Data General
Data General Corporation, Westboro, Massachusetts 01580

Customer Documentation

Achieving High Availability on AViiON® Systems

093–701133–01

AViiON®
PRODUCT LINE

Achieving High Availability on AViiON® Systems

093–701133–01

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085–series) and/or Update Notice (078–series) supplied with the software.

Printed in the United States of America
Rev. 01, July 1994

Ordering No. 093–701133

Notice

Achieving High Availability on AViiON® Systems
093–701133–01

Revision History:

Original Release –January, 1994
Revision 1 – July 1994

Effective with:

DG/UX System 5.4 Release 3.10

About this manual

This manual provides an overview of the high-availability solutions available with Data General's AViiON® systems. The manual provides details about both the hardware and software elements of these high-availability systems. It contains details about managing DG/UX system high-availability features, especially in the area of failover.

In addition, this manual contains examples of setting up Data General's high-availability features on representative systems. These examples cover both single system features and multiple system failover configurations.

Many of the components covered in Chapters 2 and 3 have their own manuals or are covered more extensively in another manual. This manual contains references to these other manuals.

How this manual is organized

This manual contains six chapters and three appendixes. The following list gives an overview of what you will find in these chapters:

Chapter 1	Introduces the concept of high availability and contrasts the capabilities of highly-available systems with those of normal systems and fault tolerant systems. Explains Data General's system approach to providing high availability.
Chapter 2	Provides details about the hardware components of Data General's high-availability systems.
Chapter 3	Provides details about the software components of Data General's high-availability systems.
Chapter 4	Provides details about managing DG/UX high-availability features, including disk failover, IP takeover and NFS failover, and multi-path I/O.
Chapter 5	Presents an example of setting up a single system high-availability configuration.
Chapter 6	Provides planning information for and examples of setting up systems in a failover configuration.
Appendix A	Contains a worksheet for planning high-availability configurations.

Appendix B Provides details about installing or upgrading the DG/UX system on hosts in a dual-initiator configuration.

Appendix C Provides details about setting up tape sharing between hosts in a dual-initiator configuration.

Related documents

This section contains references to documents that provide additional information on various high-availability features.

Data General Manuals

Within this manual, we refer to the following manuals:

Managing the DG/UX™ System (093–701088). Discusses the concepts of DG/UX system management. Explains how to customize and manage a system using commands and the **sysadm** system management tool. Includes instructions for booting and shutting down the system, backing up and restoring files and file systems, and recovering from system failure. Tells how to manage users, system services and activity, application software, and accounting.

Managing Mass Storage Devices and DG/UX™ File Systems (093–701136). Explains how to manage disk and tape drives. Also explains DG/UX file systems, virtual disks, mirrors, and caching.

Configuring and Managing a CLARiiON® Disk-Array Storage System — DG/UX™ Environment (014–002323). For people who want to configure and manage the CLARiiON® disk-array storage system in a DG/UX environment, this manual describes planning, configuring and maintaining the storage system.

The CLARiiON™ Tape-Array Storage System with the DG/UX™ or AOS/VS Operating System (014–002181). For people who want to configure and manage the CLARiiON™ tape-array storage system, this manual describes the system's hardware, firmware, and software.

Using AViiON® Diagnostics and the AV/Alert℠ Diagnostic Support System (014–002183). For people who want to use the diagnostic features available for AViiON® systems, this manual describes how to operate the AV/Alert℠ system, and how to use the service manager software (SVCMGR) and AViiON System Diagnostics.

Other software packages

Within this manual, we refer to the following manuals for other software packages with high-availability features:

Tuxedo System Release 4.2 Product Overview (069–701088). For people who want to know more about the TUXEDO® System/T software package, this manual provides an overview of the package and a list of the manuals in the manual set for the product.

Legato NetWorker Administrator's Guide (069–100495). For system administrators, this manual describes how to configure and manage the Legato Networker™ software package.

Legato NetWorker User's Guide (069–100496). For people who want to use Legato Networker, this manual describes how to operate the software from a client workstation.

*Managing a Network with OS/EYE*NODE™ for AViiON® Systems* (093–000812). For system administrators, this manual describes how to install, configure, and manage the OS/EYE*NODE software package.

NetWare® for AViiON Systems: Installation Manual (069–000488). For system administrators, this manual describes how to install the NetWare® for AViiON Systems software package.

Installing, Configuring and Operating SNA for AViiON Systems (093–000676). For system administrators or SNA network administrators, this manual describes how to configure and manage the SNA for AViiON Systems software packages.

Technical briefs

The following technical briefs contain information related to high-availability features:

High Availability and AViiON Systems (012–004567). This brief introduces the concept of high availability and explains Data General's approach to providing high availability.

The DG/UX™ File System (012–004054). This brief covers the DG/UX file system, including features such as fast recovery file systems and dynamic remapping of bad blocks.

A Look at High Availability Disk Systems (012–004035). This brief covers Data General's RAID high-availability disk systems.

Operator Initiated Failover in the DG/UX™ 5.4.2 Operating System (012–004186). This brief covers setting up two AViiON servers and a dual-initiator configured CLARiiON disk-array storage system for Operator Initiated Failover.

Machine Initiated Failover in the DG/UX™ 5.4.2 Operating System (012–004245). This brief covers adding Machine Initiated Failover functionality to a system set up for Operator Initiated Failover.

Managing Virtual Disks with On-Line Storage Management (OSM) (012–004406). This brief covers virtual disk and On-Line Storage Management concepts.

RAID technology

The RAID Book — A Source Book for RAID Technology (012–004362). This source book gives an overview of RAID technology and describes disk striping, disk mirroring, parallel access arrays, and non-Berkeley RAID levels. This document is published by the RAID Advisory Board.

Reader, please note:

Throughout this manual we use the following format conventions:

COMMAND required **required** [*optional*] ...

Where	Means
COMMAND	You must enter the command (or its accepted abbreviation) as shown.
required	You must enter some argument (such as a filename). Sometimes, we use {required1 required2} which means you must enter one of the arguments. Do not type the braces; they only set off the choices.
required	You must enter the case–sensitive characters as shown.
[*optional*]	You have the option of entering this argument. Do not type the brackets; they only identify the argument as an option.
...	You may repeat the preceding entry.

Additionally, we use certain symbols in command lines.

Symbol	Means
↵	Press the New Line, Carriage Return (CR), or Enter key on your terminal keyboard.
<Ctrl–D>	Hold the Control key down and press the D key on your terminal keyboard.

%	The UNIX® shell prompt.

Finally, in examples we use:

This typeface to show your entry.
This typeface to show system queries and responses.
`This typeface to show file contents.`

Contacting Data General

Data General wants to assist you in any way it can to help you use its products. Please feel free to contact the company as outlined below.

Manuals

If you require additional manuals, please use the enclosed TIPS order form (United States only) or contact your local Data General sales representative.

Telephone assistance

If you are unable to solve a problem using any manual you received with your system, free telephone assistance is available with your hardware warranty and with most Data General software service options. If you are within the United States or Canada, contact the Data General Customer Support Center (CSC) by calling 1–800–DG–HELPS. Lines are open from 8:00 a.m. to 5:00 p.m., your time, Monday through Friday. The center will put you in touch with a member of Data General's telephone assistance staff who can answer your questions.

For telephone assistance outside the United States or Canada, ask your Data General sales representative for the appropriate telephone number.

Joining our users group

Please consider joining the largest independent organization of Data General users, the North American Data General Users Group (NADGUG). In addition to making valuable contacts, members receive *FOCUS* monthly magazine, a conference discount, access to the Software Library and Electronic Bulletin Board, an annual Member Directory, Regional and Special Interest Groups, and much more. For more information about membership in the North American Data General Users Group, call 1–800–253–3902 or 1–508–443–3330.

End of Preface

Contents

Chapter 1 – What is high availability?

Chapter 2 – Hardware components

Chapter 3 – Software components

Chapter 4 – Managing DG/UX high-availability features

Chapter 5 – Setting up single system high-availability features

Chapter 6 – Setting up failover configurations

Appendix A – High-availability worksheet

Appendix B – Installing the DG/UX system on hosts in a dual-initiator configuration

Appendix C – Sharing tape drives

Tables

Table

Figures

Figure

1 What is high availability?

High availability (HA) is the integrated set of system resources and services required to maximize system availability and minimize application downtime. These resources and services combine to provide near-continuous availability of applications at a reasonable cost.

No single system component can provide high availability. To achieve near-continuous availability, the entire system should be designed with that purpose in mind. All system components must work together to ensure that a system keeps running.

System availability is measured by how long a system can run without failing, and how long it takes to repair a system. Reliability measures include Mean Time to Repair (MTTR),and Mean Time to Failure (MTTF). *MTTR* is the mean time it takes for a system to recover. *MTTF* is the mean time it takes a system to fail after it is repaired. Figure 1–1 illustrates the relationship between these values.

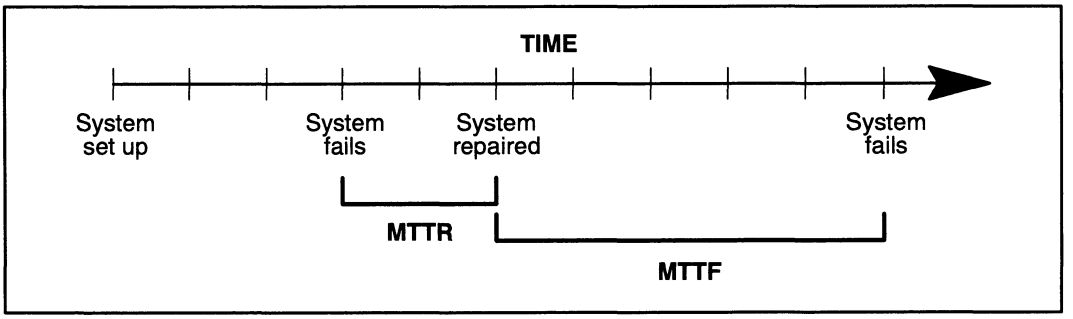


Figure 1–1 Relationship between availability measures

The measure of a system's availability is obtained by dividing the MTTF by the sum of the MTTF and the MTTR. For example, assume a system's MTTF is 10,000 hours and the MTTR is 1 hour. This sum of these measures is 10000 + 1, or 10,001 hours. In this case, the system's availability is 10000 / 10001, or 99.99%.

Generally, the MTTF in a high-availability system is relatively long, and the MTTR is relatively short.

Types of systems

High-availability systems occupy the middle ground between "typical" systems and continuous availability systems. Figure 1–2 illustrates the general relationship between these systems.

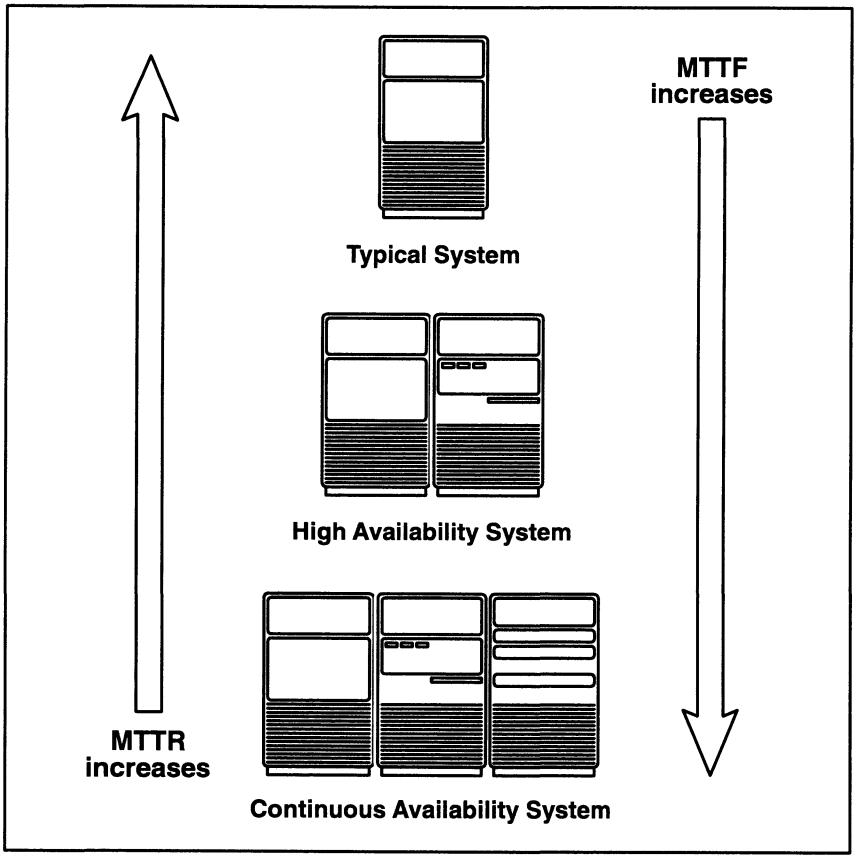


Figure 1–2 Relationship between typical, high-availability, and continuous availability systems

Typical systems

In a typical system, the MTTF is usually relatively short and the MTTR is fairly long. Such a system would go down occasionally due to problems such as disk failures, memory errors, or application and operating system panics. After the system goes down, it often takes a long time to bring it back up. Hardware may need to be replaced or files rebuilt. Once any needed repairs are done, then the system must be brought back up manually.

Continuous availability systems

At the other end of this spectrum are continuous availability systems. The goal for these systems is continuous operation or 100% availability. In an ideal continuous availability system, the MTTF would be infinite and the MTTR zero. While no systems today can reach this ideal, some can approach it.

Continuous availability environments are usually single systems. They run redundant hardware for all parts, including processors, power supplies, input/output (I/O) subsystems, storage subsystems,

and sometimes the entire computer. If a part ever fails, there is usually a built-in replacement part available. The time it takes for a redundant component to take over for a failed component can be less than one second.

The operating system for a continuous availability system should isolate as many error conditions as possible for operating system (OS) components, such as device drivers and file systems, ensuring that errors within each section of the code cannot cause the entire system to panic.

Continuous availability systems have some significant drawbacks. First, these systems tend to be expensive. The need for redundant hardware forces costs up. Also, the research and development effort required for both the hardware and specialized operating system software is time-consuming and costly. These systems also tend to be proprietary. Application software has to be written specifically to run in this environment. In addition, programmers and administrative personnel require special training to work in and manage these systems. Again, this forces costs up.

High-availability systems

High-availability systems provide much higher levels of availability than typical systems, but cost far less than continuous availability systems. A high-availability system does not go down as often as a typical system. When a high-availability system does go down, it comes back up faster than a typical system, quickly restoring applications and data to the state they were in when the system went down. In areas where it is cost-effective, such as I/O subsystems, high-availability systems may have some continuous availability components.

High-availability systems are available in both single system and multiple system configurations. Single system configurations contain features that eliminate the points of failure in a single system. Multiple system availability uses the resources of two or more systems to reduce the application recovery time from system failure.

High-availability systems can have many of the features of continuous availability systems, such as redundant hardware components. In areas where continuous-availability features would be too costly, high-availability systems substitute mechanisms that enable the system to recover quickly from a failure. Instead of guaranteeing continuous operation, high availability ensures the integrity of any data involved with active applications when a system goes down.

Features of high-availability systems

High-availability features usually come in the following two types:

- Single system
- Multiple system

Single system high availability

High-availability systems try to eliminate single points of system failure. For single hosts, efforts to increase availability usually fall into the following areas:

- I/O subsystem
- CPU and memory
- OS and system software

I/O subsystem

One of the earliest ways to add high availability to a system was to eliminate failures in the I/O subsystem. Efforts to add high availability started there for the following reasons:

- It controls the data, the most important asset in the system
- It contains the system components most likely to fail, such as disks

System features such as disk mirroring, dual porting, and redundant I/O channels provide system protection comparable to that available in continuous availability systems.

More advanced I/O high availability has been made possible through the development of Redundant Arrays of Inexpensive Disks (RAID). RAID technology groups individual disks into a single unit. This unit can use disk striping to improve performance. Disk striping enables the hardware to access multiple disks simultaneously. Features such as disk mirroring and parity checks provide data protection.

There are six levels of RAID:

Level 0	Disk array with striping but no data protection
Level 1	Mirrored disk array with duplicated data
Level 1/0	Mirrored disk array with striping
Level 2	Disk array with bit-level striping and Hamming code protection

Level 3	Disk array with byte-interleaved data and parity on one disk
Level 4	Disk array with block-interleaved data and parity on one disk
Level 5	Disk array with block-interweaved data and distributed parity

High-availability systems generally use RAID levels 1, 1/0, 3, and 5. The following sections discuss these levels in more detail:

RAID-1 — Level 1 uses disk mirroring to provide high availability to a disk array. In disk mirroring, two or more disks are bound together into a group. All data is simultaneously written to all of the disks in the group. If one of the disks goes down, one of the others can immediately take over with no loss of either data or service.

A drawback of RAID-1 is that disk hardware is at least doubled due to the need for redundant disks. Also, write times increase slightly since data is being written to two or more drives. An advantage is the capability for on-line backups and on-line disk replacement, both of which reduce both planned and unplanned downtime. Mirroring also eliminates several single points of failure because, with two disks, there are two cables and two I/O cards.

RAID-1/0 — Level 1/0 is a combination of RAID-0 and RAID-1. RAID-1/0 uses disk striping to increase performance combined with disk mirroring for high availability. This RAID level has the advantages and drawbacks of RAID-1, but provides greatly increased performance.

RAID-3 — Level 3 uses parity checking to provide continuous data access at a smaller hardware cost than disk mirroring. In RAID-3, a set of disks are bound together as a group. An extra disk is added to the group as a parity drive. Parity information, a checksum calculated from the contents of the drives, is stored on the parity disk. If one of the disks in the group goes down, the parity disk replaces it, using the parity information and the data on the other disks to reconstruct the data on the bad drive.

RAID-3 reduces hardware costs associated with data protection significantly. Since only one parity disk is needed for each RAID-3 group, the cost of disk hardware decreases as more disks are added to the group. For example, if four disks are bound together into a group, the cost increases by one fourth.

RAID-5 — Level 5 is a combination of RAID-0 and RAID-3. RAID-5 uses disk striping and parity checks to provide both increased performance and data protection. However, none of the disks in a RAID-5 setup is exclusively designated as a parity disk. For example, four disk drives' worth of data and parity information would be striped across five disks. If any of the disks fail, the other four can provide all needed data.

Although RAID-3 groups provide the same basic level of data protection, RAID-5 groups usually have better performance. When all of the parity information for a group is on a single disk, such as in a RAID-3 group, that disk can become a performance bottleneck on both disk writes during normal operation and disk reads when the group is recovering from a failed disk.

For more information on RAID technology, see *The RAID Book — A Source Book for RAID Technology.*

CPU and memory

Rather than providing redundant hardware or other continuous operation capabilities, high-availability CPU and memory features focus on error detection and failure diagnosis. The objective of these features is to protect the integrity of the system's data, so that the system can recover quickly. Features such as Error Correcting Code (ECC) memory provide this data integrity in a manner similar to RAID technology.

Some systems that have more than one of a component type, such as CPUs, still crash when one of the components fail. However, these systems can often be quickly restarted without the failed component. For example, many high-availability multiprocessor systems automatically reboot when a single processor fails and run with the remaining processors until the failed processor is repaired. This minimizes downtime and brings critical applications back on line quickly.

OS and system software

High-availability systems add features at the system software level to maintain data integrity. These features include the following:

File system logging — The ability to log changes to the file system helps ensure that files are always in a usable state. When a system fails, transaction log files ensure that files are saved in their last consistent state. This enables the system to avoid both corruption and loss of data. At most, the last transaction before the system failure is lost.

Process isolation — Process isolation ensures that errors in specific software modules cannot propagate to other areas. Such propagation might cause other software, or the entire system, to crash. Since it would be costly to isolate all errors in a system, high-availability systems have concentrated error isolation on highly-used components such as file systems.

Transaction processing monitors — Transaction monitors can detect and manage application failures. When a system fails, uncommitted transactions can be rolled back and databases returned to their last known state. Transactions can also be rerouted to a new system.

On-line system administration — The ability to perform both routine system administration tasks, such as backups, and system diagnostics on-line can reduce both planned and unplanned downtime.

Multiple system high availability

Using multiple hosts enables high-availability systems to overcome a lack of CPU fault-tolerance. Multisystem high-availability environments can greatly reduce recovery time from system failure.

There are three levels of multiple system high availability:

- Operator-initiated system failover
- Machine-initiated system failover
- Clustering

These levels are usually inclusive of each other. That is, a system providing machine-initiated failover also includes operator-initiated failover.

Operator-initiated failover

System failover involves switching accounts and applications from a primary system that has failed to a backup system. The failover process ensures that there is no data loss during either the failure or the switch to the backup system. Applications are restarted on the backup system. If a transaction monitor is on the system or the application supports it, transactions can be restored to the state they were in when the primary system failed.

Operator-initiated failover is the least complex to provide. During an operator-initiated failover, operator intervention is required to start the process that initializes disks from a failed system, cleans up the file system, and restarts applications. From a hardware

perspective, failover could be provided through dual-ported I/O bus features that enable two or more systems to connect to a set of common disks.

Machine-initiated failover

Machine-initiated failover automates the failover process. There are two types of machine-initiated failover:

- Programmable failover
- Automatic failover

Both types of failover use a *heartbeat* mechanism to communicate between the primary and backup systems. The heartbeat is a small daemon program that one of the systems uses to communicate its status to the other system. If the heartbeat program is running on the primary system, it sends out a message to the backup system at set intervals. If the backup system does not receive a message at the designated time, it initiates a preset series of actions to determine if the primary system has failed. If the backup system cannot reestablish communication with the primary system, the backup system begins the failover process.

In programmable failover, the user has an Application Programming Interface (API) or programming model which is used to write a failover application. This application sets out the steps the the backup system should perform to take over the role of the primary system.

In automatic failover, the failover process is more seamless. Applications register with the failover monitor and are automatically restarted during the failover process.

Clustering

System clustering provides a mechanism where one logical view of resources is seen across multiple systems. Clustering allows system facilities, such as file systems, to be shared across multiple systems as one logical resource. Shared facilities can include system management, backup, network services, and even computing resources.

The benefits of clustering include:

- Reliability — continuous operation if one of the systems in a cluster fails
- Performance — throughput can increase proportionally as more systems are added to the cluster

Data General's approach to providing high availability

Data General takes a system approach to providing high availability. Each component that goes into a system is developed with high availability in mind. These components work together to maximize system availability and minimize recovery time from system failure.

The steps needed to provide a high level of availability to a system are shown in Figure 1–3.

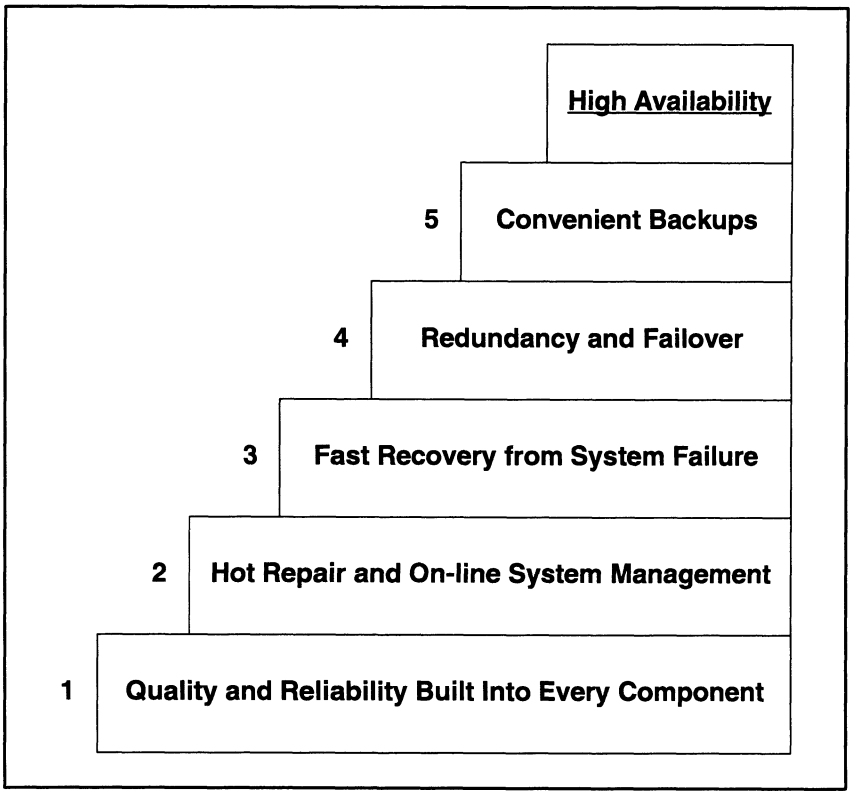


Figure 1–3 The five steps of high-availability protection

As Figure 1–3 illustrates, high availability must be provided at every level of a system, from the design of system components to backup strategies. The following sections discuss these steps of protection in more detail.

