several improved mechanisms to mitigate the effects of congestion on VMScluster traffic and to avoid having cluster traffic exacerbate congestion when it occurs. These mechanisms affect the retransmission of packets carrying user data and the multicast HELLO datagrams used to maintain connectivity.

H.1 Retransmission

Associated with each virtual circuit from a given node is a transmission window size, which indicates the number of packets that can be outstanding to the remote node (for example, the number of packets that can be sent to the node at the other end of the virtual circuit before receiving an acknowledgment [ACK]). If the window size is 8 for a particular virtual circuit, then the sender can transmit up to 8 packets in a row but, before sending the ninth, must wait until receiving an ACK indicating that at least the first of the 8 has arrived. If this ACK is not received, a timeout occurs, and the packet is assumed lost and must be retransmitted. With older versions of PEDRIVER, the window size is relatively static, usually 8, 16 or 31 (for FDDI), and the retransmission policy assumes that all outstanding packets are lost and thus retransmits them. Retransmission of an entire window of packets under congestion conditions tends to exacerbate the condition significantly. Another problem is that the timeout interval for determining that a packet is lost is fixed (3 seconds) in older versions. This means that the loss of a single packet can interrupt communication between cluster nodes for as long as 3 seconds.

The new retransmission mechanism is an adaptation of algorithms originally proposed for the Internet by Van Jacobson and improves on the old mechanism by making both the window size and the retransmission timeout interval adapt to network conditions. When a timeout occurs because of a lost packet, the window size is decreased immediately to reduce the load on the network. The window size is allowed to grow only after congestion subsides. More specifically, when a packet loss occurs, the window size is decreased to 1 and remains there, allowing the transmitter to send only one packet at a time until all the original outstanding packets have arrived. From that point, the window is allowed to grow quickly until it reaches half its previous size. Once reaching the halfway

point, the window size is allowed to increase relatively slowly to take advantage of available network capacity until it reaches a maximum value determined by the configuration variables (for example, number of adapter buffers).

The retransmission timeout interval in the new PEDRIVER is set based on measurements of actual round-trip times for packets that are transmitted over the virtual circuit. This allows PEDRIVER to be more responsive to packet loss in most networks but avoids premature timeouts for networks in which the actual round-trip delay approaches several seconds.

H.2 HELLO Multicast Datagrams

PEDRIVER periodically multicasts a HELLO datagram over each network adapter attached to the node. The HELLO datagram serves two purposes:

- It informs other nodes of the existence of the sender so that they can form channels and virtual circuits.
- It keeps communications open once they are established.

HELLO datagram congestion and loss of HELLO datagrams can prevent connections from forming or cause connections to be lost.

If all existing nodes that receive a HELLO datagram from a new node were to respond immediately to form connections, the receiving network adapter on the new node could be overrun with HELLO datagrams and be forced to drop some, resulting in connections not being formed. This is especially likely in large clusters.

To avoid this problem, nodes that receive HELLO datagrams delay for a random time interval of up to 1 second before responding. This random delay is increased to a maximum of 2 seconds to support large VMScluster systems.

HELLO datagram congestion can also occur if a large number of nodes in a network become synchronized and transmit their HELLO datagrams at or near the same time. Prior to OpenVMS VAX Version 6.0 and OpenVMS AXP Version 1.5, PEDRIVER multicast HELLO datagrams over each adapter every 3 seconds. To prevent this form of HELLO datagram congestion, PEDRIVER now distributes its HELLO datagram multicasts randomly over time. A HELLO datagram is still multicast over each adapter approximately every 3 seconds, but not over all adapters at once. Instead, if a node has multiple network adapters, PEDRIVER attempts to distribute its HELLO datagram multicasts so that it sends a HELLO datagram over some of its adapters during each second of the 3-second interval. In addition, rather than multicasting precisely every 3 seconds PEDRIVER varies the time between HELLO datagram multicasts between approximately 1.6 to 3 seconds changing the average from 3 seconds to approximately 2.3 seconds. The result is a substantial decrease in the probability of HELLO datagram synchronization and thus a decrease in HELLO datagram congestion.