Reliable and quality design

The first line of defense in high availability is the reliability of a system's components. Data General's hardware and software components combine high-quality standards with innovative designs to produce cost effective high-availability systems.

Hardware capabilities

AViiON® hardware components provide many high-availability capabilities. System memory on most AViiON systems contains error-correcting code that performs single-bit error correction and double-bit error detection. In addition, some systems have many built-in redundant capabilities, such as power and cooling components. Additional redundancies, such as I/O controllers, can be added to the base system. Uninterruptible power supply systems (UPS) can also be added to provide protection from power failures and similar problems.

Software capabilities

DG/UX system software also has a number of high-availability capabilities. These capabilities include:

- Kernel design that reduces the number of software panics that can cause the system to fail
- Automatic sealing of corrupted file systems
- Dynamic remapping of bad disk sectors
- Dynamic rerouting of network traffic around a failed network controller on a single system or from a failed system to a backup system
- Automatic detection and attempted recovery from disk, SCSI bus, and SCSI controller errors
- Software disk mirroring

On-line repair and system management

The next line of high-availability defense is the ability to repair and manage system components on line. This reduces both planned and unplanned system downtime.

Hardware capabilities

Some AViiON servers and all CLARiiON™ disk-array storage systems allow the replacement of many components while the system is running. Components that you can replace under power include disks, power supplies, and cooling components. If a redundant component exists for the one being replaced, user applications may continue running while you replace the failed component.

Software capabilities

The DG/UX system enables you to perform many system administration tasks on line. These include backups, disk

mirroring, file system manipulation, and communication path selection.

AV/Alert℠, Data General's system monitoring and diagnostic package, is also included with most AViiON servers. AV/Alert detects and reports existing or impending hardware and software problems before they result in extended downtime. The package automatically alerts Data General service personnel of any problems it detects.

Other software packages can be added to AViiON systems to reduce downtime. For example, the TUXEDO transaction processing monitor enables on-line installation of its software upgrades, in addition to the package's other capabilities.

Fast recovery from system failure

If a system component fails, it is crucial that it be brought back on line as quickly as possible. To do this, the system should be able to detect that a failure has occurred, automatically start system recovery, and expedite the recovery process where possible.

Hardware capabilities

AViiON hardware can reboot the system if the operating system fails or hangs. If the operating system panics and shuts down the system software, AViiON firmware can be set up to automatically perform a cold boot to start the system recovery process. If the operating system goes into an infinite software loop, the system's watchdog timer detects this and automatically reboots the system.

Also, if a system component fails during power up, the system deconfigures the failed component, removes it from system use, and continues the power up. The system then attempts to boot and continue processing without the failed component.

The RAID capabilities of CLARiiON disk arrays enable fast recovery from disk failure. If a disk in a RAID-1 pair fails, its mirror immediately takes over for it. Similarly, if one of the disks in a RAID-3 or RAID-5 group fails, its data is automatically reconstructed from the group's parity information. If a hot spare is built into the configuration, the data from the failed drive is rebuilt automatically on the spare.

Software capabilities

The DG/UX system can automatically recover from system panics and power outages. System recovery capabilities include:

- Automatic, fast system dumps to disk
- Automatic restart and reboot from system panics and power failures
- Fast system initialization
- Selective file checking rebuilds open files
- Fast recovery file systems
- Watchdog timer

Component redundancy and system failover

In the event of a major failure requiring extended downtime, a system should have the capability to either bypass the point of failure or continue operations on an alternate system.

Hardware capabilities

As mentioned before, AViiON systems have many built-in redundant components. If a power or cooling component fails, the remaining units automatically compensate for the missing unit. The system also attempts to compensate for the failure of memory modules, CPUs, and I/O controllers. CLARiiON disk-array storage systems have similar capabilities.

Software capabilities

In the case of component failure, DG/UX supports features such as multi-path disk and LAN I/O. To compensate for major system failure, The DG/UX system has both operator-initiated failover and machine-initiated failover capabilities. Other capabilities include IP takeover and NFS failover which enable a secondary system to also take over the Internet address and NFS services of a failed server.

Adding the TUXEDO transaction processing monitor to the system software enables a secondary AViiON system to completely take over for a failed primary system in a matter of minutes. Applications are automatically restarted with no loss of either data or transactions.

In addition to these capabilities, DG/UX supports the ORACLE Parallel Server™ through the AViiON Distributed Lock Manager (AV/DLM). AV/DLM enables a cluster of AViiON servers to share a single ORACLE® database. This database is continuously available as long as one node in the cluster remains operational.

Backups

If data is irretrievably lost from a disk, a system should be able to restore that data. Reliable, convenient backup and restore

capabilities are essential as the last line of defense in high-availability systems.

Hardware capabilities

The CLARiiON tape-array storage system provides fast, unattended highly available backup capabilities for AViiON systems.

Software capabilities

The DG/UX system enables you to do on-line backups. If you operate in a networked environment, the Legato NetWorker package helps you avoid data loss. NetWorker offers convenient, network-based back-up and recovery services to a wide variety of machines.

End of Chapter

2 Hardware components

The hardware components of Data General's systems have many built-in high-availability features. This chapter describes the features of the following system components:

- AViiON computer units
- CLARiiON disk-array storage systems
- CLARiiON tape-array storage systems
- Uninterruptible power supply systems

AViiON computer units

AViiON computers are designed to minimize the impact of hardware problems. With some interruption and possible system degradation, AViiON computers can continue processing despite CPU, memory, fan, power supply, and input/output failures.

IMPORTANT: The features described in this section may not be available in all AViiON computer units. Consult Table 2–1 and the documentation provided with your AViiON computer to determine which features are supported on that unit.

Table 2–1 High-availability features of AViiON servers

High-availability feature	Other AViiON servers	AViiON 8500 servers	AViiON 9500 servers
Automatic reboot	Yes	Yes	Yes
Boot on startup error	Yes	Yes	Yes
Cooling unit compensation	No	Yes	Yes
Power supply compensation	No	No	Yes
Central processor unit deconfiguration	No	Yes	Yes
Memory deconfiguration	No	Yes	Yes
Input/Output controller deconfiguration and failover	No	Yes	Yes

In addition to the hardware-based high-availability features described in this section, most AViiON servers support the AV/Alert

system diagnostic software. The AV/Alert software detects and reports hardware and software problems automatically. If it detects an existing or impending system problem, AV/Alert asks for remote assistance from Data General support centers. See Chapter 3 for more information on the AV/Alert system.

Memory features

Most AViiON system memory contains error-correcting code (ECC) that performs both single-bit error correction and double-bit error detection. This helps the system recover from many memory errors.

Boot features

AViiON servers have the following boot-related features:

Automatic reboot

Whenever the operating system panics and shuts down the system software, AViiON firmware performs a cold boot. The operator can disable and enable this feature, and set or alter the boot path.

Boot on startup error

Whenever a component fails during startup, AViiON servers deactivate the failed component and remove it from system use. The system then attempts to reboot and continue operation without the failed component. Depending on which component failed, there may be some system degradation. The operator can disable and enable this feature, and set or alter the boot path.

If the server has more than one input/output controller (IOC) board, the boot on error feature further enables the backup board to take over when the primary board fails.

Component failure features

AViiON servers have many features related to component failure. In some cases, the system continues operation when a component fails. In others, the system goes down, but the boot on error feature deconfigures the failed component during reboot.

AViiON servers have the following component failure features:

Cooling unit compensation

Upon fan failure, the remaining units in an AViiON computer's fan tray increase speed to compensate for the failed fan by providing

more cooling per unit. This high-availability feature requires no operator intervention. Your operating system, or power-up diagnostics, sends a warning message to the system console. Systems with the AV/Alert software enabled send a message packet to a remote service center.

If your system determines that more than one fan has failed, it will shut down the entire computer unit after sending the appropriate warning messages to the system console and AV/Alert service center.

Power supply compensation

If one of an AViiON unit's power supplies fails, the remaining supply compensates for the failure and powers the entire system. This high-availability feature requires no operator intervention. Your operating system, or power-up diagnostics, sends a warning message to the system console. Systems with AV/Alert service enabled send a message packet to a remote service center.

Central processor unit deconfiguration

AViiON servers include one or more system boards. Each system board contains one or more central processing units (CPUs). By default, an automatic or operator-initiated power cycle (cold boot) after the failure of one or more CPUs causes power-up diagnostics to deconfigure any failed units. Systems with AV/Alert enabled send a message packet to a remote service center. Power-up firmware attempts to boot the operating system. If that fails, the system attempts to reach the System Control Monitor (SCM) program to allow remote assistance and on-site diagnostics execution. The system displays a deconfiguration message, warns the operator that the system is degraded, and asks whether startup should continue. DG/UX boot messages further inform the operator of the number of processors found.

If every CPU in a system appears to fail one or more diagnostic tests, the AViiON server attempts to continue with a suspect CPU. This process can initiate an AV/Alert call to a remote service center, and the operator can run system diagnostics from the SCM. If the system board itself fails, an AViiON computer will halt processing entirely.

Memory deconfiguration

A cold boot after the failure of one or more memory modules causes power-up diagnostics to deconfigure any failed modules. Systems with AV/Alert service enabled send a message packet to a remote service center. The system displays a deconfiguration message,

warns the operator that the system is degraded, and asks whether startup should continue. Power-up messages further inform the operator of the number of memory modules found. DG/UX boot messages display the total amount of memory in use.

If a memory controller board fails, power-up firmware deconfigures the entire board and attempts to continue — using the remaining memory controller, if present. For optimal memory high availability, memory modules can be distributed evenly across two memory controller boards. Although this configuration may create a slight performance impediment, it ensures reasonable memory configuration in case an entire controller fails.

If a system includes only one memory controller board, and that controller fails, the AViiON computer unit will continue processing to the SCM prompt. However, a system without memory cannot boot the DG/UX operating system.

Input/Output controller deconfiguration and failover

If an Ethernet LAN controller fails, an AViiON server reacts as described above for other system components. That is, a cold boot causes power-up diagnostics to deconfigure the failed controller. Systems with AV/Alert service enabled send a message packet to a remote service center. The system displays a deconfiguration message, warns the operator that the system is degraded, and asks whether powerup should continue. Power-up messages further inform the operator of the number and type of IOC components found.

IOC failure in systems configured with two IOC boards initiates one of the following scenarios, depending on which IOC failed and the specific configuration of each board. Notice that in either case, the system loses whatever unique functions were assigned to the primary IOC board.

- If the failed board is the secondary controller board, power-up firmware simply deconfigures that board and its dependent controllers. As in other component deconfigurations, the system will display console messages and send AV/Alert packets to a support center.
- If the primary IOC board fails, a cold boot causes power-up diagnostics to deconfigure the controller board. As in other component deconfigurations, the system displays console messages and initiates an AV/Alert call to its support center. However, in this case, deconfiguring the primary IOC board automatically implies failover to any second IOC, as described below.

During startup, firmware assigns the base address for the primary IOC board to the first active IOC board in the system, regardless of its physical slot position. After deconfiguration of a failed primary board, the secondary IOC board assumes the controller identity and function of the primary board wherever the two boards shared configuration. Minimally, this includes the asynchronous ports, internal SCSI controller, and first LAN controller in your system.

AViiON computer units can initiate failover for the following IOC components:

- Asynchronous ports (duarts)
- SCSI controllers/adapters
- LAN controllers

For more information

For more information on the high-availability features of specific AViiON computer models, see the hardware manual or manuals provided with the unit.

CLARiiON disk-array storage systems

The CLARiiON disk-array storage system provides a compact, high capacity, high-availability source of disk storage for AViiON computers. It offers a large capacity of traditional or high-availability disk storage in multiple disk-drive modules that can be replaced while the power is on.

Along with its high-availability features, the storage system offers great flexibility: a host computer can run multiple storage systems, or two host computers can use different disks within a single storage system. An AViiON system can support up to 32 disk-array storage systems (depending on the model).

The storage system connects to the computer by a SCSI-2 (small computer system interface) differential bus. The host computer runs the storage-system disks through a SCSI-2 adapter just as if the disks were standard disk units.

There are two types of CLARiiON disk-array storage system: the Series 2000 with 20 disk slots and the Series 1000 with 10 disk slots. Each type of array comes in either a deskside model, in its own cabinet, or a rackmount model, built into the computer chassis. The Series 2000 array is shown in Figure 2–1 and the Series 1000 is shown in Figure 2–2.

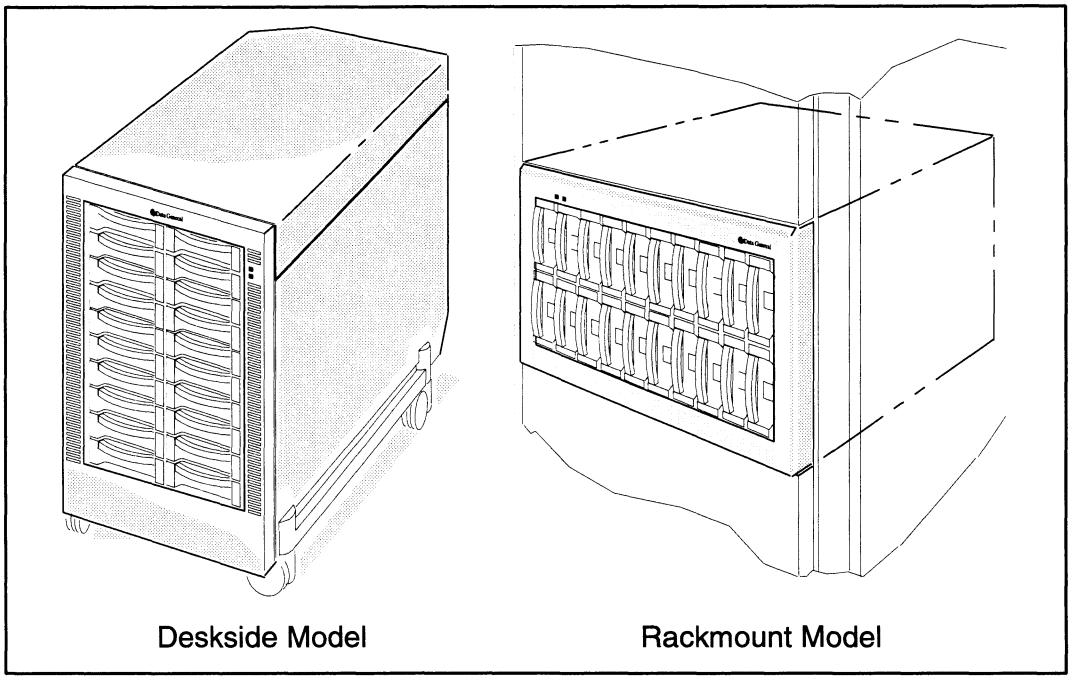


Figure 2–1 Series 2000 disk-array storage system

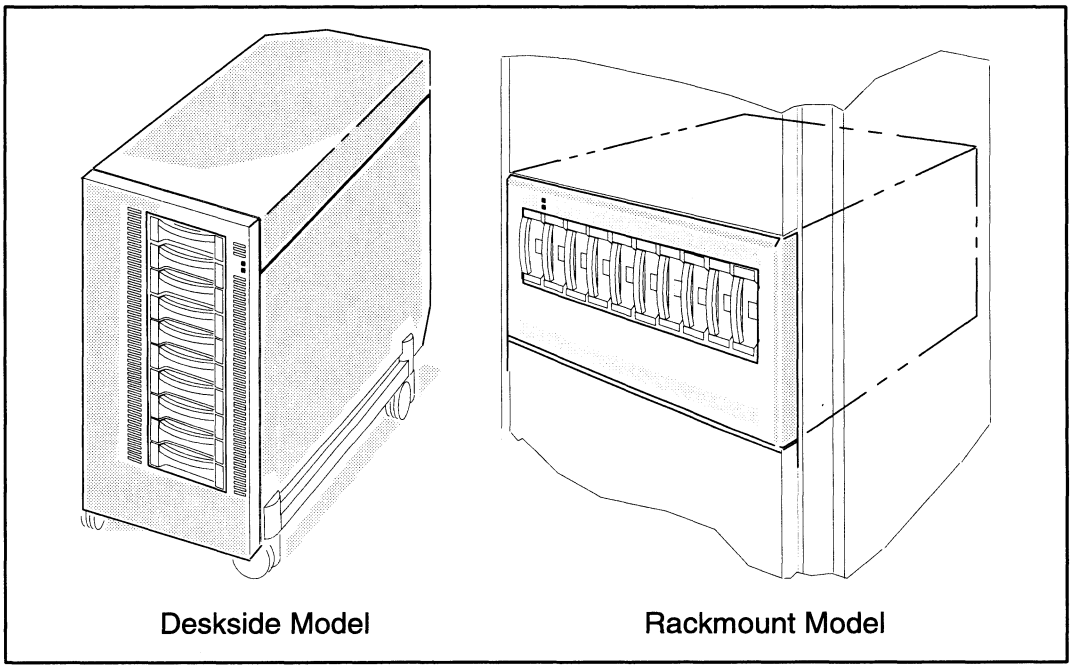


Figure 2–2 Series 1000 disk-array storage system

Storage system components

The storage system contains the following hardware:

- Cabinet or chassis

- One or two disk-array storage-control processor modules (SPs)
- From five to twenty disk modules
- One to three voltage semi-regulated converter modules (VSCs), also known as power supplies
- One fan module
- Optional additional SIMM memory and backup battery for the SPs to provide storage-system caching

Cabinet or chassis

The deskside model of the storage system has its own cabinet. The rackmount model is mounted in the computer chassis. The cabinet contains the slots for disk modules, SPs, and VSCs. The disk modules are in the front of the storage system. The SPs, VSCs, and fan module are accessible from the back.

Disk–array storage-control processor (SP)

The SP is a printed circuit board that resides in a slot in the storage system cabinet. It controls disk modules through a synchronous SCSI-2 bus. A series 2000 SP has five internal buses, each of which supports up to four disk modules. A series 1000 SP has two internal buses, each supporting up to five disk modules. A storage system can have up to two SPs.

With two SPs and other optional equipment, a CLARiiON disk array supports storage-system caching. Caching enables an SP to store modified information temporarily in local SP memory and write the information to disk at the most expedient time. This can enhance the performance of the storage system.

Disk modules

A disk module consists of a disk drive, a power regulator board, internal cabling, and a plastic carrier. Each module is self-contained and can be slid into slots in the front of the storage system. A Series 2000 supports five to twenty disk modules and a Series 1000 supports five to ten disk modules.

Voltage semi-regulated converter (VSC)

This is the essential element of the storage system power supply. A Series 2000 supports two to three power supplies and a Series 1000 supports one to two power supplies.

Fan module

This is the cooling unit for the storage system. It is a module, containing six fans, that is attached to the back of the unit.

High-availability features

CLARiiON disk-array storage systems have the following high-availability features:

- Redundant (replace with power on) components
- RAID technology
- Dual SP, adapter, and host configurations

Redundant components

The following components can be replaced without powering down the storage system:

- Disk modules

  If a failed disk module is part of a RAID group, it can be replaced while the storage system is running without interrupting applications.

- VSCs

  If a storage system has the maximum number of VSCs, it can survive the failure of one VSC. A failed VSC can be replaced while the storage system is running without interrupting applications.

- SPs

  A storage system with two SPs installed can continue full operation if one of the SPs fails. After an SP fails, the operator can transfer the failed SP's disks to the remaining controller and restart applications. The failed SP can then be replaced while the storage system is running without interrupting applications.

- Fan modules

  If one of the fans stops working, the other five speed up to maintain air flow. The entire fan module can be replaced while the storage system is running without interrupting applications.

RAID technology

The storage system supports RAID level 1, 1/0, 3, or 5 data redundancy. This technology provides redundant disk resources in disk-array and disk-mirror configurations.

RAID-1 — Two disk modules can be bound together as a mirrored pair. Data is written simultaneously to each of the disks. If either of the disks fails, the remaining disk continues providing data to applications.

Disks in a RAID-1 group are bound together by the storage system hardware and cannot be split into individual units. Disk mirroring can also be accomplished at the virtual disk level through the DG/UX operating system, as discussed in Chapter 3.

RAID-1/0 — From four to sixteen disk modules can be bound together as a RAID-1/0 group. These modules are bound together in mirrored pairs as in RAID-1. In addition to this data protection, the storage system uses disk striping to improve performance. A RAID-1/0 group can survive the failure of multiple disk modules, as long as one module from each mirrored pair survives.

RAID-3 — Five disk modules can be bound together as a RAID-3 group. Four of the disk modules are used for data storage, the fifth disk contains parity data for the other modules in the group. If one of the disk modules fails, the storage system uses the group's parity information to continue operations without interruption.

When a failed module is replaced, the SP rebuilds the group using the information stored on the working disks. Performance may suffer while the SP rebuilds the data or parity on the new module. However, the storage system continues running and provides users access to all data stored in the group.

RAID-5 — Generally five disk modules, but from as few as three to as many as sixteen modules, can be bound together in a RAID-5 group. All of the disk modules in the group are used for storing both data and parity information. As with RAID-3 groups, the storage system uses the parity information to continue operating if one of the disk modules fails.

When a failed module is replaced, the SP rebuilds the group using the information stored on the working disks. Performance generally suffers while the SP rebuilds the data and parity on the new module. However, the storage system continues running and provides users access to all data stored in the group.

In addition to data protection, a RAID-5 group uses disk striping to improve performance. Disk striping enables the system to write to or read from multiple disk modules simultaneously.

Dual configurations

The storage system can have redundancies built into its configuration to provide greater levels of high availability. Each level of dual configuration includes the preceding levels.

The storage system can have the following dual configuration:

- Dual SP configuration

In a dual SP configuration, a backup SP is available if an SP fails. If one SP fails, the system operator can transfer control of the failed SP's disks to the remaining SP, and then restart applications.

- Dual SCSI-2 adapter channel configuration

  This configuration provides the highest availability when the storage system is connected to one host computer. In a dual adapter configuration, the host computer has one SCSI-2 adapter with each of its channels connected by SCSI-2 bus to an SP. If either an SP or a channel fails, the operator can transfer control to the remaining channel, and then restart applications.

- Dual host configuration

  In a dual host configuration, two host computers share access to a single storage system. Each host has a SCSI-2 adapter connected to one of the SPs in the storage system. If one of the hosts fails, the operator can use the other host to access the storage system.

  For the highest availability, each host can have two SCSI-2 adapter connections to an SP. This is called a *dual-initiator* configuration. If a host fails in this case, the other host can take over the failed host's virtual disks, restart applications, and resume user processing in a matter of minutes. This is accomplished through the DG/UX system's failover features. See Chapter 3 for more information about failover.

For more information

For more information on the high-availability features of CLARiiON disk-array storage systems, see *Configuring and Managing a CLARiiON® Disk-Array Storage System — DG/UX™ Environment* and the hardware manual provided with the unit.

CLARiiON tape-array storage systems

The CLARiiON tape-array storage system provides a compact, high-capacity source of tape backup for computer systems or disk-array storage systems. It offers up to 30 gigabytes of tape storage, with as many as seven tape drives.

This storage system offers great flexibility. A host computer can run two storage systems on each SCSI-2 (small computer system interface) adapter channel, or two host computers can control different tape drives within a single storage system.

The storage system connects to the computer by a SCSI-2 differential bus. The host computer accesses the storage-system tapes through a SCSI-2 adapter just as if the tapes were standard tape units.

There are two models of CLARiiON Tape-Array Storage System: the deskside model, in its own cabinet, and the rackmount model, built into the computer chassis. Both models are shown in Figure 2–3.

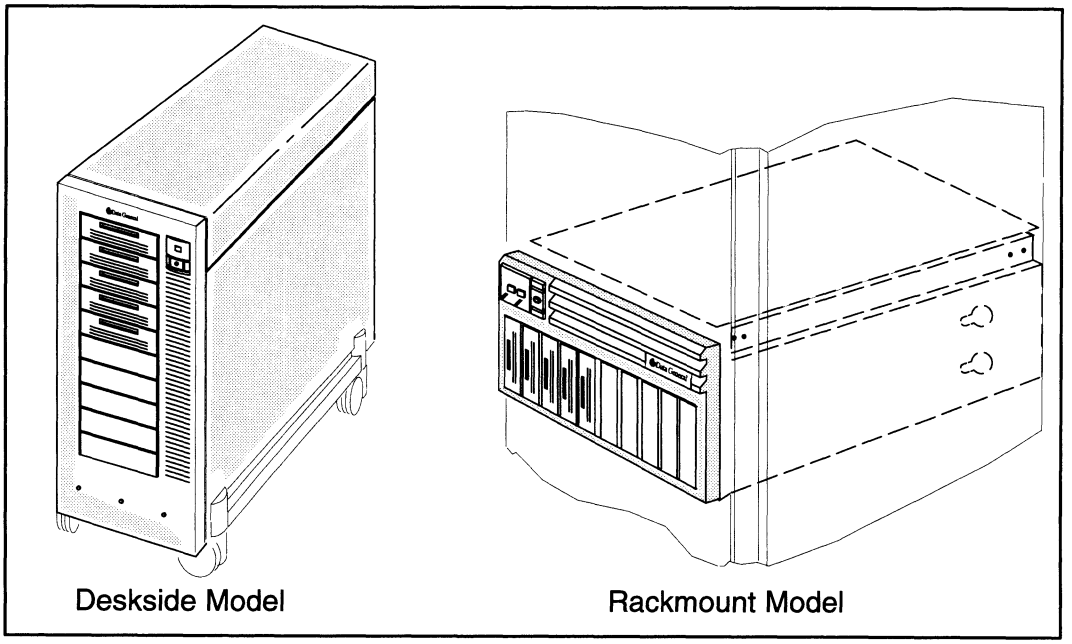


Figure 2–3 CLARiiON Tape-Array Storage System, deskside model and rackmount model

Storage system components

The tape-array storage system contains the following hardware:

- Chassis
- One tape-array processor (TAP)
- Five to seven tape drives
- Two power supplies
- Two fans

Cabinet or chassis

The deskside tape-array storage system chassis contains the compartments for tape drives, TAP, and power supplies. The tape drives are in the front of the system. The TAP, power supplies, and fans are accessible from the side of the deskside model.

Tape-array processor (TAP)

The TAP is a printed-circuit board that resides in the storage-system chassis. It controls tape drives through a

synchronous SCSI-2 bus. The TAP has an internal bus which supports up to seven drives.

Tape drives

A storage system has slots for up to seven half-height drive assemblies. A full-height drive can be installed in two slots. The storage system can support tape drives, CD-ROM drives, and digital audio tape drives.

Power supplies

Power supplies are self-contained units that convert AC line voltage into 5 and 12 volt DC required to power the storage systems components. Each storage system has two power supplies.

Fans

Each storage system has two 5-inch fan assemblies that attach to the rear of the chassis.

High-availability features

CLARiiON tape-array storage systems have the following high-availability features:

- Convenient backups
- Tape drive configurations
- Dual adapter and host configurations

Convenient backups

Each storage system can back up as much as 30 gigabytes of data. Using multiple tape drives makes the backups fast. Tape striping, where the system reads from or writes to multiple tape drives at the same time, can be used to make backups even faster. Also, backups can be unattended.

Tape drive configurations

A group of tape drives can be bound together to provide data protection in a way similar to RAID technology. The storage system supports the following drive configurations:

Mirrored pairs — Two drives are bound as a redundant tape group. In this configuration, the second drive in the group becomes a mirror image of the first.

Redundant tape groups — From two to seven drives can be grouped, but generally five drives are bound together into a redundant tape group. Redundant tape groups use tape striping to improve performance, but also write parity information to each of the tapes. If one of the tape drives or the tape media fails, the group continues operating and reconstructs any lost data with the parity information on the remaining drives in the group.

Dual configurations

The tape array storage system can have redundancies built into its configuration to provide greater levels of high availability. Each level of dual configuration includes the preceding levels.

The storage system can have the following dual configuration:

- Dual SCSI-2 adapter channel configuration

  In a dual adapter configuration, the host computer has two SCSI-2 adapters connected to the storage system's TAP. If an adapter fails, the operator can access tape drives by powering down the storage system and rebooting the host computer. This enables the second SCSI-2 adapter to access the tape array.

- Dual host configuration

  In a dual host configuration, two host computers share access to a single storage system. Each host has a SCSI-2 adapter connected to the tape array's TAP. If one of the hosts or SCSI-2 adapters fails, the operator can transfer control of the storage system and reboot the working host to access the storage system.

For more information

For more information on the high-availability features of CLARiiON tape-array storage systems, see *The CLARiiON™ Tape-Array Storage System with the DG/UX™ Operating System*.

Uninterruptible power supply systems

Uninterruptible power supply systems (UPS) protect computer equipment against electrical anomalies such as power surges, brownouts, improper grounding, and load imbalances.

The DG/UX operating system works with a UPS system to check the status of both the line power supply and the backup battery. If there is a power failure, UPS systems have backup batteries to keep computer systems running for a time. The DG/UX system keeps

track of the status of the backup battery. If battery power runs low before line power is restored, the DG/UX system will bring the system down in an orderly fashion.

High-availability features

UPS systems have the following high-availability features:

- Data protection
- Orderly system shutdown

Data protection

Computer equipment protected by UPS systems are protected from many power problems that might otherwise bring the system down abruptly. Host systems are protected from electrical fluctuations such as power surges that might bring the system down. Also, host systems are protected from complete power failures. If the host system is brought down abruptly, data could be lost.

Orderly system shutdown

If power fails, UPS systems keep computer equipment running for a time. Should line power be restored before the UPS battery runs low, the host system suffers no downtime. If battery power runs low before line power is restored, the host system is brought down to single-user mode. Then, if line power is restored before all battery power is lost, the host system is automatically brought back up to its default run level. Finally, if battery power runs out, the host system powers down. In this case, the host is automatically rebooted once line power is restored.

For more information

For more information on UPS systems, see *Managing the DG/UX™ System*.

End of Chapter

3 Software components

The software components of Data General's systems have many high-availability features. This chapter describes the features of the following system components:

- DG/UX operating system
- AV/Alert Diagnostic Support System
- Other Packages

DG/UX operating system

The DG/UX system is an enhanced UNIX System V Release 4 operating system. Data General has enhanced the DG/UX system with features such as symmetric multiprocessing, BSD extensions, and threads. The DG/UX system also contains a number of advanced high-availability features that maximize system availability and minimize downtime should a failure occur.

Reliability built in

The DG/UX system is designed with reliability in mind. Very deliberate software engineering and feature integration make the operating system very reliable.

An operating system's reliability is determined by its mean time to failure (MTTF). This is the average time from when a kernel boots to when it panics or hangs. MTTF of the operating system excludes hardware failures, non-fatal errors, and similar problems. Note that the MTTF of a given operating system is based on a particular set of system usage characteristics. If these usage characteristics are changed, the system may have a different MTTF.

To ensure that the DG/UX system's MTTF continues to increase, Data General started the RELiiON project. The goal of RELiiON is to improve even further the overall quality of the development process that produces the operating system. These improvements in the production of the DG/UX system will yield an even more reliable operating system.

High-availability features

The DG/UX system has the following high-availability features:

- "Hardened" kernel
- File sealing
- On-line system administration
- Fast recovery from system problems
- Multi-path I/O
- Disk mirroring
- Disk failover
- IP takeover and NFS failover

"Hardened" kernel

The DG/UX kernel is designed to reduce the number of software panics that can bring a system down. This design works on two levels.

First, the code that goes into kernel components is well designed and rigorously reviewed. This greatly reduces the number of bugs in the code.

Second, wherever the kernel code might call for the system to fail, developers have tried to find alternatives to failure. For example, when a hardware device fails, some operating systems might go down. However, the DG/UX system either tries to restart the failed device or just fails the device instead of the system.

File system sealing

If a DG/UX file system ever becomes corrupted, that corruption cannot spread to other file systems and possibly panic the system. File system corruption can be caused by a hardware problem, such as a failed or damaged disk. Occasionally a software problem can corrupt a file system.

If a file system becomes corrupted, the DG/UX system automatically changes the file system's access privileges to read only. If the read-only file system continues to generate corruption errors, the DG/UX system seals the file system so that it cannot be accessed. After the file system is sealed, the operator usually can use the **fsck** utility to repair the file system without taking the system down.

On-line system administration

The DG/UX system has a user interface for system administration called **sysadm** that has both a graphical user interface (GUI) and

an ASCII interface. This system administration tool is easy to use, and provides help messages and step-by-step prompting. These features help reduce operator errors. **sysadm** is organized in a menu hierarchy.

Figure 3–1 shows the initial GUI **sysadm** window.

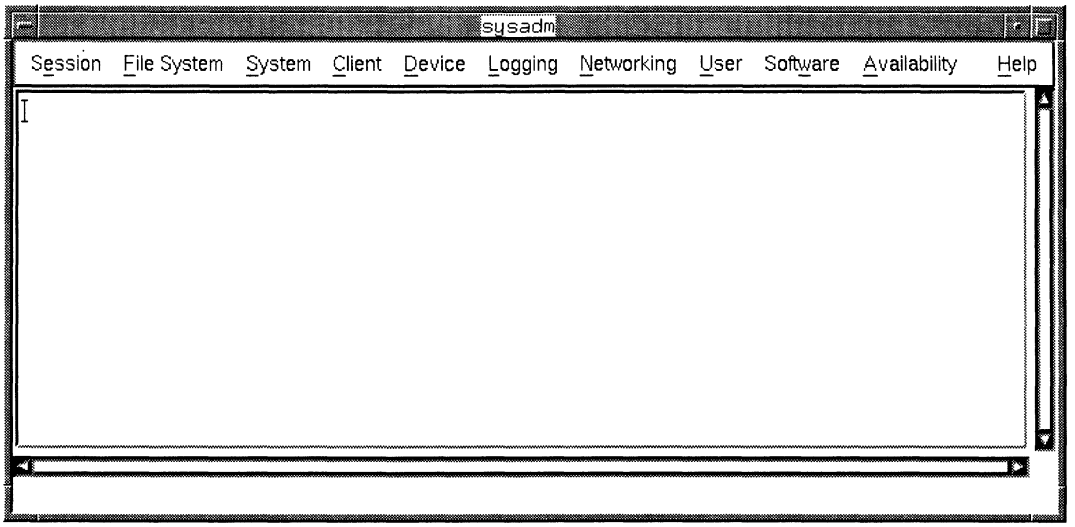


Figure 3–1 Sysadm interface

The DG/UX system enables administrators to do many operations on line that might otherwise require application interruption or system downtime. These operations include the following:

- Storage management
- Communications controller restart
- Dynamic bad block remapping

On-line Storage Management — With the DG/UX System 5.4, Rev. 3.00, the DG/UX system uses On-line Storage Management (OSM) to manage disk resources. With OSM you can perform many disk management tasks on line that formerly required taking disks off line. This means that users and applications have uninterrupted access to the data on any disks that are being manipulated.

OSM uses virtual disks instead of logical disks for disk management. Virtual disks are much more versatile than logical disks, giving administrators more flexibility.

Figure 3–2 shows the OSM **sysadm** menu.

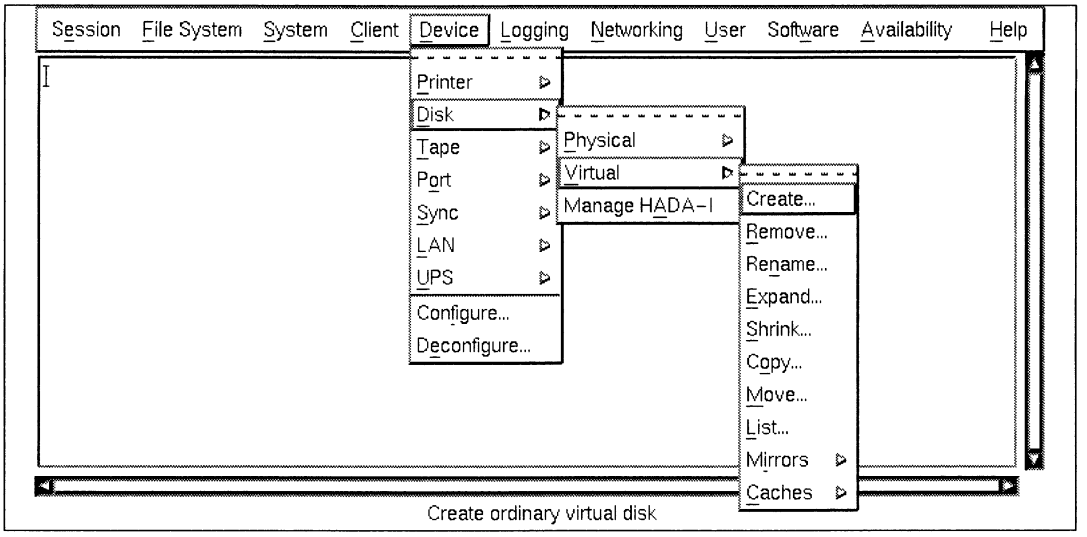


Figure 3–2 OSM menu

OSM enables administrators to perform the following disk management operations on line:

- Rename disks

  You can rename a virtual disk on line with a single command. Prior to OSM, logical disks could not be renamed, though the same result could be achieved off-line through a long series of commands.

- Move and copy disks

  You can move or copy a virtual disk to a new disk on line. After creating the destination disk for the move, the contents of the source disk can be moved or copied to the destination disk in a single step. As with renaming disks, previously moving or copying a logical disk required a lengthy operation off-line.

- Expand disks

  You can expand virtual disks on line. Users never lose access to the data on the disks. Before OSM, logical disks had to be taken off-line for this operation.

- Perform advanced disk mirroring

  OSM contains advanced disk mirroring features. Virtual disks can have multi-image mirrors created from any combination of virtual disk types. This enables administrators to perform a number of tasks on line, such as recovering from failing physical disks. With logical disks, only full logical disks could be mirrored. OSM enables you to mirror individual pieces of a virtual disk.

- Perform on-line backups

  Through the use of disk mirroring, you can backup any virtual disk on line. Users never lose access to the data on the disks being backed up. However, an application's data must be in a consistent state before the data is backed up.

See *Managing Mass Storage Devices and DG/UX™ File Systems* and *Managing Virtual Disks with On-Line Storage Management (OSM)* for more information on OSM.

Dynamic bad block remapping — While rare, some disk drives may have media defects. When an administrator first formats a disk, the formatting routines mark as "bad" any sector that has a defect. However, new defects can appear as the disk is used, which can make a sector unreadable.

When the DG/UX system detects a flaw on a disk, it flags the block (a 512-byte section of the disk) as bad. If a write operation was being performed when the bad block was found, the DG/UX system finds a good block to replace the bad block. The operating system automatically takes care of redirecting reads and writes intended for the bad block to the remapped block. This feature requires no operator intervention and happens on line.

Restart of communication controllers — If a communication controller fails, the DG/UX system can restart it on line. This helps ensures that there is no system downtime due to controller failure. See *Managing the DG/UX™ System* for more information.

Fast recovery from system problems

The DG/UX system has many features that enable the system to recover quickly from any panics or hangs. These features include the following:

- Operatorless dumps to disk
- Automatic reboot
- Fast recovery file systems
- Fast system initialization
- Watchdog timer

Operatorless dumps to disk — Whenever the DG/UX system panics or hangs, the system memory contents at the time of the failure can be written to a file. This file is called a *system dump* and is used by the Data General Customer Support Centers to investigate the cause of the system failure.

The DG/UX system can be set to automatically take a system dump whenever a panic occurs. These automatic dumps can be sent to a

tape device, but that requires the operator to always have a tape ready in the device. The dumps can also be sent to a designated virtual disk. This ensures that a fast system dump is taken after a system failure. As soon as the dump is complete, the operator can set the DG/UX system to automatically reboot. The disk dump can later be copied to tape for analysis.

Automatic reboot — If the system panics or hangs, or if there is a power failure, the operator can set the system beforehand to automatically reboot on a default or designated boot path. Also, the operator can set up the system to send mail to designated users whenever the system reboots. This is useful if the system automatically reboots in the operator's absence.

Fast recovery file systems — The DG/UX system uses the **fsck** utility to check and repair damaged file systems. Whenever the system goes down abnormally, the DG/UX system runs **fsck** to check the integrity of file systems and see if any are corrupt and need repair.

To reduce the amount of time **fsck** needs to check a file system, operators can designate any file system as a *fast recovery file system*. For fast recovery file systems, the system keeps an *intent log* containing records about file system updates. If a failure occurs and the operator needs to fix the file system, **fsck** uses the log to speed the process of verifying and restoring file system integrity. This logging minimizes the amount of time the file system is unavailable to users. If the application using a fast recovery file system is write intensive, runtime performance will suffer somewhat.

Fast system initialization — Whenever the DG/UX system reboots, the system initializes quickly. Within a few minutes, the system is back up to **init 3**.

Watchdog timer (WDT) — For this feature, the DG/UX system works in conjunction with AViiON hardware to recover from system hangs. The operator must configure the **wdt()** pseudo-device in the DG/UX kernel to communicate with the WDT hardware on the AViiON. Note that not all AViiON computers have the WDT hardware.

The kernel communicates with the WDT about every second to let the WDT know that the operating system is still running. If the kernel hangs, the WDT will expire. In this case, the server can be configured to automatically reset and reboot the system.

In addition to using the **wdt()** pseudo-driver in conjunction with the hardware watchdog driver to detect and recover from operating system or hardware hangs, you can also use the driver to recover

from hangs in user applications through the **failovermon** monitoring process. The **failovermon** monitor can communicate with the **wdt()** pseudo-driver to determine whether there is a hang in a user application. If such a hang exists, the watchdog timer can detect it and panic the system in 90 seconds or less. The system can then automatically reboot as described earlier.

See *Managing the DG/UX™ System* and the **failovermon**(1M) manual page for more information on these fast recovery features.

Multi-path I/O

The DG/UX system has two features that enable a system to automatically reroute I/O when an I/O path fails. These features include the following:

- Multi-path LAN I/O
- Multi-path disk I/O

Multi-path LAN I/O — Most types of LAN controllers in AViiON servers can be configured to support multi-path LAN I/O. For this feature, two LAN controller cards are connected to the same LAN (such as Ethernet) to form a primary and backup pairing. If the primary LAN controller fails, the DG/UX system automatically switches from the primary LAN controller to the backup LAN controller card. This switch is transparent to any applications running on the server. This ensures that a system does not lose network services when a controller fails. The backup LAN controller is redundant and cannot be used by network software such as TCP/IP and X.25 while it is backing up the primary LAN controller.

Figure 3–3 shows the multi-path LAN I/O **sysadm** menu.

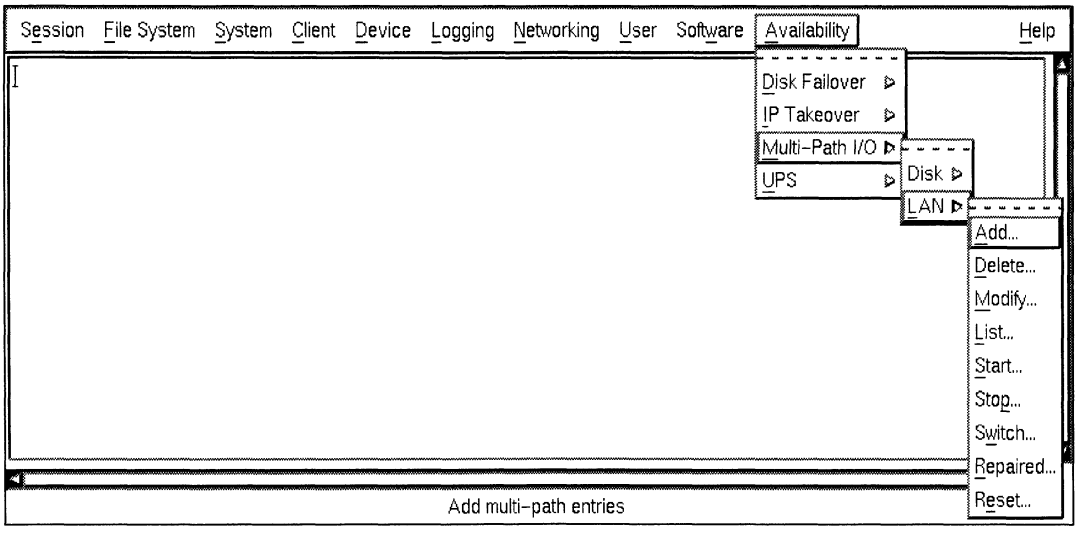


Figure 3–3 Multi-path LAN I/O menu

Multi-path disk I/O — Similar to multi-path LAN I/O, multi-path disk I/O enables a system with multiple paths to the same disk to automatically switch I/O to a backup disk path when the primary path fails. This ensures that a system does not lose access to the data on a disk when a path fails. Unlike multi-path LAN I/O, multi-path disk I/O can have up to three backup disk paths, and these paths do not have to be idle.

Figure 3–4 shows the multi-path disk I/O **sysadm** menu.

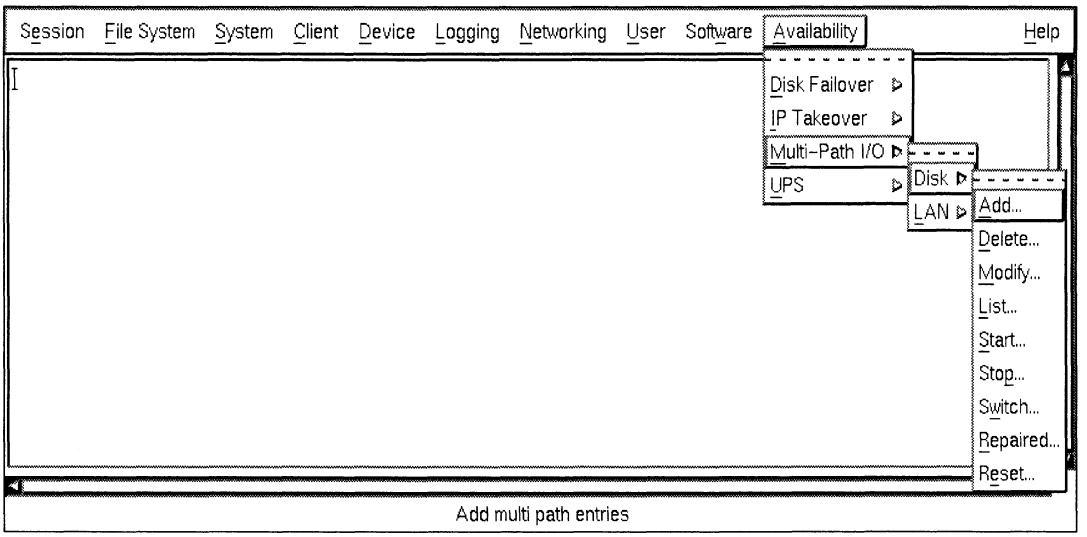


Figure 3–4 Multi-path disk I/O menu

See Chapter 4 for more information on these features.

Disk mirroring

The DG/UX system provides software disk mirroring. This is different from the hardware disk mirroring provided by the CLARiiON disk-array storage system. See Chapter 2 for more information about hardware disk mirroring.

Software disk mirroring involves setting up multiple redundant virtual disks that are all mirror images of each other. Therefore, all of the virtual disks contain the same data. Either two or three virtual disks can be set up as disk mirrors.

When a user or application writes to a file on a software-mirrored disk, the system duplicates the write operation on every image in the mirror. When a user or application reads from a file on a software-mirrored disk, the system selects one of the images to satisfy the read request.

Software disk mirroring provides the following high-availability benefits:

- Data availability

  Since the virtual disks making up the mirror all contain the same data, a single disk failure no longer interrupts user or application access to data. Setting up a three way mirror provides even greater data protection. As long as one disk in the mirror is available, the system automatically gets data from the remaining image and users experience no interruption in service.

- Data integrity

  Having multiple, identical images of a virtual disk's data greatly reduces the risk of losing data because of a hardware failure on one physical disk drive.

- Disk space conservation

  With software disk mirroring, administrators do not have to duplicate an entire physical disk worth of data. Instead, they can use virtual disks to mirror only those parts of the physical disk containing the critical data or files that need to be protected.

See *Managing Mass Storage Devices and DG/UX™ File Systems* for more information on software disk mirroring.

Disk failover

The DG/UX system supports disk failover between two AViiON servers set up in a *dual-initiator* SCSI-2 bus configuration. In a dual-initiator configuration, a CLARiiON disk-array storage system is connected to two AViiON servers. Each server can have two SCSI-2 adaptors, each of which are connected by SCSI bus to a storage-control processor (SP) in the CLARiiON. The servers can share SCSI buses.

Figure 3–5 shows an example of a dual-initiator configuration.