I

Transmit Channel Selection

Of the channels available to a given remote node, PEDRIVER uses a single channel to transmit datagrams and all channels to receive datagrams. The driver software chooses a transmission channel for each remote node. A selection algorithm for the transmission channel ensures that messages are sent in the order they are expected to be received. Sending the messages in this way also maintains compatibility with previous versions of the operating system. The selected transmission channel is called the **preferred channel**.

PEDRIVER continually computes a network delay value for each channel. PEDRIVER switches the preferred channel based on observed network delays or network component failures. Switching to a new transmission channel sometimes causes messages to be received out of the desired order. PEDRIVER now uses a receive cache to reorder these messages instead of discarding them. PEDRIVER's use of the receive cache prevents the remote node from retransmitting the messages.

Some restrictions apply to remote nodes running a version of the operating system prior to VMS Version 5.4–3. Messages received out of order on such remote nodes are discarded because they lack the receive cache. For these remote nodes, PEDRIVER cannot switch channels based on the observed network delays. In this case, PEDRIVER chooses a single transmission channel and uses it until the channel fails. Only at that time will PEDRIVER switch to another channel.

With these algorithms, PEDRIVER has a greater chance of utilizing adapters on a server node that communicates continuously with a number of clients. In a two-node cluster, PEDRIVER will actively use, at most, two LAN adapters: one to transmit and one to receive. Additional adapters provide both higher availability and alternate paths that can be used to avoid network congestion. As more nodes are added to the cluster, PEDRIVER is more likely to use the additional adapters.

Index

D

E

F

G

H

I

J

L

M

N

R

S

T

U

V

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825) and press 2 for technical assistance.

Electronic Orders

If you wish to place an order through your account at the Electronic Store, dial 800-234-1998, using a modem set to 2400- or 9600-baud. You must be using a VT terminal or terminal emulator set at 8 bits, no parity. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825) and ask for an Electronic Store specialist.

Telephone and Direct Mail Orders

From	Call	Write
U.S.A.	DECdirect Phone: 800-DIGITAL (800-344-4825) FAX: (603) 884-5597	Digital Equipment Corporation P.O. Box CS2008 Nashua, NH 03061
Puerto Rico	Phone: (809) 781-0505 FAX: (809) 749-8377	Digital Equipment Caribbean, Inc. 3 Digital Plaza, 1st Street Suite 200 Metro Office Park San Juan, Puerto Rico 00920
Canada	Phone: 800-267-6215 FAX: (613) 592-1946	Digital Equipment of Canada Ltd. 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: DECdirect Sales
International	————	Local Digital subsidiary or approved distributor
Internal Orders[1] (for software documentation)	DTN: 241-3023 (508) 874-3023	Software Supply Business (SSB) Digital Equipment Corporation 1 Digital Drive Westminster, MA 01473
Internal Orders (for hardware documentation)	DTN: 234-4325 (508) 351-4325 FAX: (508) 351-4467	Publishing & Circulation Services Digital Equipment Corporation NR02-2 444 Whitney Street Northboro, MA 01532

[1]Call to request an Internal Software Order Form (EN–01740–07).

Reader's Comments

VMScluster Systems for OpenVMS
AA–PV5WA–TK

Your comments and suggestions help us improve the quality of our publications.

Thank you for your assistance.

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

I would like to see more/less _____

_____

_____

What I like best about this manual is _____

_____

_____

What I like least about this manual is _____

_____

_____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

For software manuals, please indicate which version of the software you are using: _____

_____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

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

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OpenVMS Documentation
110 SPIT BROOK ROAD ZKO3–4/U08
NASHUA, NH 03062–2642

--- Do Not Tear - Fold Here ----------