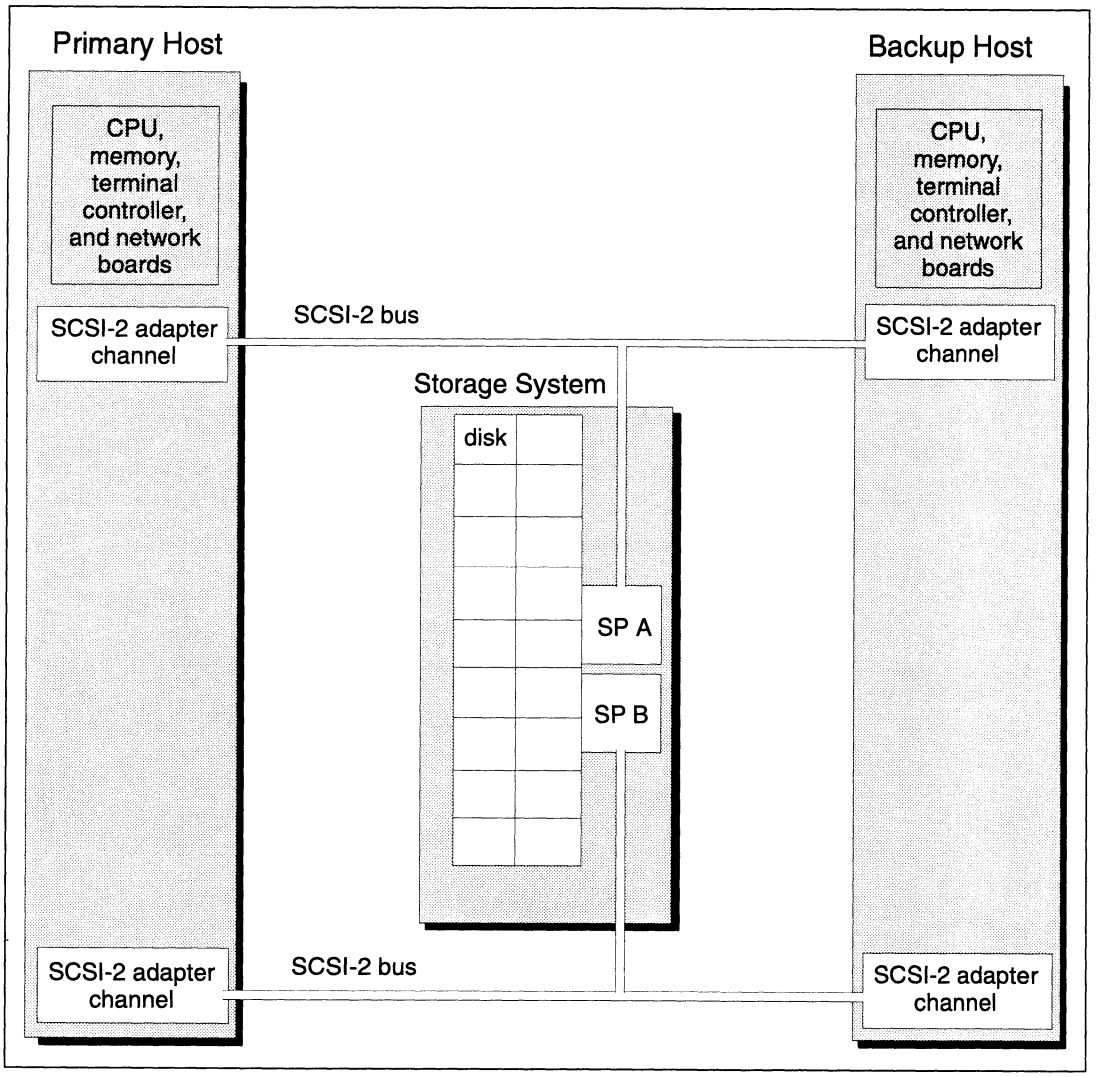


Figure 3–5 Dual-initiator configuration

When a physical disk is part of a dual-initiator configuration, only one host can "own" the physical disk. The server that has the disk open is the owner of the physical disk. Only the owner of the physical disk can directly access the contents of the physical disk, read or write directly to the virtual disks, or mount file systems on the virtual disks. The other system could access the contents of the disk indirectly, such as through NFS.

With a dual-initiator configuration, the operator can transfer control of a physical disk from one system to another. This reduces the time that data is inaccessible to users when a system fails or needs maintenance. When the primary host in a dual-initiator configuration has a problem, the operator can transfer ownership of the physical disk to the backup system and restart applications that were running on the failed disk. This is called *failover.*

The DG/UX system supports the following two types of failover:

- Operator Initiated Failover (OIF)
- Machine Initiated Failover (MIF)

Figure 3–6 shows the failover **sysadm** menu.

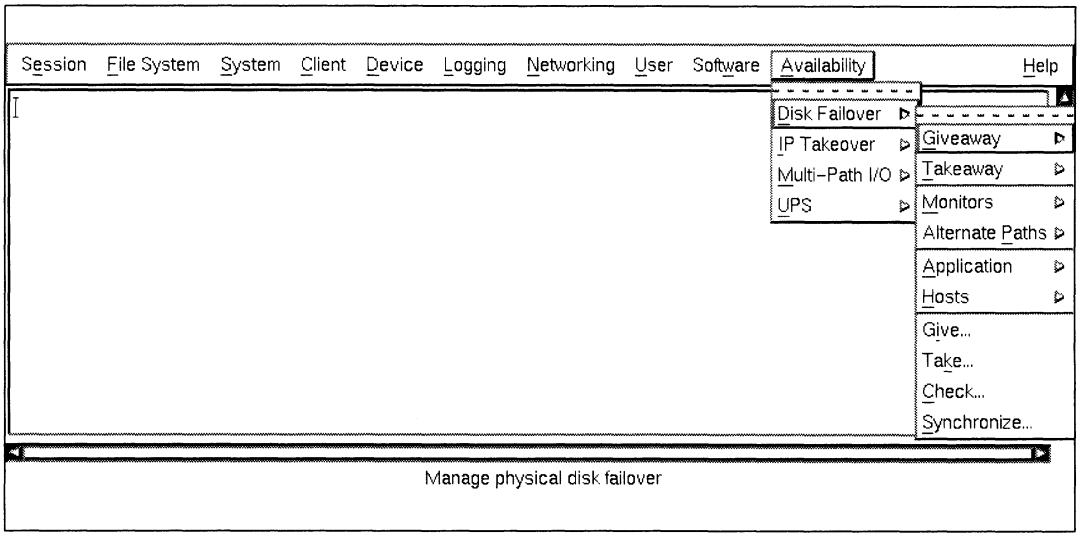


Figure 3–6 Failover menu

Operator Initiated Failover — Operator Initiated Failover (OIF) requires that the system operator detect the failure of the primary system in the configuration and manually transfer control of shared physical disks to the backup system. However, the OIF software automates the transfer of physical disks from one host to the other, ensures that the transfer is complete, and restarts applications on the new host. Once the systems in a dual-initiator configuration are set up for OIF, the operator can transfer control of any shared disk with a single **sysadm** operation.

While recovery from system failure is the most obvious high-availability feature of OIF, OIF is also useful in other situations. For example, if one of the systems in a dual-initiator configuration needs maintenance, the operator can use OIF to transfer applications to the other system and proceed with the maintenance. Similarly, OIF can be used in situations where one of the systems in the configuration has load balancing problems.

Machine Initiated Failover — Once a dual-initiator configuration is set up for OIF, the operator can then set up the hosts for Machine Initiated Failover (MIF). This eliminates the need for the operator to detect a system failure and initiate the failover process. MIF automatically transfers physical disks and restarts applications when a system fails.

MIF has a monitoring process that detects any failure of the primary system and initiates OIF functionality. The MIF process

can use multiple communications paths, such as multiple LANs, to ensure that the monitored host has actually failed before it takes action. After transferring disks and applications to the backup host, MIF continues to check the status of the primary host. Once the primary host is rebooted and comes back on line, the MIF process automatically transfers the physical disks back to the primary host.

See Chapters 4 and 6 and *Configuring and Managing a CLARiiON® Disk-Array Storage System — DG/UX™ Environment* for more information on this feature.

IP takeover and NFS failover

In a networked environment, some file systems on AViiON servers may be mounted remotely on a number of NFS client systems. With IP takeover and NFS failover, two systems in a dual-initiator configuration can ensure that both remote login and NFS services are still available if one of the systems fail.

IP takeover is similar to disk failover, since a resource currently available from one system, in this case an Internet protocol address, is transferred to a second system. Two AViiON servers are set up in a dual-initiator configuration. In addition to their assigned IP addresses, the two systems manage a "floating" IP address. This IP address is not assigned exclusively to one of the servers, but can be transferred between them. If one of the systems fails, the backup system takes over the floating IP address.

The backup system can provide both login and NFS services for the failed server by quickly bringing the floating IP address back on line. With the failover of NFS services, NFS clients of the primary system can access mounted file systems through the backup system. However, those file systems must be on a disk that also fails over to the backup system.

After the backup system takes over the floating IP address, **telnet** and **rlogin** users need to re-establish their login connections. However, NFS services are unavailable only while the failover is taking place. Once NFS services have failed over, processing can continue with no further interruption. Applications using NFS services do not need to be restarted.

Figure 3–7 shows the IP takeover **sysadm** menu.

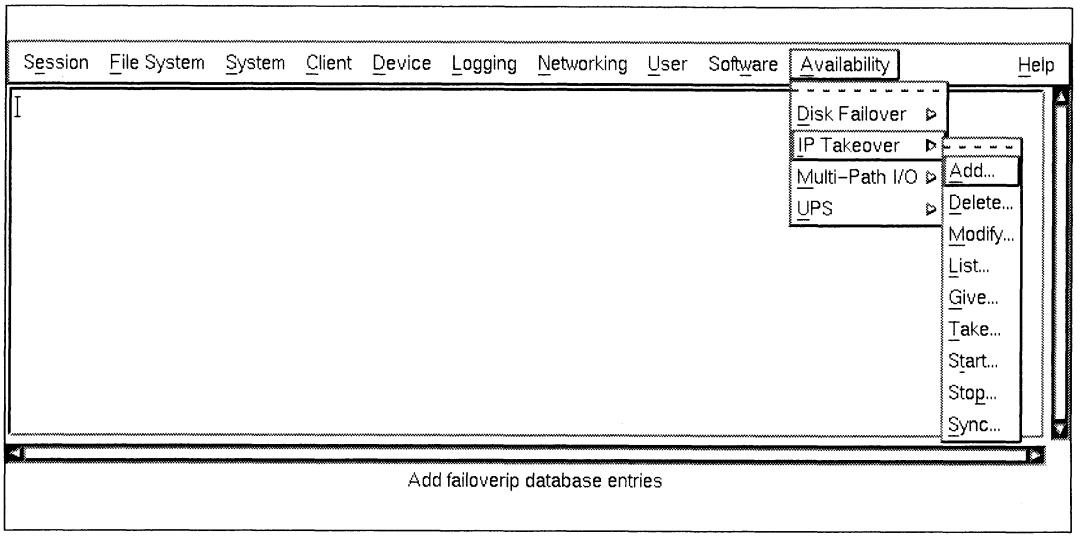


Figure 3–7 IP takeover menu

See Chapters 4 and 6 for more information on these features.

AV/Alert diagnostic support system

Every AViiON server with model numbers 4300 and above contains AV/Alert hardware and software components. The AV/Alert service is an automated support system that enhances high availability through automatic problem detection and remote assistance. When a hardware or software problem occurs, the AV/Alert system automatically notifies a Data General Customer Support Center. By providing early detection of existing faults and impending problems, the AV/Alert system catches minor problems before they cause system downtime.

The AV/Alert system continuously monitors DG/UX error logs and device drivers for error conditions. If it detects an error, the system sends an automatic machine-initiated (MI) message to a designated Support Center for handling.

The AV/Alert system integrates its automated error detection and reporting features with remote hardware testing and problem resolution. The result is a single, comprehensive diagnostic support system with the following problem resolution features:

- Machine-initiated (MI) calling
- Automated call handling
- Remote assistance services

Figure 3–8 illustrates how the AV/Alert Support System works.

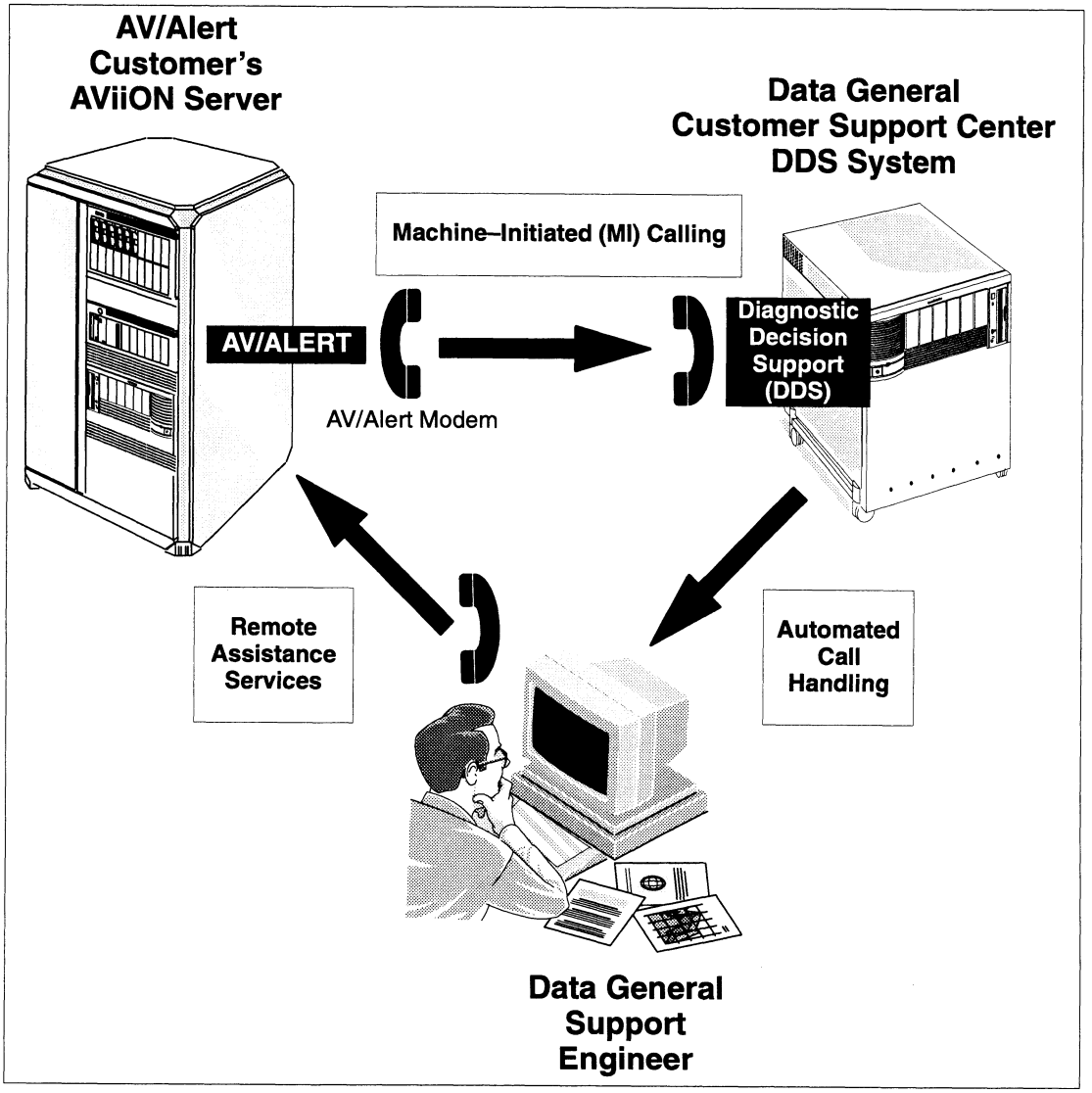


Figure 3–8 The AV/Alert support system

Machine-initiated calling

Machine-initiated (MI) calling has the following components:

- Automated problem detection
- Automated problem reporting

If the AV/Alert system detects a hardware or software problem resulting in a system error condition, the support system automatically transmits information about the incident to a Data General Customer Support Center computer through an attached modem. This automated transmission is called a machine-initiated incident report, or MI call. The AV/Alert system also reports errors reported to an AViiON server from connected peripherals, such as CLARiiON storage systems.

Automated call handling

Automated call handling has the following components:

- Problem solution identification
- MI incident tracking
- Rapid service dispatch
- Problem escalation

Diagnostic Decision Support (DDS) software installed on the AV/Alert support computer automatically determines the most valid response to a problem reported in an MI call. DDS is a sophisticated expert system that determines the proper response to a call. The support computer then distributes information as necessary about the MI to initiate replacement part orders, remote diagnostic support, service dispatch, or support engineer response.

Remote assistance services

Remote assistance services have the following components:

- Remote problem resolution
- Electronic MI incident closure
- Remote diagnostic testing
- Software updates

The modem connection between an AViiON computer with AV/Alert service and the Data General support staff provides a constant platform for remote diagnostic applications. A support engineer can log in to the AViiON server through the modem to investigate a problem reported in an MI report. The problems reported in MI incidents can be resolved remotely in about one third of the incidents. Besides handling MI incidents, remote assistance is available for preventive maintenance and similar tasks.

For more information

For more information about the AV/Alert system, see *Using AViiON® Diagnostics and the AV/Alert℠ Diagnostic Support System*.

Other packages

Other optional software packages can help an AViiON computer system achieve high availability. The following packages are covered in this section:

- TUXEDO® System/T
- Legato NetWorker
- OS/EYE*NODE
- Communications packages
- Relational Database Management Systems

TUXEDO System/T transaction processing monitor

The TUXEDO® System/T Enterprise Transaction Processing Monitor provides a platform for building UNIX-based On-Line Transaction Processing (OLTP) systems. The TUXEDO system ensures high availability by detecting and managing application failures and incorporating a two-phase transaction commit protocol. This protocol enables TUXEDO System/T to coordinate distributed transactions across a network.

TUXEDO System/T implements the notion of global transactions and provides data integrity by ensuring that transaction commit and rollback are properly handled at each network node. Also, TUXEDO coordinates the recovery of a global transaction when a node or network fails. The TUXEDO system enables applications to put service requests into a queue for delivery at a later time. This feature is useful in failover operations.

TUXEDO System/T provides the following high-availability benefits:

- Location-transparent user access to servers
- Redirection of user requests to a backup server when the primary server fails
- Application load balancing across servers
- On-line installation of TUXEDO software upgrades that cut down on planned system downtime.

  The addition of TUXEDO System/T provides quick, seamless system failover in AViiON systems set up for Machine Initiated Failover. For example, the TUXEDO system eliminates the need for users to log on to the backup system by automatically redirecting their requests to the backup server.

For more information

For more information on the TUXEDO System/T Enterprise Transaction Processing Monitor, see the set of manuals for the product. The *Tuxedo System Release 4.2 Product Overview* provides an overview of the product and a list of the additional manuals in the set.

Legato NetWorker backup software

The Legato NetWorker™ package protects against data loss through network-based back up and recovery. This package protects files against loss across an entire network of systems. This can provide AViiON computers with high performance back up and recovery services for a wide variety of machines.

The NetWorker package provides the following features:

- Network-wide backup and recovery
- Fast, easy daily backup operations
- Graphical user interface (GUI)
- Database of backups for easy file recovery

Convenient backups and easy file recovery are crucial for high-availability systems.

For more information

For more information on Legato NetWorker, see *Legato NetWorker Administrator's Guide* and *Legato NetWorker User's Guide*.

OS/EYE*NODE network management system

The OS/EYE*NODE™ system manages multi-vendor, multi-protocol networks. Since networks are increasingly crucial for an organization's internal and external information exchange, network downtime is often just as serious as system downtime.

The OS/EYE*NODE package provides the following features to ensure network uptime:

- Centralized network management of OSI and TCP/IP networks, including a gateway to NetView
- OSF/Motif compliant GUI
- Support for Ethernet, Token Ring, FDDI, and X.25 networks

As a component of a high-availability system, the OS/EYE*NODE system provides the following benefits:

- Monitors and controls network devices, systems, and applications
- Allows the setting of threshold alerts to warn of potential network problems before they occur
- Reduces the time needed to locate, diagnose, and fix a network problem

For more information

For more information, see *Managing a Network with OS/EYE*NODE™ for AViiON® Systems.*

Communications packages

The following communications packages contain high-availability features:

- NetWare® for AViiON Systems
- SNA for AViiON Systems

NetWare for AViiON Systems

NetWare 3.11 for AViiON Systems Revision 3.00 supports the following high-availability features:

- On AViiON systems with redundant LAN controllers, NetWare supports multi-path LAN I/O.
- On properly configured AViiON systems, NetWare supports disk failover.

  During NetWare's installation, it will configure MIF for defined NetWare volumes.

For more information, see *NetWare® for AViiON Systems: Installation Manual.*

SNA for AViiON Systems

SNA for AViiON Systems supports the following high-availability features:

- On AViiON systems with a single network interface, SNA can reestablish lost physical network connections and restart user sessions and applications.
- On AViiON systems with redundant network interfaces, SNA supports supports failover from a lost physical network connection to a standby physical network connection..
- On AViiON systems with redundant LAN controllers, SNA supports multi-path LAN I/O.
- On properly configured AViiON systems, SNA supports disk failover.

For more information, see *Installing, Configuring and Operating SNA for AViiON Systems.*

Relational database management systems

The addition of a relational database management system (DBMS) is a key component to AViiON high-availability systems. A Relational DBMS provides data integrity, consistency, and recovery through features such as the following:

- Transaction-based concurrency control
- Two-phase commits
- Rollback recovery
- Log files

In high-availability systems, these features are essential in ensuring application-transparent protection against data loss and application-level data integrity.

AViiON systems support all of the major relational DBMS, including ORACLE®, SYBASE®, INFORMIX™, INGRES™, and PROGRESS®.

For more information

For more information on a specific relational DBMS, see the documentation provided with that system.

End of Chapter

4 Managing DG/UX high-availability features

This chapter addresses three primary hardware and software configurations that enhance high availability. They are:

- Failover disks
- Internet Protocol (IP) takeover
- Multi-path I/O

Managing failover disks

Disk failover involves one or more disk drives accessible by two different routes (paths). Generally, the routes run to controllers in two hosts but sometimes they run to different controllers within the same host.

Disk failover is most useful where two host computers are connected to a common disk-array storage system; it provides higher availability of the stored data. If one host or a disk controller fails, the second system can take control of the disk by performing a *trespass* and make the data available again.

A failover disk can be on a shared SCSI bus in a *dual-initiator* configuration. Or with a disk-array storage system, a failover disk can be on a split bus in a cabinet-sharing configuration. Figures 4–1 and 4–2 show devices in dual-initiator configurations with a disk-array storage system. Figure 4–3 shows two hosts with a disk-array storage system in a split-bus configuration.

IMPORTANT: Each virtual disk mounted on a host must have a unique name. Therefore, if you want to set up a system disk for failover to another host, the virtual disks on that system disk must have different names from the virtual disks on the other host's system disk (for example, **root1**, **usr1**, and **opt1**).

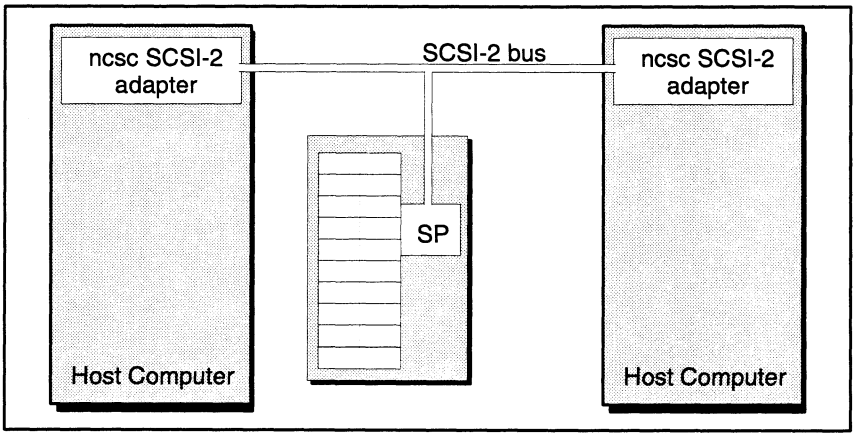


Figure 4–1 Two hosts with disk-array storage system in single bus dual-initiator configuration

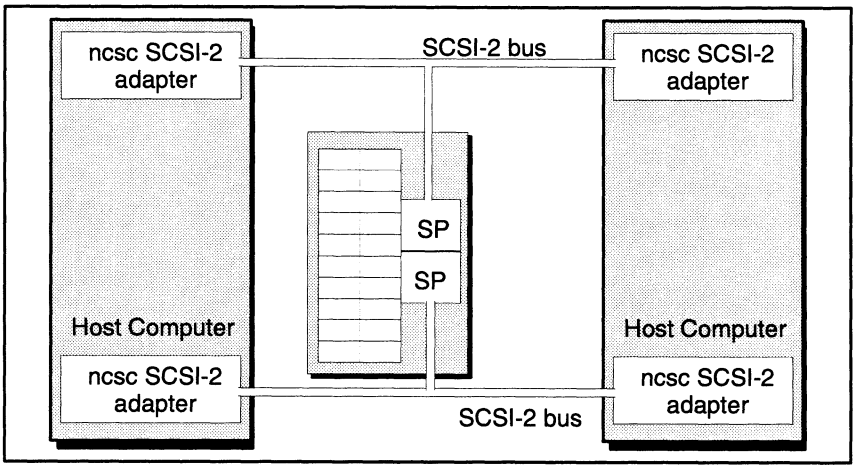


Figure 4–2 Two hosts with disk-array storage system in dual bus dual-initiator configuration

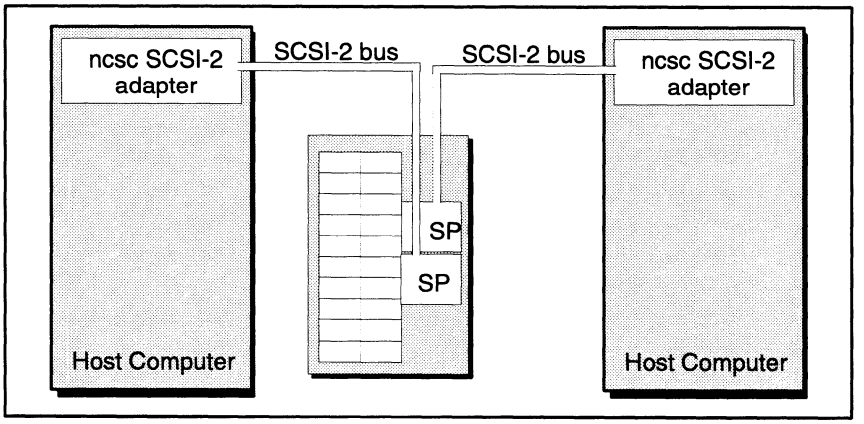


Figure 4–3 Two hosts with a disk-array storage system in a cabinet sharing configuration

The way you set up a device for failover depends on the configuration you want to support. Skip to the pertinent section following:

- Setting up Disks on a Shared SCSI Bus with a Disk-Array Storage System
- Setting up Disks on a Split SCSI Bus with Disk-Array Storage System

Shared SCSI bus with a disk-array storage system

This section tells how to set up two computers to share a SCSI bus in a dual-initiator configuration. It applies only if you have a disk-array storage system with an adapter that supports SCSI-2 protocol (such as a **ncsc**). For both hosts to use a shared bus, each must specify a different SCSI ID for its SCSI-2 adapter channel. For example, one host uses disk unit 0 via the disk unit name **sd(ncsc(1,7),0,0)** and the other host uses disk unit 1 via the disk unit name **sd(ncsc(1,6),0,1)**. The 7 and 6 in the unit names are the SCSI-2 adapter SCSI IDs; they allow each host to initiate its own requests to the bus.

The basic requirement for a shared SCSI bus configuration is that every device on the bus must have a different SCSI ID. This means the SCSI-2 adapters themselves (one or two on each system), must have different SCSI IDs as well. Essentially, with a disk-array storage system, the disk units themselves have LUN IDs, not SCSI IDs. The SPs (system control processors) have SCSI IDs. Therefore in the disk unit names shown above, if you omit the LUN numbers at the end, the SP device names become the unique names **sd(ncsc(1,7),0)** and **sd(ncsc(1,6),0)**; this satisfies the requirement for a unique SCSI ID.

CAUTION: If two or more devices on the SCSI bus have the same SCSI ID, either or both systems may hang or behave erratically. Be sure that each device on the bus, including each SCSI adapter, has a unique SCSI ID.

To set up two systems to share a SCSI bus, follow these steps:

1. If there are two SPs, make sure each is set to a unique SCSI ID (usually 0 and 1). The default SCSI ID for the first SP (SP A) is 0; for the second SP (SP B), it is 1. Do not set any devices to SCSI ID 6 or 7; reserve these numbers for the the SCSI-2 adapters.

   For example, you are setting up systems **host1** and **host2** to share a SCSI bus. The SCSI bus will connect to an SP that runs three disk drives on one SP, SP A, SCSI ID 0. The device name of the SP from each host will need to be unique. The disks, the SCSI-2 adapters, SPs, and the disk drive names on the two hosts are as follows in Table 4–1.

Table 4–1 Sample dual-initiator configuration with disk-array storage system

Device	SCSI ID adapter	SP	Physical unit number (LUN)	Disk drive name relative to host1	Disk drive name relative to host2
SCSI-2 adapter on **host1**, channel 1	6	0	Does not apply	**ncsc(1,6)**	Not accessible
SCSI-2 adapter on **host2**, channel 1	7	0	Does not apply	Not accessible	**ncsc(1,7)**
1.0 Gbyte disk (**host1 system disk**	6	0	0	**sd(ncsc(1,6),0,0)** (name for **host1** system file)*	Not accessible
1.0 Gbyte disk (**host2 system disk**	7	0	1	Not accessible	**sd(ncsc(1,7),0,1)** (name for **host2** system file)*
4.0 Gbyte disk (failover disk)	7	0	2	**sd(ncsc(1,6),0,2)** (name for **host1** system file)*	**sd(ncsc(1,7),0,2)** (path for failover, not in system file)

* For a host that has a second VME channel, the device name must also specify the VME channel. The VME channel name has the form **vme(*n*),** and immediately follows the SCSI-2 adapter name. For example, if **host1**'s system disk is on the second VME channel, **vme(1)**, the **host1** system disk device name is **sd(dgsc(vme(1),1,6),0,0)**. The **probedev** autosizer program automatically generates **vme** entries in the system file for you.

2. Install the hardware: connect the devices to either of the two systems in the standard fashion, such that you can boot and set up both systems independently.

3. Select one of the systems to set up first. Power up and boot the system. If necessary, install the DG/UX system software. This is the system whose SCSI-2 adapter will have the SCSI ID of 6. You set the adapter's SCSI ID at the SCM as discussed below.

4. Execute the **sysadm** operation System –> Kernel –> Build to build a new kernel. Proceed as usual. When **vi** runs, and you look for disk controllers in the system file, you will see lines that look something like this:

```
sd(ncsc(1),0)    ## SCSI disk 0 on Data General SCSI adapter 1
sd(ncsc(2),1)    ## SCSI disk 1 on Data General SCSI adapter 2
```

5. Make sure the entries in the system file include the correct SCSI-2 adapter and SCSI ID, correct SP SCSI ID, and correct disk drive unit number for each disk. For this host, this adapter, and SP, you want the system file to include the the **host1** system disk and

failover disk:

sd(ncsc(1,6),0,0)
sd(ncsc(1,6),0,2)

So for **host1** you would make sure the sd lines for that disk-array storage system look like this, adding the comments for clarity:

```
sd(ncsc(1,6),0,0)   ## 1st SCSI-2 adapter, 2nd channel (1); SP A (ID 0); unit 0
                    ## Host1 system disk.
sd(ncsc(1,6),0,2)   ## 1st SCSI-2 adapter, 2nd channel (1); SP B (ID 0); unit 2
                    ## failover disk.
```

IMPORTANT: For a shared or split bus configuration, never use the asterisk notation for SCSI devices [**sd(ncsc(1),***)]. Specify the entire, specific device names as explained above. If two hosts sharing a SCSI bus use the asterisk to configure all devices on the bus, the first host to start up will assume control of all devices, and the other will not be able to use any.

6. After making the changes you want to the kernel configuration, exit from the text editor (for **vi**, enter Esc, then ZZ).
7. Proceed with the Build operation and install the new kernel. If an error occurs, **sysadm** will display an error message. For more information on building a kernel, see *Managing the DG/UX™ System*.
8. Shut down the first system. While it is in the SCM, set the system's dual-initiator SCSI ID to **ncsc(1,6)**. Be sure that you set the SCSI ID to the same one you used in the system file.

IMPORTANT: If your system does not enable you to set the dual-initiator SCSI ID from the SCM, you must explicitly set it in the system file.

Also from the SCM, change the default boot path so that it includes the correct SCSI adapter SCSI ID. For example, if the default boot path had been **sd(ncsc(),0,0)**, change it to **sd(ncsc(1,6),0,0)**. If you also intend to set your system's boot path using **dg_sysctl**(1M) after you have brought the system back up, remember to specify the correct SCSI adapter SCSI ID there as well.

CAUTION: Be careful to specify the correct SCSI adapter SCSI ID in the boot path. If the boot path you use to boot your system specifies the SCSI ID of the SCSI adapter installed in the other system, the boot will fail and the SCSI bus will hang. If the SCSI bus hangs, an attempt by either system to access the SCSI bus will hang the system. When this happens, recover by resetting the hardware and rebooting.

9. Power off the system.

10. Power on and boot the second system. If necessary, install the DG/UX system software. This system's SCSI adapter will have SCSI ID 7.

11. Execute System –> Kernel –> Build and perform essentially the same steps that you performed for the first system: add device entries for any tape drives connected to the other system that you want to share; then in each device entry set the SCSI ID for the SCSI adapter. On this system, the SCSI adapter SCSI ID is 7.

    For example, in **vi**, the sample entries look like this:

```
sd(ncsc(1),0)    ## SCSI disk 0 on Data General SCSI adapter 1
sd(ncsc(2),1)    ## SCSI disk 1 on Data General SCSI adapter 2
```

    And for **host2**, you will change these to

```
sd(ncsc(1,7),0,1)    ## 1st SCSI-2 adapter, 2nd channel (1); SP A (ID 0); unit 0
                     ## Host2 system disk.
```

    And build and install the kernel as on the other host.

12. Shut down the second system. While it is in the SCM, set the system's dual-initiator SCSI ID to **ncsc(1,7)**. Be sure that you set the SCSI ID to the same one you used in the system file.

IMPORTANT: If your system does not enable you to set the dual-initiator SCSI ID from the SCM, you must explicitly set it in the system file.

Also from the SCM, change the SCM's default boot path so that it includes the correct SCSI adapter SCSI ID. For example, if the default boot path is **sd(dgsc(),1,0)**, change it to **sd(dgsc(1,7),1,1)**. If you also intend to set your system's boot path using **dg_sysctl**(1M) after you have brought the system back up, remember to specify the correct SCSI adapter SCSI ID there as well.

CAUTION: Be careful to specify the correct SCSI adapter SCSI ID in the boot path. If the boot path you use to boot your system specifies the SCSI ID of the SCSI adapter installed in the other system, the boot will fail and the SCSI bus will hang. If the SCSI bus hangs, an attempt by either system to access the SCSI bus will hang the system. When this happens, recover by resetting the hardware and rebooting.

13. After resetting the SCM boot path, power off the system.

14. Power on and boot the first system with its new kernel.

15. When the first system has completed booting, power on and boot the second system with its new kernel.

You have set up the configuration shown in Table 4–1.

At this point, only **host1** can access the failover disk on SCSI ID 2. To enable **host2** to take over this disk, you must use **sysadm** to configure the disk for operator-initiated failover (OIF) and then optionally for machine-initiated failover (MIF). To enable OIF, you will do this by adding the disk to the **sysadm** giveaway database on both hosts, as described later in this chapter, section "Using Failover Disks." You can then use **sysadm** to transfer the disk from **host1** to **host2** and, if you want, enable machine-initiated failover.

Split SCSI bus with a disk-array storage system

This section tells how to set up two computers to share a SCSI bus in a cabinet sharing configuration. You can use this only if you have a disk-array storage system with an adapter that supports SCSI-2 protocol (such as a **ncsc**).

A cabinet-sharing configuration actually has two busses, each connected to its own SP. A split bus has one important advantage over a shared (dual-initiator) bus. Since the split bus uses a path of entirely different components, it offers higher availability than a shared bus. For example, if the SP in a shared-bus path fails, all access to the SP's disks is lost; if an SP in a split-bus path fails, the system can route I/O though the other SP.

The single-host and dual-host cabinet sharing configurations are shown in Figure 4–4.

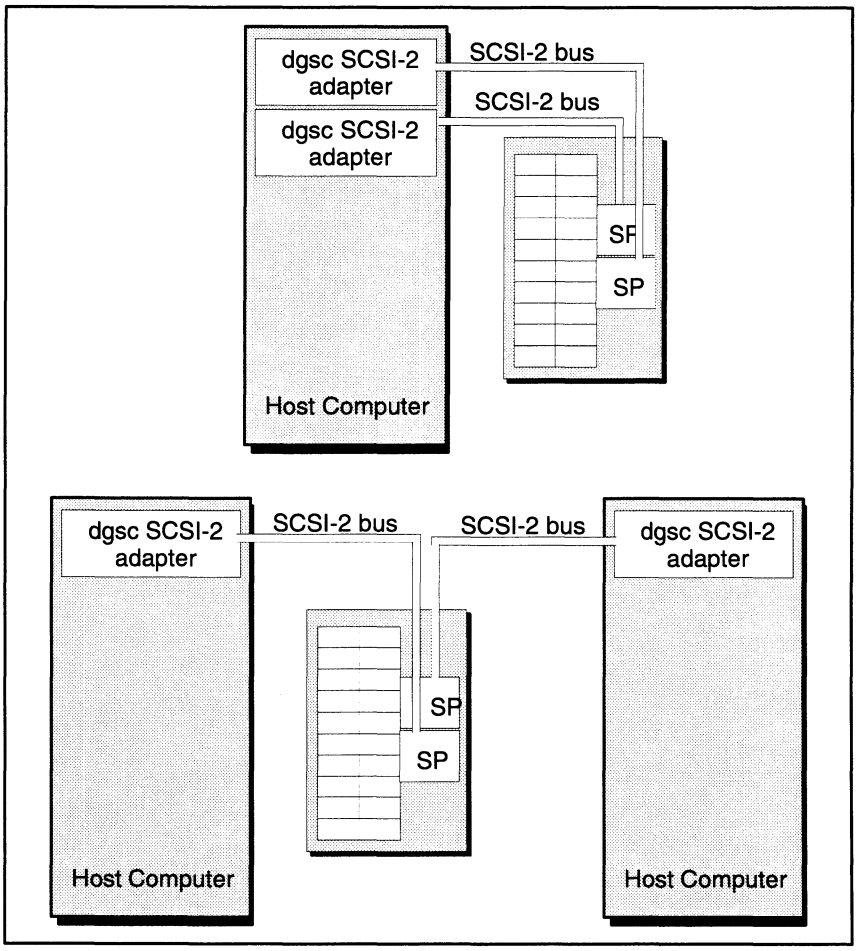


Figure 4–4 Cabinet sharing configuration in one host and two hosts

When you generate a kernel (in one host or two) to support a cabinet sharing configuration, you must fully qualify the disk unit names in the system file so that each host (or path) registers only the disks you want it to. For example, one host uses disk unit 0 via the disk unit name **sd(ncsc(1),0,0)** and the other host uses disk unit 1 via the disk unit name **sd(ncsc(1),0,1)**. (You do not need to specify the SCSI-2 adapter SCSI IDs as with a shared bus.)

For more information on setting up and generating kernels to support a split-bus configuration, see the the 014-series manual that accompanies the disk-array storage system hardware.

Using failover disks

The section explains what failover does and the tasks involved in managing failover disks. You can implement disk failover with any disk on a shared SCSI-2 bus or with any disk on a split bus. Disk failover is most useful with disks in a disk-array storage system.

The DG/UX failover feature allows you to connect a physical disk to two systems and transfer control of the disk from one system to the

other as needs arise. The failover feature not only handles operations involved in shutting off access to the disk on one system, but it also handles operations involved in making the disk accessible on the other system and starting any applications used to access it.

The systems that will be using the failover disks must be connected by a TCP/IP network or SLIP connection. The failover daemons on the systems communicate over a port running a SAF **listen** port service added during setup of the DG/UX system software.

Failover types

There are two types of disk failover: operator-initiated failover (OIF), which requires human intervention, and machine-initiated failover (MIF) which occurs without human intervention. To set up OIF, you need two hosts and a disk-array storage system (with one or two SPs) or one host and two SPs.

To set up MIF, you need OIF set up, and you need two hosts, each ideally with two TCP/IP network interfaces connected to the other host. With MIF, a failover monitor daemon process runs in a host and periodically checks the other host to make sure it is still running. If the other host is not running, the monitoring host takes over the disks specified in the giveaway database (set up via **sysadm**). Then the monitoring host runs a script whose name you specified; this script can restart applications and issue whatever other shell commands are needed to restore the production environment. When the surviving host detects that the other host is running again, it runs a regain-pulse script whose name you specified; this script can unmount and deregister the disks it took over so that the original owning host can mount and use them.

MIF will work with only one TCP/IP network interface (or SLIP connection), but if the network fails, the monitoring host will assume that the other has failed and will take over the other host's disks.

What failover does

Only one system may configure and register a physical disk at a time; therefore, a failover disk can be accessible only on one system at a time. When you use an OIF procedure to transfer control of a failover disk from one system to another, the system performs several tasks.

If the system that normally uses the failover disk fails, a human operator or the MIF software performs the failover from the other host by *trespassing*. Trespassing forces disk ownership to change. The system taking over the disk

1. Issues the command needed to transfer control of the disk.
2. Registers the disk.
3. Runs **fsck**(1M) to check file systems.
4. Mounts and exports file systems.
5. Optionally, starts applications used to access the disk.
6. Logs and reports operation status.

If the system giving the failover disk is running (as will often not be true), it

1. Terminates processes accessing the disk.
2. Unmounts file systems on the disk.
3. Deregisters the disk.

The effect of these tasks varies depending on how you have configured the systems. For example, if you have not built file systems on the disk, the steps related to file systems do not occur.

Depending on how you have configured the disks on your system, failover may involve multiple physical disks. This is true if the virtual disks or file systems designated for failover span multiple physical disks, as in the following cases:

- A file system was built on a virtual disk whose components span multiple physical disks.
- A striped virtual disk, which necessarily resides on multiple physical disks.
- A software-mirrored virtual disk whose images reside on multiple physical disks.

In any of the cases above, the system requires that all such interdependent physical disks be failed over together. The system does not allow you to perform failover such that only part of a virtual disk or mirror is being moved to the other system; either all parts of the virtual disk or mirror failover, or the operation fails.

Depending on how you have set up your system, there may also be other cases where a failover involves multiple physical disks. For example, you may have a database application in a file system on one physical disk and the database itself located on another physical disk. In this case, you set up both disks for failover and move them together.

The **sysadm** operations that support the failover feature are beneath the Availability –> Disk Failover menu. The **sysadm** operations call the **admfailoverapplication**(1M), **admfailoverdisk**(1M), **admfailovergiveaway**(1M), **admfailoverhosts**(1M), and **admfailovertakeaway**(1M) commands. The manual pages provide detailed information.

OIF failover databases maintained by the DG/UX system

DG/UX uses the following databases. You create and maintain them using the **sysadm** menu sequences Availability –> Disk Failover –> Giveaway or equivalent **adm** failover commands.

- **/etc/failover/application**

  The **application** database contains information about application start-up scripts to be executed when a physical disk is failed over. The full pathname of the script must be included. Any number of scripts may be included in **application**.

- **/etc/failover/hosts**

  The **hosts** database contains information about hosts in dual-initiator configurations with this host. Entries contain the name of the host, the communication path to the host ("network" is the only path currently supported), and the current synchronization status of the host. The host must be INSYNC before it can participate in a **give** or **take** operation.

- **/etc/failover/giveaway**

  The **giveaway** database contains information about the physical disks that can be given to other hosts in the dual-initiator configuration. Entries contain any and all file system information that was gathered at the time the disk was added. This includes the name of the host that is giving away the disks, information about the virtual disks on the local and remote systems, and information about the file system to unexport, unmount, and delete during a Give operation.

  Some operations that you use to maintain the **giveaway** database, Add and Delete, offer the option of synchronizing databases. If you do not synchronize databases, the system flags them as out of sync, or NOTSYNC, and will not allow you to transfer control of any failover disk. As an alternative to the Synchronize Database option in the Add, Modify, and Delete operations, you can synchronize databases with the Synchronize operation. A later section discusses the Synchronize operation.

- **/etc/failover/takeaway**

  The **takeaway** database contains information about the physical disks that can be taken from other hosts in the dual-initiator configuration. The entries are similar to those in the **giveaway** database.

For more information about these databases, see the **failover**(4M) manual page.

Setting up and using OIF

To set up Operator Initiated Failover (OIF) for any disk, you must execute the following steps.

1. On the host that will be the primary owner of the disk, use the **sysadm** sequence Device –> Disk –> Physical –> List to make sure the disk is configured and registered.
2. Still on the host that will be the primary owner of the disk, with the network running, add the disk as a failover disk with the **sysadm** sequence Availability –> Disk Failover –> Giveaway –> Add operation. When you have added all the disks you want to use for failover, be sure to answer **yes** to the synchronize prompt. The synchronize step enters the disk name in the takeaway database of the other host, allowing the disk to be failed over. A later section discusses the Add operation in more detail.
3. With the failover disk(s) added and databases synchronized, you can transfer control of the disk(s) should the need arise. There are several ways to perform failover, as follows.
   - When the giving system has failed, you can use the taking system to transfer control to the taking system with the **sysadm** sequence Availability –> Disk Failover –> Take and the **Use Trespass** option.
   - When the giving system is functioning normally, you can transfer control using either system:

     from the giving system using the **sysadm** sequence Availability –> Disk Failover –> Give; or

     from the taking system using the sequence Availability –> Disk Failover –> Take *without* the **Use Trespass** option.

After setting up OIF as explained in this section, you can then think about setting up MIF, described in a later section.

Managing the giveaway database

The **sysadm** sequence Availability –> Disk Failover –> Giveaway enables you to add, delete, modify, and list entries in the **giveaway** failover database. The following sections describe each of these operations.

Adding a giveaway entry — Before you can transfer control of a disk from your system to another, you must add the disk to the **giveaway** database using the Availability –> Disk Failover –> Giveaway –> Add operation. This operation examines the contents of the specified physical disk looking for virtual disks, disk mirrors, and export information. This information is then formatted and added to the **giveaway** database. When you synchronize, the information is added to the other host's **takeaway** database.

The Add operation queries you for the following information:

`Local Physical Disk`
: Enter the name of a local physical disk, for example, **sd(dgsc(0,6),1,0)**, that is currently configured and registered.

`Remote Physical Disk`
: Enter the name that the physical disk will have on the remote system; for example, **sd(dgsc(0,7),1,0)**.

`Hostname to Fail Disk over to`
: Enter the host name of the remote system. The system must be currently accessible on the network. If necessary, **sysadm** adds this name to the **/etc/failover/hosts** database.

`Application Command Line`
: Enter the command name to be executed on the remote system after transferring the failover disk. Typically, this command line starts the application used on the remote host to access the data on the failover disk. If necessary, **sysadm** adds this command line to the **application** database.

`Synchronize database`
: Select this option to copy the information for this failover disk to the **takeaway** database on the remote system. If the remote system is inaccessible, synchronization will fail. When the remote system becomes accessible again, synchronize databases with the Synchronize operation, discussed later. Until you synchronize databases, the system considers the local giveaway database as out of sync, or NOTSYNC, and restricts you from giving *any* failover disks to other systems.

  The **giveaway** database includes information normally associated with a local file system, for example, export options, **fsck** pass number, and so on. The Add operation derives this information from the local file system table (**fstab**(4)), which you may review with the File System –> Local Filesys –> List operation. To change this information in the giveaway database, select Availability –> Disk Failover –> Giveaway –> Modify, discussed later.

The Add operation scans the failover disk for dependencies on other physical disks. A failover disk is dependent if it contains any virtual disk with a part residing on another physical disk. If any such dependencies exist, the Add operation displays a warning. You

cannot transfer control of a failover disk without also transferring any codependent disks with it.

Adding the disk(s) you want to use for failover is the essential software task required for OIF. After adding the disk(s) you want, you can then think about setting up MIF, described in a later section. To maintain the **giveaway** and other OIF databases, continue in this section.

Deleting a giveaway entry — To remove a failover disk from the local **giveaway** database, select the operation Availability –> Disk Failover –> Giveaway –> Delete. If this physical disk has dependencies on other physical disks, the operation will warn you. A dependent physical disk is one that contains part of a virtual disk that also has parts residing on other physical disks. You cannot failover a disk without also failing over any disks that have dependencies on it.

The Delete operation queries you for the following information:

`Host name` Select the name of the remote host designated for the failover disk.

`Disk name` Select the name of the local physical disk designated as the failover disk.

`Virtual Disk name`
Select the name of the virtual disk you want to delete.

`Synchronize database`
This option removes the information for this failover disk from the takeover database on the remote system. If the remote system is inaccessible, synchronization will fail. When the remote system becomes accessible again, synchronize databases with the Synchronize operation, discussed later. Until you synchronize databases, the system considers the local giveaway database as out of sync, or NOTSYNC, and restricts you from giving *any* failover disks to other systems.

Modifying a giveaway entry — To change the information about a failover disk in your **giveaway** database, select the operation Availability –> Disk Failover –> Giveaway –> Modify. You may, for example, wish to change some of the file system mounting information (such as mount point, export options, and so on) associated with a file system on the failover disk. The Modify operation queries you for the following information:

`Host name` Select the remote host designated for the failover disk.

`Disk name` Select the physical disk used for failover.

`Virtual Disk name`
: Select the pathname of the virtual disk's special file. For example, the special file for a virtual disk named **db1** would be **/dev/dsk/db1**.

`Synchronize database`
: This option updates the information about this failover disk in the takeover database on the remote system. If the remote system is inaccessible, synchronization will fail. When the remote system becomes accessible again, synchronize databases with the Synchronize operation, discussed later. Until you synchronize databases, the system considers the local giveaway database as out of sync, or NOTSYNC, and restricts you from giving *any* failover disks to other systems.

Sysadm then allows you to change any of the other information associated with the virtual disk entry, including the remote disk name and the file system mounting information. For complete discussion of file system mounting information, see the discussion of local file systems in *Managing Mass Storage Systems and DG/UX™ File Systems*.

Finally, **sysadm** offers the Synchronize database option, which you select to copy changes you made to the takeover database on the remote system. If the remote system is inaccessible, synchronization will fail. When the remote system becomes accessible again, synchronize databases with the Synchronize operation, discussed later. Until you synchronize databases, the system considers the local **giveaway** database as out of sync, or NOTSYNC, and restricts you from giving *any* failover disks to other systems.

Listing giveaway entries — To display information about local failover disks, select the Availability –> Disk Failover –> Giveaway –> List operation. The display includes information for failover disks on the local system, stored in the **giveaway** database.

The entries in the **giveaway** database include the following information for each virtual disk piece on each failover disk:

`Hostname` The remote host designated for failover.

`Local Diskname`
: The local name of the failover disk on which the piece resides.

`Remote Diskname`
: The remote name of the failover disk on which the piece resides.

`File System Source`
: The pathname of the special file for the virtual disk, for example, **/dev/dsk/db1**.

`<FS INFO>` The information used for mounting the file system, such as mount point directory, **fsck** pass number, and so on. For a discussion of this information, see the discussion of local file systems in *Managing Mass Storage Systems and DG/UX™ File Systems*.

Managing the takeaway database

The **sysadm** sequence Availability –> Disk Failover –> Takeaway enables you to delete, modify, and list entries in the **takeaway** failover database as follows. The other host does not need to synchronize after you change the takeaway database.

Deleting a takeaway entry — To remove a failover disk from the local **takeaway** database, select the operation Availability –> Disk Failover –> Takeaway –> Delete.

The Delete operation queries you for the following information:

`Host name` Select the name of the remote host you want to delete.

`Disk name` Select the name of the physical disk module or modules for which you want to delete **takeaway** entries.

`Virtual Disk name`
: Select the name of the virtual disk entry you want to delete.

Modifying a takeaway entry — To change the information about a failover disk in the **takeaway** database, select the operation Availability –> Disk Failover –> Takeaway –> Modify. The entries in this file are usually taken from the remote system's **giveaway** database and do not need to be modified. However, there are times when you might need to modify **takeaway** entries.

For example, if you want to failover a system disk, you would need to modify the **takeaway** database. On your system the **giveaway** entries would have file system information for / and **/usr** file systems. These get copied to the **takeaway** database on the remote host. However, a / and **/usr** already exist on this system. You could then modify the **takeaway** entry on the remote system to give the file systems different names, such as **/root_failover** and **/usr_failover**. Now the file systems would get mounted as normal file systems which enables you, for example, to correct a corrupted system file.

The Modify operation queries you for the following information:

`Host name` Select the host you want to modify.

`Disk name` Select the physical disk module or modules for which you want to modify **takeaway** entries.

`Virtual Disk name`
Enter or select the name of the virtual disk you want to modify.

The operation then allows you to change any of the other information associated with the virtual disk entry, including the remote disk name and file system mounting information. For detailed file system mounting information, see the discussion of local file systems in *Managing Mass Storage Systems and DG/UX™ File Systems*..

Listing takeaway entries — To display information about remote failover disks, select the Availability –> Disk Failover –> Takeaway –> List operation. The display includes information for failover disks on the remote system, stored in the **takeaway** database. The entries in the **takeaway** database include the following information for each virtual disk piece on each failover disk:

`Hostname` The remote host designated for failover.

`Local Diskname`
The local name of the failover disk on which the piece resides.

`Remote Diskname`
The remote name of the failover disk on which the piece resides.

`File System Source`
The pathname of the special file for the virtual disk, for example, **/dev/dsk/db1**.

`<FS INFO>` The information used for mounting the file system, such as mount point directory, **fsck** pass number, and so on. For a discussion of this information, see the discussion of local file systems in *Managing Mass Storage Systems and DG/UX™ File Systems*.

Managing the application database

The **sysadm** sequence Availability –> Disk Failover –> Application menu enables you to add, delete, modify, and list entries in the **application** failover database, as follows.

Adding an application entry — Use this to enter application scripts to be executed when a disk is failed over. You can use these scripts to minimize a system's down time by restarting applications and similar tasks. To add an entry to the **application** database, select the Availability –> Disk Failover –> Application –> Add operation.

The Add operation queries you for the following information:

`Host Name` Select the host containing the physical disk for which you want to add an application script.

`Disk Name` Select the physical disk module or modules for which you want to add an application script.

`Remote Physical Diskname`
Enter the name of physical disk on the remote system.

`Application Command Line`
Enter the full pathname of a script or command to be executed when the local physical disk is failed over.

Deleting an application entry — To remove an entry in the **application** database, select the Availability –> Disk Failover –> Application –> Delete operation. You can either delete specific entries or all **application** entries for a specified physical disk.

The Delete operation queries you for the following information:

`Host Name` Select the host containing the physical disk for which you want to delete an application script.

`Disk Name` Select the physical disk module or modules for which you want to delete an application script.

`Application`
Choose the application for which you want to delete entries.

Modifying an application entry — To modify an entry in the **application** database, select the Availability –> Disk Failover –> Application –> Modify operation.

The Modify operation queries you for the following information:

`Host Name` Select the host containing the physical disk for which you want to modify an application script.

`Disk Name` Select the physical disk module or modules for which you want to modify an application script.

`Application`
Select the application for which you want to modify entries.

`Remote Physical Diskname`
Enter the name of physical disk on the remote system.

`Application Command Line`
Enter the full pathname of a script or command to be executed when the local physical disk is failed over.

Listing application entries — To display information about the entries in the **application** database, select the Availability –> Disk Failover –> Application –> List operation.

The entries in the **applications** database include the following information:

`Hostname` The remote host designated for failover.

`Local Diskname`
The name of the physical disk as it appears on the current host.

`Remote Diskname`
The name of the physical disk as it appears on the host designated for failover.

`Application Command`
The full pathname of the script to be executed when the disk is failed over.

Managing the hosts database

The **sysadm** Availability –> Disk Failover –> Hosts menu enables you to add, delete, and list entries in the **hosts** failover database. The following sections describe each operation.

Adding a hosts entry — To add an entry to the **hosts** database, select the Availability –> Disk Failover –> Hosts –> Add operation. The name of the host to add is validated against the TCP/IP local and ONC/NIS **hosts** databases to ensure it is a valid host name.

The Add operation queries you for the following information:

`Host name` Enter the name of the host to which a physical disk can be failed over. This must be a system that shares a disk in a dual-initiator configuration with your system.

Deleting a hosts entry — To remove an entry in the **hosts** database, select the Availability –> Disk Failover –> Hosts –> Delete operation.

The Delete operation queries you for the following information:

`Host name` Select the host you want to delete.

Modifying a hosts entry — To modify an entry in the **hosts** database, select the Availability –> Disk Failover –> Hosts –> Modify operation. This operation currently does nothing.

Listing hosts entries — To display information about the entries in the **hosts** database, select the Availability –> Disk Failover –> Hosts –> List operation.

The entries in the **hosts** database include the following information:

`Hostname` The remote host designated for failover.

`Comms Path` The communication path to the host designated for failover.

`Remote Diskname`
The current synchronization state of the **failover** databases for the designated host.

Giving away a failover disk

To transfer control of a failover disk from your system to another, select the **sysadm** operation Availability –> Disk Failover –> Give. The operation requires that you enter the Host Name of the remote host and the Disk Name of the failover disks you wish to give. You should have already added the disks to the failover databases.

The **admfailoverdisk**(1M) command then consults the failover databases and performs the following actions:

- Terminates all user processes of the virtual disks and file systems on the specified physical disk.
- Unmounts the file systems and deregisters the physical disks.
- Registers the disks on the specified remote host.
- Checks and mount the file systems on the remote host.
- Executes any commands or scripts associated with the physical disk.

You cannot give *any* failover disk if your local giveaway database is out of sync, or marked NOTSYNC in the failover hosts file. Synchronize databases with the Synchronize operation, discussed later.

Taking away a failover disk

To transfer control of a failover disk from another system to your own, select the **sysadm** operation Availability –> Disk Failover –> Take. The operation requires that you enter the host name of the remote host and the disk name of the failover disks you wish to take. You can take a failover disk only if

- The administrator of the remote system has added the disk to the failover databases, and
- The failover databases on your system have been successfully synchronized with the databases on the remote system.

The process of taking a failover disk involves having the **admfailoverdisk**(1M) code consult the failover databases and perform actions similar to those described previously for giving away a failover disk.

Optionally, you may select the Use Trespass option. This option is intended for situations where the remote system is hung or crashed. If the remote system is functioning normally, do not select the Use Trespass option. If you select the Use Trespass option when taking a failover disk from a normally functioning system, access to the failover disk on the remote system will be terminated ungracefully, possibly resulting in unnecessary data loss.

You cannot take a failover disk if your local takeaway database is out of sync, or NOTSYNC. Synchronize databases with the Synchronize operation, discussed later.

Verifying the failover database

To make sure that the local giveaway database has up-to-date information about failover disks and their virtual disks, or to list physical disk dependencies, select the operation Availability –> Disk Failover –> Check.

The operation requires that you specify the Host name of the remote system, the Disk name of the failover disk, and File name of the database files whose entries you would like to verify. The two files are the **giveaway** database, which contains information about physical and virtual disks designated to be given to other systems, and the **takeaway** database, which contains entries for physical and virtual disks designated to be taken from the other systems.

The operation scans the physical disk for all virtual disk pieces and correlates them with their entries in the local file system table, **fstab**, which you can display with the File System –> Local Filesys –> List operation. The operation then updates entries as necessary in the selected failover databases.

The operation also scans the physical disks for dependencies on other physical disks. A physical disk is considered dependent when it contains parts of virtual disks or mirrors that have parts on other physical disks. The operation reports any such dependencies that it finds. You may then add these other physical disks, using the Availability –> Disk Failover –> Giveaway –> Add operation, as necessary.

Synchronizing failover databases

To make the failover databases on your system consistent with those on a remote system, select the operation Availability –> Disk

Failover –> Synchronize. For the specified remote system, the operation scans the local **giveaway** database and copies to the remote **takeaway** database entries for any virtual disks designated to be given to the remote system. The operation does *not* copy the remote takeaway disk entries to the local **giveaway** database.

Some operations that change the local **giveaway** database, such as Add, Delete, and Modify, offer the option of synchronizing databases. If you select this, **sysadm** copies the changes as appropriate to the takeaway database on the remote system. If you do not synchronize databases, the system flags the local **giveaway** database as out of sync, or NOTSYNC, and restricts you from giving any failover disks to another system until you have synchronized databases.

Machine Initiated Failover (MIF)

After your system is set up for OIF, you can set up Machine Initiated Failover (MIF) to eliminate the need for a human operator to detect a system failure and initiate the failover process. MIF has a monitoring process, **failovermon**(1M), that detects a failed host and initiates OIF disk takeover; the process can also detect a restored host and take steps to restore the disks taken over. The monitor process can use multiple communication paths, such as redundant LANs, to ensure that the monitored host has actually failed before taking action. When you define (add) a monitor process, you can specify that it is to run automatically on system reboot. If you do this, the process will run automatically at startup (by a command **sysadm** inserts in an init 3 script) until you stop the process or the system comes down.

MIF has two administrative commands, **admfailoveraltcommpath**(1M) and **failovermon**(1M), with associated **sysadm** interfaces. **Admfailoveraltcommpath** is used to maintain the **altcommpath** database and to check alternate communication path accessibility. **Failovermon** is used to maintain the **monitors** database and to perform failover monitoring. The manual pages provide detailed information.

MIF uses the following databases in addition to the OIF databases.

- **/etc/failover/altcommpath**. This contains information about alternate communication paths to the host that is being monitored. Entries contain the primary hostname of the host to be monitored and the hostname associated with the alternate communication path.
- **/etc/failover/monitors**. This contains information about the **failovermon** monitors that a system will run to monitor another system. Before another system can be monitored, there must be an entry for it in **monitors**.

Details on these two databases appear later in the chapter.

Setting up and using MIF

Before setting up MIF, you must set up and synchronize all of the OIF databases as explained earlier. Then follow these steps on each monitoring host to set up MIF. More details on each step appear in the following sections.

1. Execute Availability –> Disk Failover –> Alternate Paths –> Add on each monitoring host to set up the **altcommpath** database. Add the alternate communications paths that you want the monitor to use. **Sysadm** will request, and you must provide, the following values.

   - Primary host name: the network interface name of the host to monitor. DG/UX uses this name to look up alternate communication path entries.
   - Alternate remote host name: the secondary network interface name of the host to monitor. DG/UX uses this name to establish an alternate communication route to the host.

     After providing both names, tell **sysadm** to check the paths you specified; then confirm to perform the operation.

2. Execute Availability –> Disk Failover –> Monitors –> Add on the monitoring host to set up the monitors database and start the monitor.

   Sysadm will request, and you must provide, the following values.

   - Name of the host to monitor.
   - Seconds between heartbeats: the interval in seconds the monitor will sleep between message cycles.
   - Number of retries: the number of times the monitor will retry communication on each path before deciding the monitored host has failed.
   - Full pathname to the lost-pulse script to be executed when the monitoring host decides that the monitored host has failed. Data General supplies a default lost-pulse script in **/etc/failover/failovermon_lost_pulse**. You can edit this and use it or create another one.
   - Full pathname to the regain-pulse script to be executed when the monitoring host decides that the failed host has returned. Data General supplies a default regain-pulse script in **/etc/failover/failovermon_regain_pulse**. You can edit this and use it or create another one.
   - **Yes** or **no** value indicating whether you want to start the monitor automatically on each system reboot.

- **Yes** or **no** value indicating whether you want to start the monitor immediately after the successful completion of the Add operation.

The monitoring process

The failover monitor, **failovermon**(1M), is a daemon process that continues to run until you explicitly stop it or the system comes down. The monitor executes a loop of the following 6 tasks, shown in Figure 4–5. When you define (add) a monitor process, you can specify that it is to run automatically on system reboot. If you do this, the process will run automatically at startup (by a command **sysadm** inserts in an init 3 script) until you stop the process or the system comes down.

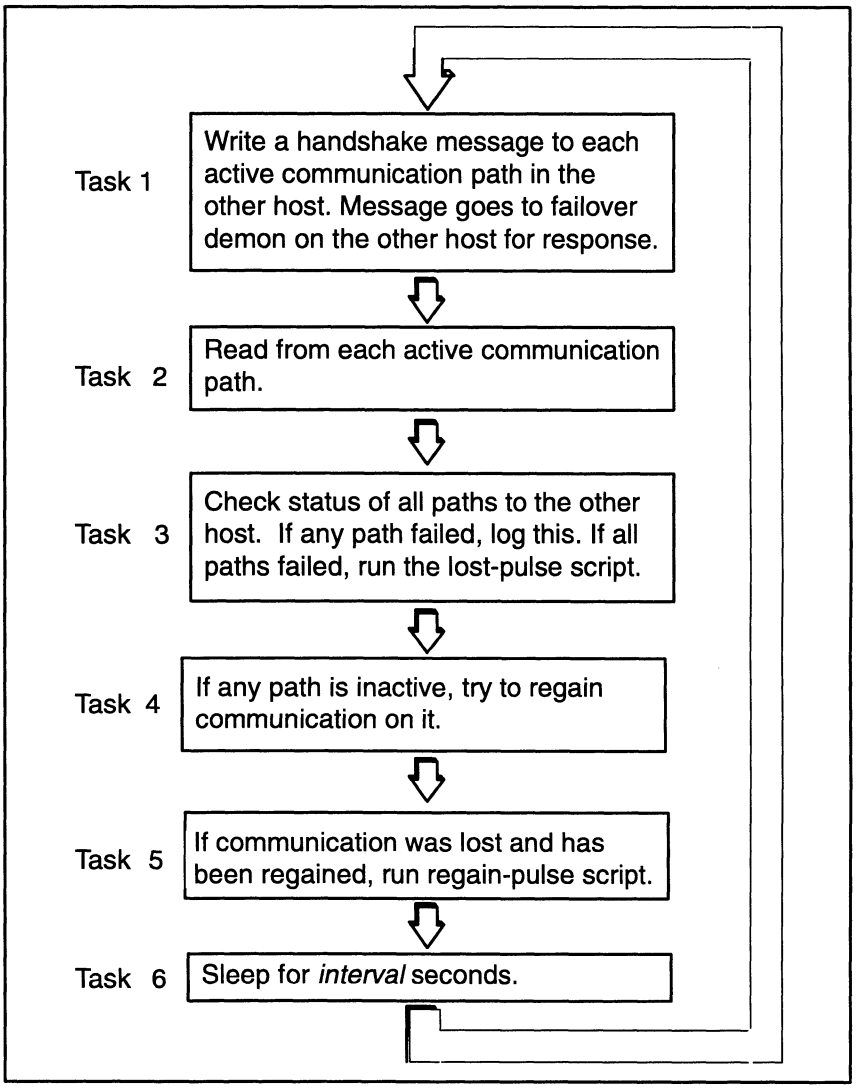


Figure 4–5 Tasks performed by the MIF failover monitor process

Managing the alternate paths database (altcommpath)

The **sysadm** sequence Availability –> Disk Failover –> Alternate Paths lets you add, check, delete, modify, and list alternate paths. Note that you do not need to perform this operation if your system does not have an alternate path available. The following sections describe these operations.

Adding an alternate path — To add an alternate path for failover, use the **sysadm** sequence Availability –> Disk Failover –> Alternate Paths –> Add. **Sysadm** prompts for the following information:

`Primary Host Name [moe]`
: Enter the primary host name of the remote host. If the remote host does not have an alternate hostname, the primary hostname is the only name.

`Alternate Remote Host Name [moe]`
: Enter the alternate host name of the remote host. For example, you could enter `moe-alt`.

`Check path [no]`
: Answer **yes** if you want to check the alternate communication path at the conclusion of the Add operation; otherwise answer **no**. Generally, we recommend a **yes** answer.

As with many other **sysadm** sequences, it asks for confirmation by prompting `Ok to perform operation`. Answer **yes** to add the alternate path as specified or **no** if you want to change a value.

If you told **sysadm** to check the path, it does so. The path check takes a few moments. If the alternate path is usable, **sysadm** will display

`Alternate path [`*alternate-remote-hostname*`] to [`*primary-hostname*`] is accessible.`

If the path is not usable, **sysadm** will display a message to that effect. Make sure the network(s) are operational; if they are, you should respecify the alternate path with Modify.

Checking an alternate path — To check an alternate path, use the **sysadm** sequence Availability –> Disk Failover –> Alternate Paths –> Check. **Sysadm** prompts for the primary and secondary host names as with Add above.

The path check takes a few moments. If the alternate path is usable, **sysadm** will display

`Alternate path [`*alternate-remote-hostname*`] to [`*primary-hostname*`] is accessible.`

If the path is not usable, **sysadm** will display a message to that effect. Make sure the network(s) are operational; if they are, you should respecify the alternate path with Modify.

Deleting an alternate path — To delete an alternate path, use the **sysadm** sequence Availability –> Disk Failover –> Alternate Paths –> Delete. **Sysadm** prompts for the primary and secondary host names as with Add above. After you confirm, it deletes the alternate path.

Modifying an alternate path — To change an alternate path, use the **sysadm** sequence Availability –> Disk Failover –> Alternate Paths –> Modify. **Sysadm** prompts for the primary and secondary host names and offers to check the path as with Add above. Enter the changed values you want. After you confirm, **sysadm** modifies the alternate path.

Listing an alternate path — To list the alternate paths, use the **sysadm** sequence Availability –> Disk Failover –> Alternate Paths –> List. **Sysadm** displays path information as in the following.

```
Hostname    Alt Hostname
--------    ------------

primary-     alternate-remote-
hostname      hostname
```

Managing the monitors database

The **sysadm** sequence Availability –> Disk Failover –> Monitors lets you add, delete, modify, list, start, and stop monitors. The following sections describe these operations.

Adding a monitors entry — To add an entry to the **monitors** database, use the **sysadm** sequence Availability –> Disk Failover –> Monitors –> Add. **Sysadm** prompts for the following information:

`Host to Monitor`
: Specify the hostname you want this host to monitor for failure.

`Seconds Between Heartbeats: [0]`
: Enter the interval in seconds the monitor will sleep between message cycles to the other host. For example, **15**.

`Number of Retries: [0]`
: Enter the number of times the monitor will retry communication on each path before deciding the monitored host has failed. For example, **2**.

`Lost Pulse Action: [/etc/failover/failovermon_lost_pulse]`

Enter the full pathname to the lost-pulse script to be executed when the monitoring host decides that the monitored host has failed. This lost-pulse script (and the companion regain-pulse script) is not the same as the application script specified to OIF. The lost-pulse and regain-pulse scripts are executed by the failover monitor; their main purpose is to include **sysadm** take commands that transfer control of the disks. The application script is executed by OIF after the disk transfer; its main purpose is to include commands that start application(s). Data General supplies a default lost-pulse script in **/etc/failover/failovermon_lost_pulse**. You can edit this and use it or create another one.

`Regain Pulse Action: [/etc/failover/failovermon_regain_pulse]`

Enter the full pathname to the regain-pulse script to be executed when the monitoring host decides that the failed host has returned to service. See the comments under `Lost pulse action` above. Data General supplies a default regain-pulse script in **/etc/failover/failovermon_regain_pulse**. As with the lost–pulse script, you can edit this and use it or create another one.

`Start Monitor On Reboot? [no]`

Specify the value **yes** or **no**, to indicate whether you want to start the monitor automatically on each system reboot. For example, **y**.

`Start Monitor Now? [no]`

Specify the value **yes** or **no**, to indicate whether you want to start the monitor immediately after the successful completion of the Add operation. For example, **Y**.

As with many other **sysadm** sequences, it asks for confirmation by prompting `Ok to perform operation`. Answer **yes** to add the monitor as specified or **no** if you want to change a value.

Deleting a monitor — To remove a monitored host from the **monitors** database, use the **sysadm** sequence Availability –> Disk Failover –> Monitors –> Delete. **Sysadm** prompts for the following information:

`Monitored Host: [`*xxx*`]`

Enter the hostname you want this host to stop monitoring. This hostname must have been added earlier to the **monitors** database.

Modifying a monitor — To change the settings for a monitor entry in the **monitors** database, use the **sysadm** sequence Availability –> Disk Failover –> Monitors –> Modify. The **sysadm** prompts are exactly the same as for "Adding a monitors entry," above. Make any changes you want.

Listing monitors — To list the entries in the **monitors** database, use the **sysadm** sequence Availability –> Disk Failover –> Monitors –> List. **Sysadm** displays monitor information as in the following.

```
Hostname  Act  Reb Int Ret  Lost Pulse              Regain Pulse
--------- --- --- --- ---  ----------------------  ------------------------

junior    yes  yes 15  2  /etc/failover/lost_pulse  /etc/failover/regain_pulse
```

The columns correspond roughly to the Add and Modify prompts. Act indicates active or not active status; Reb indicates whether or not the monitor will run on reboot; Int is the heartbeat interval; Ret is the number of retries, and Lost pulse and Regain pulse show the pathnames of the lost-pulse and regain-pulse scripts.

Starting a monitor process — To start a monitor process, use the **sysadm** sequence Availability –> Disk Failover –> Monitors –> Start. You must have already defined the monitor with the **sysadm** Monitors Add sequence. **Sysadm** prompts

`Host to monitor: [`*xxx*`]`
: Enter the hostname you want this host to monitor. This hostname must have been added earlier to the **monitors** database.

After you answer and confirm, **sysadm** tries to start the monitor process on your system.

Stopping a monitor process — To stop a monitor process, use the **sysadm** sequence Availability –> Disk Failover –> Monitors –> Stop. The monitor process you want to stop must be running (usually started using a **sysadm** Monitor Add or Start sequence). **Sysadm** prompts

`Host to monitor: [`*xxx*`]`
: Enter the hostname you want this host to stop monitoring.

After you answer and confirm, **sysadm** tries to stop the monitor process on your system.

Disk failover troubleshooting

While testing failover, you can use two diagnostic aids: file **/var/adm/messages** and the **admpdisk** command or equivalent **sysadm** menu sequence.

Using /var/adm/messages

While testing a MIF setup, run **tail –f** on file **/var/adm/messages**. The **failovermon** daemon writes to **/var/adm/messages**. If you watch this file while simulating a failure, you will see messages indicating that the communications path is inaccessible, and that the various action scripts are being run. If one of the scripts is not executable or not available, then information saying this will be written to **/var/adm/messages**.

admpdisk(1M)

The **admpdisk** command with various options will tell you whether a disk is registered or if it can be registered.

Managing IP (Internet Protocol) takeover

This section builds on the previous major section "Managing Disk Failover" and assumes you understand that section.

The IP (Internet Protocol) takeover feature extends disk failover features to make a failed host's disk and resources available to NFS clients. IP takeover depends on and operates much the same way as Operator Initiated Failover (OIF) and Machine Initiated Failover (MIF). The features differ in that IP takeover transfers a floating network address from a host to another host, while disk failover transfers the disks themselves. Together the two features provide high availability of the failover disk file system to NFS client systems.

For routine operation, network hosts access the primary server host using this floating network address instead of a fixed, hard-wired address. If the primary host fails, the secondary host can take over the floating network address along with the failover disks. After a brief delay, NFS clients that have remote-mounted file systems on the failed host can continue running since the surviving host provides access to the file systems.

IP takeover is primarily useful because it provides higher availability of NFS remote-mounted file systems, **telnet**, and **rlogin** functionality. You can use IP takeover in conjunction with OIF or MIF. For the NFS clients to maintain access to the remote-mounted file system, the host's disk resources must have been transferred. You can transfer the disks manually using OIF or the DG/UX system can do it using an MIF failover script.

For IP takeover to work, two hosts on the same network or subnet must be connected in a configuration that will support OIF or MIF. That is, the hosts must be connected in a dual-initiator configuration

(or with a disk-array storage system, a cabinet-sharing configuration).

Figure 4–6 shows two hosts with NFS clients in a model configuration for IP takeover.

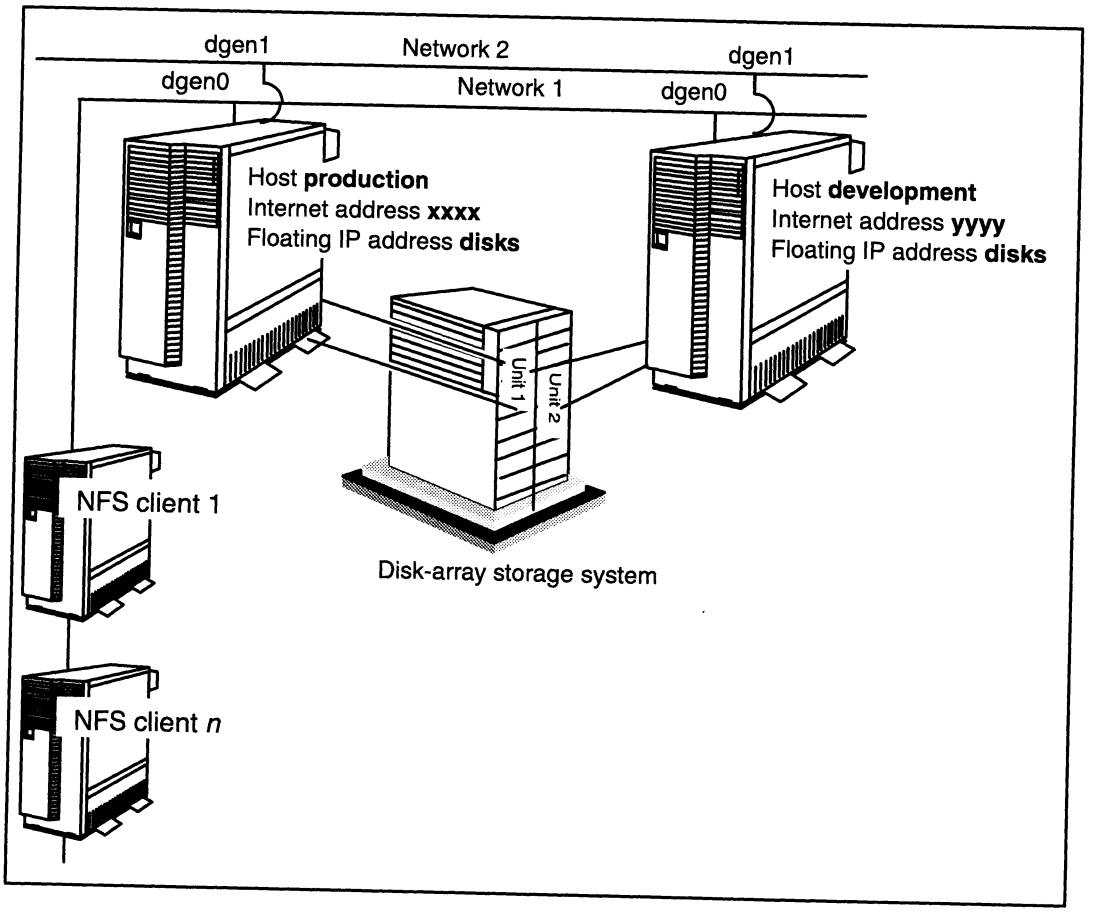


Figure 4–6 Two hosts with NFS clients in a sample IP takeover configuration

In Figure 4–6, host **production** provides NFS services to all the NFS clients. If **production** fails, the clients will lose NFS services — until the floating IP address server is transferred (along with the pertinent disks in the disk-array storage system) from **production** to **development**). OIF/MIF provide the means to transfer the disks; IP takeover provides the means to transfer the IP address.

Figure 4–6 shows two network interfaces for use in disk failover. The IP takeover example shown above uses only the network to which the **dgen0** interfaces is attached; **dgen1** serves as a secondary interface for the failover monitor.

Prerequisites for IP takeover

IP takeover requires the following conditions to be true.

- The network between the hosts must use an Ethernet or FDDI LAN controller; token ring LANs are not supported. The LAN controllers may be of different types, such as VLCi and integrated.
- The two hosts must be on the same network or subnet.
- You must have added the hostname and floating IP address to the **/etc/hosts** database (using the **sysadm** sequence Networking –> TCP/IP –> Hosts), and to the NIS database (if applicable). The NFS clients must also have the hostnames and floating IP addresses added to these databases.
- The file systems to be involved in the takeover must be exported (made exportable and mounted via **sysadm**).
- You must have set up the network and host hardware and DG/UX software for OIF and, if you want automatic transfer of the network address, for MIF. Setting up OIF and MIF are explained earlier in this chapter.

  You do not have to change the OIF/MIF hardware configuration or build a new kernel to use IP takeover.

Setting up and using IP takeover

To set up IP takeover for two hosts, you must execute the following steps (assuming the prerequisite conditions above have been fulfilled).

1. On the host that will be the primary NFS server on the network, with the network running, add the IP takeover entry. To do this, use the **sysadm** sequence Availability –> IP Takeover –> Add. When you have added the IP entry, be sure to answer **yes** to the synchronize prompt. The synchronize step enters the floating IP address in the IP takeover database of the other host, allowing it to take over the name. If you do not synchronize databases, the system flags them as out of synchronization (noted as NOTSYNC on the **sysadm** Availability –> Disk Failover –> Hosts –> List status display); and it will not allow you to transfer control of the IP address.
2. With the IP takeover entry added and databases synchronized, you can transfer the IP address as necessary. There are several ways to transfer this address, depending on whether both hosts are running, as follows.
   - When the giving host has failed, you can use the taking host to transfer control to the taking system with the **sysadm** sequence Availability –> IP Takeover –> Take and the **Use Trespass** option.
   - When the giving host is functioning normally, you can transfer control using either system:

from the giving host using the **sysadm** sequence Availability –> IP Takeover –> Give; or

from the taking host using the sequence Availability –> IP Takeover –> Take *without* the **Use Trespass** option.

Managing the IP takeover database

On each host, DG/UX uses the database file named **/etc/failover/failoverip** to manage IP takeover. The **failoverip** database contains information about IP host names and the network interface used on each host. You maintain this database using a **sysadm** menu sequence described here or the equivalent **admfailoverip**(1M) command. A manual page provides detailed information on this command.

The **sysadm** sequence Availability –> IP Takeover lets you add, delete, modify, and list database entries; give and take the floating IP address; start and stop the floating IP address mechanism; and synchronize databases. The following sections describe each operation.

Adding an IP takeover entry

Before anyone can transfer the floating IP address to another host, you must add the address to the IP takeover database using the Availability –> IP Takeover –> Add operation. After entering the needed information, you synchronize databases to add the information to the other host's failover databases.

The Add operation prompts for the following information:

`Host name:` Enter the name of the remote host that will get the IP address if your host fails. For the sample hosts in Figure 4–6, earlier,, host **production** would add hostname **development**).

`Floating Address Name:`
Enter the name for the floating IP address. Unlike most hostnames, which are associated with a single interface on a single system, the floating address name can float between the two hosts for use by either host's network interface. This name will allow NFS clients to access remote file systems on disks that were transferred to the other host by OIF/MIF scripts. (The floating IP address must always belong to the same host that the disks belong to.) NFS clients will mount the remote file system using the floating address name you specify here; they will not use a hostname.

Since the same floating address name will serve on both hosts, choose a name that identifies the disk resources, not a host. The floating IP address is unique to IP takeover and must differ from other hostnames on the network. You can use as many as nine DG/UX filename characters. For the example in Figure 4–6, the address is named **acmedisks**.

`Local Network Interface:`
Enter the name of the local host's network interface to use for IP takeover. For example, **dgen0** or **cien0**.

`Remote Network Interface:`
Enter the name of the remote host's network interface to use for IP takeover. For example, **dgen0** or **cien0**.

`Start Floating IP Address On System Reboot? [yes]`
Specify **yes** or **no** to indicate whether you want to start the IP takeover mechanism automatically on each system reboot.

`Start Floating IP Address Now? [no]`
Specify **yes** or **no** to indicate whether you want to start the IP takeover mechanism immediately after the successful completion of the Add operation.

`Synchronize database? [no]`
This option updates the information about this floating IP address entry in the failover databases on the remote system. If the remote system is inaccessible, synchronization will fail. When the remote system becomes accessible again, synchronize databases with the Synchronize operation, explained later. Until you synchronize databases, the system considers the local IP failover database as out of synchronization (noted as NOTSYNC on the **sysadm** Availability –> Disk Failover –> Hosts –> List status display). With the databases out of synchronization, no one can use the IP takeover mechanism with that host.

Adding the IP takeover entry is the essential software task required for IP takeover. For IP takeover maintenance tasks, continue with the next sections.

Deleting an IP takeover entry

To remove an IP takeover entry from the IP failover database, select the operation Availability –> IP Takeover –> Delete. **Sysadm** prompts for the following information:

`Host name:` Enter the name of the remote host from whose IP failover database you want to delete the floating IP address.

`Floating Address Name:`
Enter the name of the floating IP address you want to delete.

`Synchronize database? [no]`
This option removes the information for the floating IP address from the other host's failover databases. We recommend that you answer **yes** to this query. For more information, see "Synchronize database" under the previous section "Adding an IP Takeover Entry."

Modifying an IP takeover entry

To change the information about an IP takeover entry in your host's failover database, select the operation Availability –> IP Takeover –> Modify. **Sysadm** prompts for the hostname, floating address name, and other information as shown in the earlier section "Adding an IP Takeover Entry."

Listing IP takeover entries

To display information about local IP takeover entries, select Availability –> IP Takeover –> List. The display includes the following information:

- the hostname of the remote host;
- the floating IP address name to share;
- the local network interface name;
- the remote network interface name;
- yes or no flag that shows whether the floating IP address mechanism will be restarted when the system is rebooted; and
- hostname the floating IP address is active on, or NONE if the address is not active on either host.

For synchronization status (not included on the List screen), use the **sysadm** sequence Availability –> Disk Failover –> Hosts –> List.

Giving a floating IP address

To transfer control of an IP Takeover address from your host to another, select the **sysadm** operation Availability –> IP Takeover –> Give. For transfer to work, the failover databases must be synchronized, as noted on the Availability –> Disk Failover –>

Multi-path disk I/O

With multi-path disk I/O, you can configure your system to automatically reroute disk I/O. If the primary disk path fails, you can set the DG/UX system to automatically switch from the primary disk path to a backup path. The process is transparent to applications and protects your system from both SCSI disk controller failures and I/O channel failures.

Multi-path disk I/O has the following restrictions:

- Your system must support multiple interfaces to the same disk drive.

  Note that multi-path disk I/O can have three backup paths, and those paths do not need to be idle.

- Disk interfaces must be one of the following SCSI adapter types:
  - **dgsc**
  - **ncsc**

  Unlike multi-path LAN I/O, multi-path disk I/O does not require either the **mpl()** device driver or the **failovermon** monitor. It does require the **vdmmpio()** device driver, but this is built into the kernel by default. When the DG/UX system detects a failure on an I/O path, the system automatically switches to a disk's backup I/O path.

Adding multi-path disk I/O entries

Select Availability –> Multi-Path I/O –> Disk –> Add to add an entry to the **iopath.params** database. The Add operation displays the following queries:

`Primary Disk I/O Path`
: Enter the device specification of the disk path to be used as the primary I/O path. For example, you could enter `sd(ncsc(0,6),0,0)` for a CLARiiON disk-array that supports two SPs.

`Backup Disk I/O Path`
: Enter the device specification of the disk path to be used as the backup I/O path. For example, you could enter `sd(ncsc(0,6),1,0)`.

  Since you can have up to three backup paths, the system displays this query three times. If you do not want to enter additional paths, press the Carriage Return key at the query.

Starting the IP takeover mechanism

Normally, the IP takeover mechanism starts automatically, according to the answer you gave to the start-on-reboot prompt when you added or modified this IP takeover entry. To start (enable) the IP Takeover mechanism on your host, use the **sysadm** sequence Availability –> IP Takeover –> Start. **Sysadm** prompts for the remote hostname for which you want to start IP takeover and then prompts for the floating address name. Starting of takeover affects only the current state on the local host; remote host databases are not affected by this. When the start operation succeeds, **sysadm** will display a "Floating IP address started" message.

Stopping the IP takeover mechanism

To stop (disable) the IP takeover mechanism on your host, use the **sysadm** sequence Availability –> IP Takeover –> Stop. **Sysadm** prompts for the remote hostname for which you want to stop IP takeover and then prompts for the floating address name. Stopping of takeover affects only the current state on the local host; remote host databases are not affected by this. The action on the next reboot will depend on the answer you gave to the start-on-reboot prompt when you added or modified this entry.

Synchronizing databases

To make all the failover databases on a remote system consistent with the ones on your system, select the operation Availability –> IP Takeover –> Synchronize. **Sysadm** then prompts for the hostname whose databases you want to synchronize with yours. **Sysadm** scans the local databases and copies to the remote host the entries for that host. **Sysadm** does *not* copy the remote **failoverip** entries to the local database.

Some operations that change the local IP takeover database, such as Add, Delete, and Modify, offer the option of synchronizing databases. If you select this option, **sysadm** copies the changes as appropriate to the **failoverip** database on the remote system. If you do not synchronize databases, the system flags the local databases as out of sync, or NOTSYNC, and restricts you from giving any floating IP address to another system until you have synchronized databases.

Mounting file systems for use with IP takeover

For the NFS clients to access the file systems on the failover disks, the clients must mount the file systems using the floating IP network address. This allows the clients to access the file system as long as the floating IP network address is available.

Hosts –> List status display. If they are not synchronized (NOTSYNC), synchronize the databases with the Synchronize operation, explained later.

IMPORTANT: Transferring a floating IP address prevents NFS clients from accessing your host by the network interface associated with this address. You should do this only after the failover disk(s) have been transferred by OIF or MIF operations, explained in the previous section on disk failover. NFS clients may regain access to your host through the other host after a few moments. Any user who was running a **telnet** or **rlogin** session must restart the session.

Sysadm prompts for the following information.

`Host Name:` Enter the name of the remote host that will get the IP address.

`Floating Address Name:`
Enter the name for the floating IP address you want to transfer to the other host.

Sysadm then prompts for confirmation. After you confirm, **sysadm** consults the IP failover databases and performs the following actions:

- on the local host, checks the IP takeover database and, if the floating IP address is active on this host, deactivates the address; and
- on the remote host, activates the floating IP address.

Taking a floating IP address

The process of taking a floating IP failover address involves having the **admfailoverip**(1M) code consult the failover databases and perform actions similar to those described previously for giving away a floating IP address.

To transfer a floating IP address from another host to your own, select the **sysadm** operation Availability –> IP Takeover –> Take. The operation requires that you enter the remote hostname and the floating address name you wish to take. You can take a floating IP address only if the administrator of the remote system has added a floating IP address for your host to the failover databases and has synchronized databases.

Optionally, you may select the Use Trespass option. This option is intended for situations where the remote system is down and therefore unable to respond. If the remote system is functioning normally and you select the Use Trespass option, the transfer will fail and your system will display an error message.

While simulating a failure, run **tail –f** on file **/var/adm/messages**. You will see messages indicating that the communications path is inaccessible, and that the various action scripts are being run. If one of the scripts is not executable or not available, then information saying this will be written to **/var/adm/messages**.

After starting the IP takeover mechanism, you can check status from the starting host as follows:

- To make sure that the floating IP address interface has been added to the file **/etc/tcpip.params**, examine that file.
- To make sure that the failover database shows the exported file systems, on the host that has the disks, use the **sysadm** sequence Availability –> Disk Failover –> Giveaway –> List; on the host that does not have the disks, use the **sysadm** sequence Availability –> Disk Failover –> Takeaway –> List. The export flag field of the display should show "YES"and this should be followed by any export options specified.
- To make sure the file system has been exported, on the host that has the disks, use the sysadm sequence File System –> Local Filesys –> List and at the query "Which local file systems?," answer with "exported."
- To make sure that the interface is up and running, on the host that has the disks, type

```
# admipinterface -o list -g failover↵
```

From an NFS client, to make sure that the floating IP address interface is available, you can type

ping *floating-ip-address-name*

An example follows:

```
# ping acmedisks↵
```

Managing multi-path I/O

The high-availability features discussed in the previous sections were for configurations with multiple systems. Multi-path I/O is a feature that enables you to protect a single system from LAN and disk interface failures. There are the following two types of multi-path I/O:

- Multi-path LAN I/O
- Multi-path disk I/O

Both multi-path I/O types enable your system to automatically redirect I/O if a LAN or disk controller fails. To use these features,

For two serving hosts to be able to mount a file the original owning host must use the floating IP network address when it originally mounts the file system; if the host does not use the floating IP address, the physical disk failover may fail.

Make sure the mount point directory is what your application expects. If you must change the local mount point directory name to satisfy the application or IP failover, then you will need to change mount point specification in the OIF giveaway database and then synchronize databases.

Technical details on floating IP mounts

When one of the server hosts uses the floating IP address to mount a file system, this results in a two–level mount scheme on that server. The lower-level mount will be a file system on the actual virtual disk. It results in an **/etc/fstab** entry of the following form:

/dev/dsk/*virtual-disk-name* */file-system-name* **dg/ux rw x 0**

For example,

```
/dev/dsk/acme_data      /acme_data_fs   dg/ux rw x 0
```

This file system (*/file-system-name*) needs to be exported when the physical disk is added to the failover databases. The OIF software will store the export information in the failover databases and ensure the file system is mountable by NFS clients. This lower level mount will be managed by the OIF software during a failover of the disk. Because the file system is not directly accessed (no process opens a file in */file-system-name*), this mount can be managed when the system controlling the disks fails.

The higher level mount is to mount the exported file system the same way as the NFS clients. This results in a **/etc/fstab** entry of the following form:

floating-ip-address:*/file-system-name /mount-point-directory* **nfs rw,hard,intr,bg x x**

For example,

```
acmedisks:/acme_data_fs   /acme   nfs  rw,hard,intr,bg  x  x
```

All applications that need to access the file system will do so by accessing */mount-point-directory*; in this example, **/acme**.

Troubleshooting IP takeover

The failover monitor writes to the file **/var/adm/messages**. While testing IP takeover, as with other types of failover, you can look at the file **/var/adm/messages** for messages.

Multi-path LAN I/O

With multi-path LAN I/O, you can configure your system to automatically reroute LAN I/O. You can configure most types of LAN interfaces in AViiON systems so that two LAN interface cards can be connected to the same LAN. If the primary LAN interface fails, you can set the DG/UX system to automatically switch from the primary LAN interface to a backup LAN interface. The process is transparent to applications and protects your system from both LAN interface failures and I/O channel failures. Multi-path LAN I/O works on both Ethernet and FDDI networks.

Multi-path LAN I/O has the following restrictions:

- Your system must support two LAN network interfaces, one of which is idle.
- Both interfaces must be the same type, such as **dgen**.
- LAN network interfaces must be one of the following types:
  - **cien**
  - **dgen**
  - **hken**
  - **pefn**

You must add the **mpl()** device driver to the kernel to implement multi-path LAN I/O. Note that the driver is not added to the kernel by default. You must add it explicitly, rebuild your kernel, and reboot your system. Once the **mpl** driver is in the kernel (that is, **/dev/mpl** is present), you can use **sysadm** to set up your two LAN interfaces in a multi-path relationship.

The multi-path LAN I/O process follows this sequence of events:

1. The **failovermon** monitor periodically requests that the **mpl** driver have both LAN interfaces send a message intended for the other interface.

   For more information on **failovermon**, see the **failovermon**(1M) manual page.

2. If both interfaces receive their message, the **mpl** driver determines that the I/O paths are open and takes no further action until the next check.

3. If only one interface receives a message, the **mpl** driver uses that controller's driver to determine the state of the interface.

4. If the primary interface has failed, the **mpl** driver switches I/O to the backup interface.

your system must support multiple disk controllers to the same disk drive for multi-path disk I/O and multiple LAN controllers to the same LAN for multi-path LAN I/O. One of these multiple controllers is established as the primary controller, while one or more other controllers serve as backup I/O paths. When the system detects a failure in the primary I/O controller, it can automatically switch I/O to one of the backup controllers. You can also manually switch I/O to a different path.

Note that in addition to the **sysadm** interfaces discussed later you can also use the **admiopath** command to manage multi-path I/O. See the **admiopath**(1M) manual page for more information.

Multi-path I/O database

Both types of multi-path I/O use the **/etc/iopath.params** database to control their operation. This database contains the devices set up in a multi-path I/O relationship and information about whether that relationship should be started when the system boots.

Following is an example of the **iopath.params** file:

```
#############################################################################
#
#                       iopath database parameters
#
#############################################################################
#
#  The entries in this file correspond to the I/O path names
#  that will be bound together to allow the system to perform
#  multi-path I/O. The format of these entries is:
#
#       <primary_path>          Name of primary I/O path.
#       <start_on_reboot>       Flag indicating relationship starts
#                               when system is rebooted.
#       <backup_path>           Name of first backup I/O path.
#       <backup_path>           Name of second backup I/O path (disk only).
#       <backup_path>           Name of third backup I/O path (disk only).
#
#----------------------------------------------------------------------------
dgen0 YES dgen1 NONE NONE
sd(ncsc(1,7),2,1) NO sd(ncsc(2,6),3,1) NONE NONE
```

The name of the primary I/O path is used as the key in all related **sysadm** operations to locate entries in this database. For more information on the **iopath.params** database, see the **iopath.params**(4M) manual page.

`Primary LAN I/O Path`
: Enter the device specification of the primary LAN interface in the multi-path LAN I/O relationship you want to delete. For example, you could enter `dgen0`.

This operation stops the multi-path I/O relationship for the specified controller and deletes its entry from the **iopath.params** database.

Modifying multi-path LAN I/O entries

Select Availability –> Multi-Path I/O –> LAN –> Modify to change an entry in the **iopath.params** database. The Modify operation displays the same queries as the Add operation:

`Primary LAN I/O Path`
: Enter the device specification of the LAN interface to be used as the primary I/O path.

`Backup LAN I/O Path`
: Enter the device specification of the backup LAN interface to be used as the backup I/O path.

`Start Multi-Path LAN I/O on Reboot?`
: Enter `yes` if you want to automatically start this multi-path relationship when the system is rebooted and returns to init level 3. Enter `no` if you do not.

`Start Multi-Path LAN I/O Now?`
: Enter `yes` if you want to stop and restart the modified multi-path relationship when you complete the Modify operation. Enter `no` if you do not.

Displaying multi-path LAN I/O entries

Select Availability –> Multi Path I/O –> LAN –> List to display an **iopath.params** database entry to standard output. The List operation displays the following query:

`Primary LAN I/O Path`
: Enter the device specification of the primary LAN controller in the **iopath.params** database entry you want to list. For example, you could enter `dgen0`. Entering `all` lists all entries in the database.

The List operation reports the following information:

- Name of the primary I/O path
- Flag indicating whether to start the multi-path relationship when the system boots

Adding multi-path LAN I/O entries

Select Availability –> Multi-Path I/O –> LAN –> Add to add an entry to the **iopath.params** database. The Add operation displays the following queries:

`Primary LAN I/O Path`

Enter the device specification of the LAN interface to be used as the primary I/O path. For example, you could enter `dgen0`.

The primary path should be the interface used by the upper level networking software, such as TCP/IP. In the case of TCP/IP, the primary interface should be the one listed in the **/etc/tcpip.params** file.

`Backup LAN I/O Path`

Enter the device specification of the secondary LAN interface to be used as the backup I/O path. For example, you could enter `dgen1`. Note that this controller must be of the same type as the primary interface. For example, both controllers could be of type **dgen**.

The backup path should not be the interface used by default by the upper level networking software. If the backup interface is being used, Multi-path LAN I/O will not start for that pair of interfaces.

`Start Multi-Path LAN I/O on Reboot?`

Enter `yes` if you want to automatically start this multi-path relationship when the system boots. Enter `no` if you do not. This process is controlled by the **rc.failover** script.

`Start Multi-Path LAN I/O Now?`

Enter `yes` if you want to start this multi-path relationship when you complete the Add operation. Enter `no` if you do not.

If a **failovermon** monitor is not presently running on the system, this operation automatically starts the monitor to control the timing of the **mpl** driver's checks on the state of the I/O paths.

Deleting multi-path LAN I/O entries

Select Availability –> Multi-Path I/O –> LAN –> Delete to stop a multi-path LAN I/O relationship and delete its entry from the **iopath.params** database. The Delete operation displays the following query:

`Primary LAN I/O Path`
: Enter the device specification of the primary LAN interface in the multi-path LAN I/O relationship for which you want to switch I/O paths. For example, you could enter `dgen0`.

`Backup Physical LAN`
: Enter the device specification of the inactive I/O path to which you want to switch LAN I/O. For example, you could enter `dgen1`.

Indicating that a LAN I/O path is repaired

Select Availability –> Multi-Path I/O –> LAN –> Repaired to indicate that a failed path in a multi-path LAN I/O relationship has been repaired. An example of a situation when you would need to use this operation is when the cable to a controller for a **hken** interface becomes detached. In this case, multi-path LAN I/O would automatically switch to the backup interface. However, after you reconnect the cable, you would need to use the Repaired operation on the primary interface before you could switch I/O back to it.

The Repaired operation displays the following queries:

`Primary LAN I/O Path`
: Enter the device specification of the primary LAN interface in the multi-path LAN I/O relationship for which you want to indicate an I/O path has been repaired. For example, you could enter `dgen0`.

`Backup Physical LAN`
: Enter the device specification of the repaired I/O path.

You should use this operation only after a failed interface has been repaired or replaced and then reconfigured into your system. However, if you had to bring your system down to repair the problem, you do not need to use the Repaired operation.

Note that this operation does not change the currently active I/O path. It just tells the system that a failed path has been repaired and is now available for use.

Resetting the multi-path LAN I/O relationship

Select Availability –> Multi-Path I/O –> LAN –> Reset to restore the multi-path LAN I/O relationship to its original state. This operation returns LAN I/O to the primary I/O path. The Reset operation displays the following query:

`Primary LAN I/O Path`
: Enter the device specification of the primary LAN interface in the multi-path LAN I/O relationship you want to reset. For example, you could enter `dgen0`.

- Name of the currently active I/O path
- Name of the backup I/O path

  Only one backup path is used in multi-path LAN I/O.

  Following is an example of the List operation's output:

```
Primary Path    Rbt    Active Path   Backup Path(s)
------------    ----   -----------   ---------------
dgen0           YES    dgen0         dgen1 NONE NONE
```

Starting multi-path LAN I/O

Select Availability –> Multi-Path I/O –> LAN –> Start to start the multi-path LAN I/O relationship for a specified I/O path. The Start operation displays the following query:

`Primary LAN I/O Path`

Enter the device specification of the primary LAN interface in the multi-path LAN I/O relationship you want to start. For example, you could enter `dgen0`. Entering `all` starts all the relationships listed in the **iopath.params** database.

If a **failovermon** monitor is not presently running on the system, this operation automatically starts the monitor to control the timing of the **mpl** driver's checks on the state of the I/O paths.

Stopping multi-path LAN I/O

If a LAN I/O path other then the primary one is currently active, you must switch back to the primary path prior to stopping the relationship.

Select Availability –> Multi-Path I/O –> LAN –> Stop to stop the multi-path LAN I/O relationship for a specified I/O path. The Stop operation displays the following query:

`Primary LAN I/O Path`

Enter the device specification of the primary LAN interface in the multi-path LAN I/O relationship you want to stop. For example, you could enter `dgen0`. Entering `all` stops all the relationships listed in the **iopath.params** database.

Switching to an inactive LAN I/O path

Select Availability –> Multi-Path I/O –> LAN –> Switch to manually switch LAN I/O from the currently active I/O path to an inactive path. The Switch operation displays the following queries:

`Start Multi-Path Disk I/O on Reboot?`

Enter `yes` if you want to automatically start this multi-path relationship when the system boots. Enter `no` if you do not. This process is controlled by the **rc.failover** script.

`Start Multi-Path Disk I/O Now?`

Enter `yes` if you want to start this multi-path relationship when you complete the Add operation. Enter `no` if you do not.

Deleting multi-path disk I/O entries

Select Availability –> Multi-Path I/O –> Disk –> Delete to stop a multi-path disk I/O relationship and delete its entry from the **iopath.params** database. This operation can also be used to delete individual backup paths from the entry in the database.

The Delete operation displays the following query:

`Primary Disk I/O Path`

Enter the device specification of the primary disk path in the multi-path disk I/O relationship on which you want to perform the Delete operation.

`Backup Disk I/O Path`

Enter the device specification of a secondary disk path in the multi-path disk I/O relationship you want to delete. If there are other backup paths in the relationship, this operation deletes only the specified backup path. Enter `all` to delete the entire entry.

This operation stops the multi-path I/O relationship for the specified controllers and deletes its entry from the **iopath.params** database.

Modifying multi-path disk I/O entries

Select Availability –> Multi-Path I/O –> Disk –> Modify to change an entry in the **iopath.params** database. The Modify operation displays the same queries as the Add operation:

`Primary Disk I/O Path`

Enter the device specification of the disk path to be used as the primary I/O path.

`Backup Disk I/O Path`

Enter the device specification of the disk path to be used as the backup I/O path.

Since you can have up to three backup paths, the system displays this query three times. If you do not want to modify additional paths, press the Carriage Return key at the query.

`Start Multi-Path Disk I/O on Reboot?`

Enter `yes` if you want to automatically start this multi-path relationship when the system boots. Enter `no` if you do not.

`Start Multi-Path Disk I/O Now?`

Enter `yes` if you want to stop and restart the modified multi-path relationship when you complete the Modify operation. Enter `no` if you do not.

Displaying multi-path disk I/O entries

Select Availability –> Multi-Path I/O –> Disk –> List to display an **iopath.params** database entry to standard output. The List operation displays the following query:

`Primary Disk I/O Path`

Enter the device specification of the primary disk path in the **iopath.params** database entry you want to list. For example, you could enter `sd(ncsc(2,6),0,0)`. Entering `all` lists all entries in the database.

The List operation reports the following information:

- Name of the primary I/O path
- Flag indicating whether to start the multi-path relationship when the system boots
- Name of the currently active I/O path
- Name of the backup I/O paths

Following is an example of the List operation's output:

```
Primary Path       Rbt  Active Path        Backup Path(s)
------------       ---  -----------        --------------
sd(ncsc(0,6),0,0)  NO   sd(ncsc(0,6),0,0)  sd(ncsc(0,6),1,0) NONE NONE
```

Starting multi-path disk I/O

Select Availability –> Multi-Path I/O –> Disk –> Start to start the multi-path disk I/O relationship for a specified I/O path. The Start operation displays the following query:

`Primary Disk I/O Path`

Enter the device specification of the primary disk path in the **iopath.params** database entry you want to start. For example, you could enter `sd(ncsc(0,6),0,0)`. Entering `all` starts all the relationships listed in the **iopath.params** database.

Stopping multi-path disk I/O

Select Availability –> Multi-Path I/O –> Disk –> Stop to stop the multi-path disk I/O relationship for a specified I/O path. The Stop operation displays the following query:

`Primary Disk I/O Path`
: Enter the device specification of the primary disk path in the **iopath.params** database entry you want to stop. Entering `all` stops all the relationships listed in the **iopath.params** database.

Switching to an inactive disk I/O path

Select Availability –> Multi-Path I/O –> Disk –> Switch to manually switch disk I/O from the currently active I/O path to an inactive path. The Switch operation displays the following queries:

`Primary Disk I/O Path`
: Enter the device specification of the primary disk path in the multi-path disk I/O relationship for which you want to switch I/O paths.

`Inactive Disk Path`
: Enter the device specification of the inactive disk path to which you want to switch disk I/O.

Indicating that a disk I/O path is repaired

Select Availability –> Multi-Path I/O –> Disk –> Repaired to indicate that a failed path in a multi-path disk I/O relationship has been repaired. An example of a situation when you would need to use this operation is when a disk controller fails in an AViiON server that allows replacement of the controller without bringing the system down. In this case, multi-path disk I/O would automatically switch to the backup interface. However, after you replace the failed controller, you would need to use the Repaired operation on the primary interface before you could switch I/O back to it.

The Repaired operation displays the following queries:

`Primary Disk I/O Path`
: Enter the device specification of the primary disk path in the multi-path disk I/O relationship for which you want to indicate an I/O path has been repaired.

`Repaired Disk Path`
: Enter the device specification of the repaired I/O path.

You should use this operation only after a failed interface has been repaired or replaced and then reconfigured into your system.

However, if you had to bring your system down to repair the problem, you do not need to use the Repaired operation.

Note that this operation does not change the currently active I/O path. It just tells the system that a failed path has been repaired and is now available for use.

Resetting the multi-path disk I/O relationship

Select Availability –> Multi-Path I/O –> Disk –> Reset to restore the multi-path disk I/O relationship to its original state. This operation returns disk I/O to the primary I/O path. The Reset operation displays the following query:

`Primary Disk I/O Path`

Enter the device specification of the primary disk path in the multi-path disk I/O relationship you want to reset.

End of Chapter

5 Setting up single system high-availability features

This section presents an example of how to set up and operate Data General's high-availability components in a single system environment.

This example considers the case of Acme Novelties. Acme was a mail order company that had outgrown its old proprietary minicomputer system. This system could no longer handle the transaction load that Acme's order processing application required from their host. Acme's executives decided it was time to move to a new, open systems computing environment.

After an extensive evaluation of vendors, Acme decided to purchase the following system components from Data General:

- AV 8500 AViiON server with the DG/UX operating system and AV/Alert diagnostic support system
- CLARiiON deskside disk-array storage system
- CLARiiON deskside tape-array storage system
- Two LAN controllers for multi-path LAN I/O
- Two SCSI-2 adapters for multi-path disk I/O
- ORACLE RDBMS order processing application

Figure 5–1 shows how Acme's new system will look after all of these components have been set up.

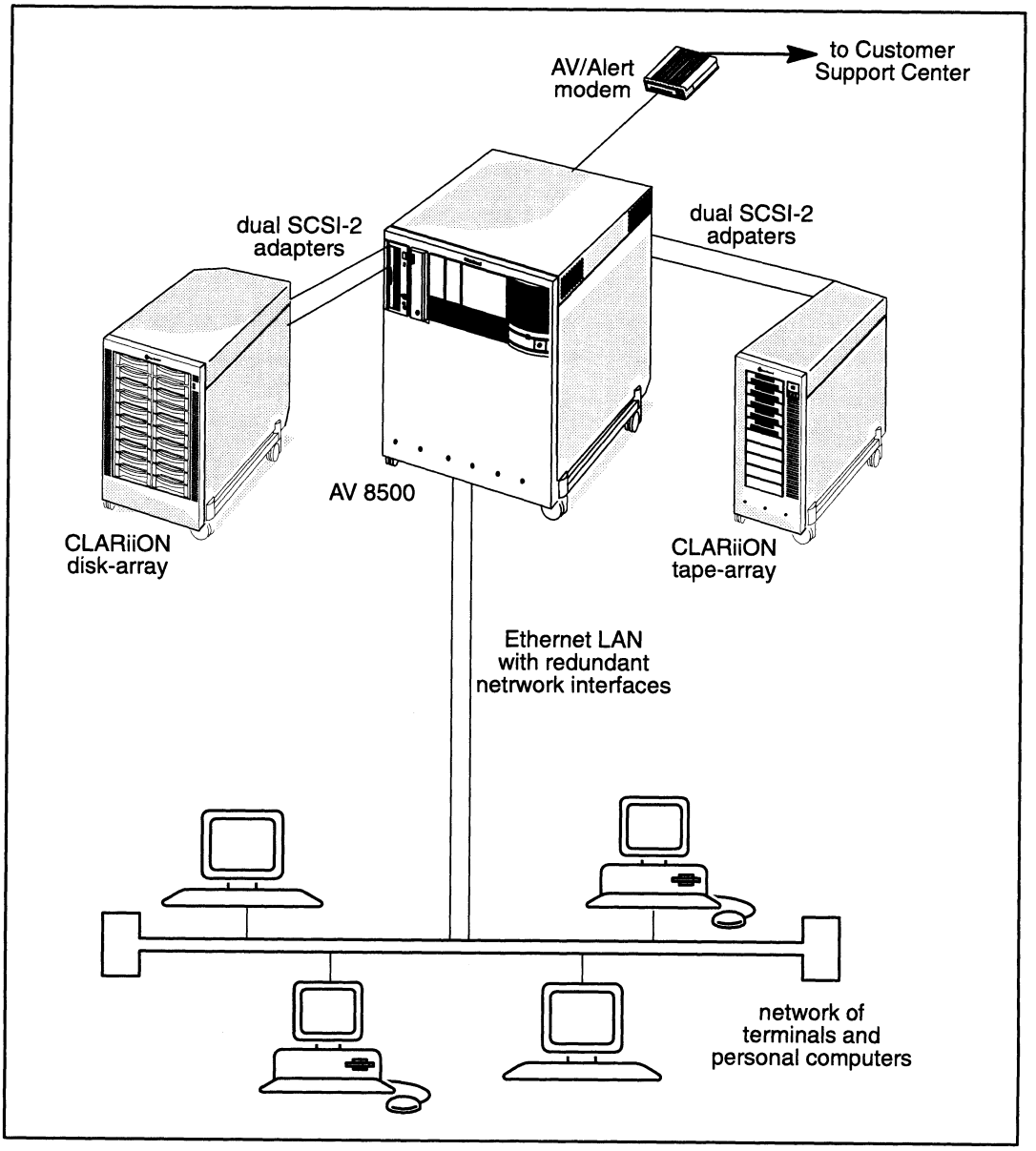


Figure 5–1 Acme's new computer system

Setting up the system

Now that they had their new system, Acme's technical staff wanted to configure it for maximum high availability. This section describes some of the steps they took to set up the high-availability features on the various components of their system.

Setting up the AViiON server's high-availability features

The Acme technical staff set up the following high-availability features on their AViiON server:

- Enable automatic reboot on operating system failure and boot on error during powerup
- Set up modem for the AV/Alert service
- Set up Local Area Network (LAN) controllers for Multi-path LAN I/O

The following sections describe how to set up these features.

Enabling automatic reboot and boot on error

Acme followed these steps to enable automatic rebooting on operating system failure and boot on error during powerup on the AViiON:

1. **Enter "f" at the System Control Monitor (SCM) prompt.**

   The system displays the following menu:

```
View or Change System Configuration

1  Change default boot paths
2  Setup dual-initiator SCSI IDs
3  Modify port parameters
4  View system configuration
5  Modify system parameters
6  Return to previous screen

Enter choice->
```

2. **Select choice 1 from the View or Change System Configuration menu.**

   The system displays the following menu:

```
        Change Default Boot Paths

1  Change 1st path [sd(ncsc(1),0,0]
2  Change 2nd path [ ]
3  Change 3rd path [ ]
4  Auto-reboot/Boot on error [Disabled]
5  Return to previous screen

Enter choice->
```

3. **Select choice 4 from the Change Default Boot Paths menu.**

   The system enables auto-reboot and reboot on error. Note that you can also change the default boot paths on this menu.

Setting up the AV/Alert modem

Before Acme's technical staff can set up and test the AV/Alert software, they have to set up their modem. Figure 5–2 shows how the modem is set up.

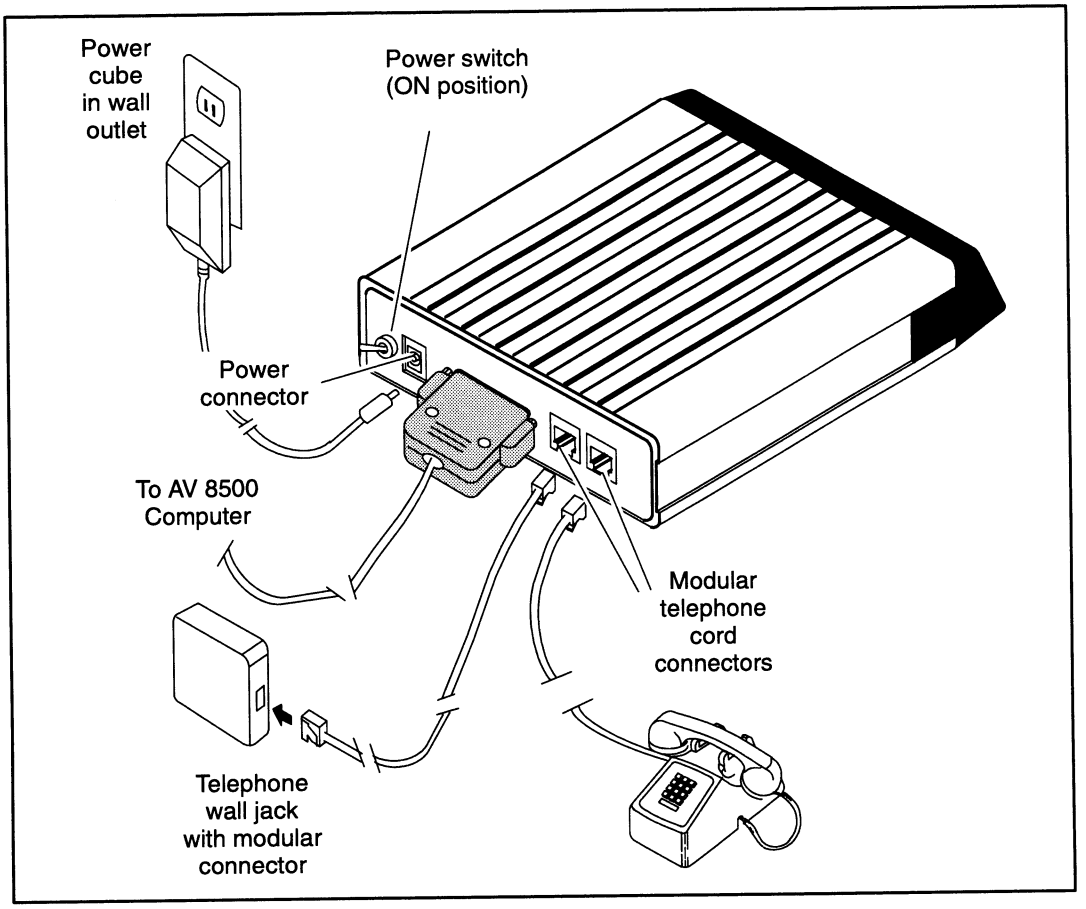


Figure 5–2 AV/Alert modem

The modem is connected to the port labeled SERVICE on the back of the AV 8500. Figure 5–3 shows where that port is located on the AViiON.

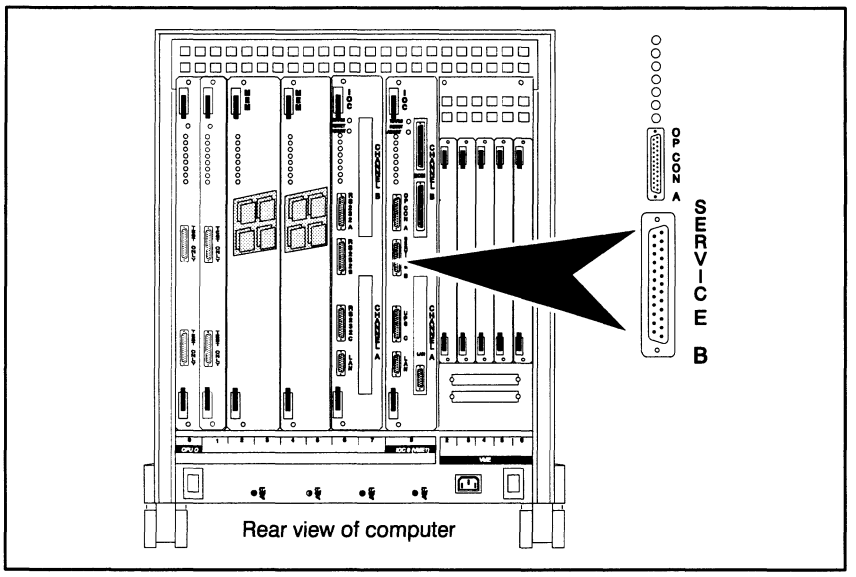


Figure 5–3 AV/Alert Port on AV 8500

Setting up the LAN controllers

To support multi-path LAN I/O, the AV 8500 must have two LAN controller cards connected to the same network. One of these cards is used as the primary network interface. The other card serves as a backup and is idle most of the time.

Setting up the CLARiiON disk-array's high-availability features

The Acme technical staff set up the following high-availability features on their CLARiiON disk-array storage system:

- Set up dual SCSI-2 adapter channels
- Bind together five disk modules as a RAID-5 group for the ORACLE database

The following sections describe how to set up these features.

Setting up the SCSI-2 adapter channels

To set up their CLARiiON storage system for maximum high availability, the Acme staff connected their CLARiiON storage system as shown in Figure 5–4.

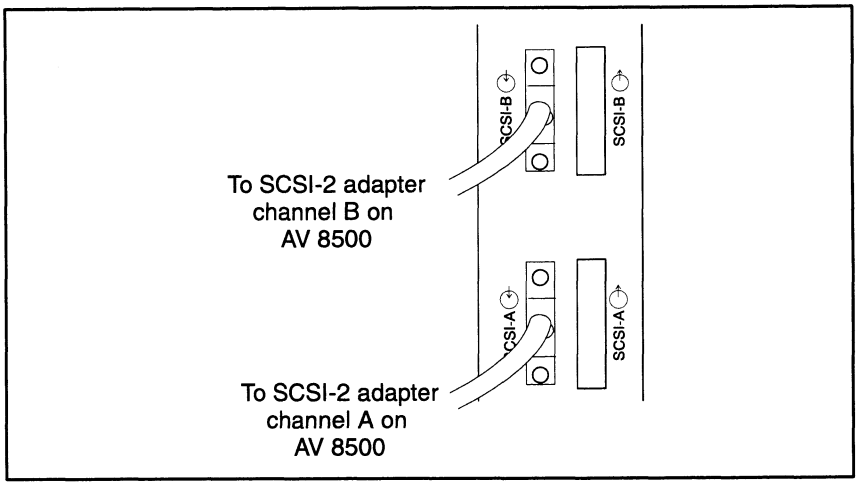


Figure 5–4 CLARiiON disk-array SCSI-2 cabling

This cabling configuration sets the CLARiiON system up in a dual-adapter-with-dual-SP configuration providing the maximum level of high availability. Figure 5–5 shows one of the SCSI adapter channels on the AV 8500.

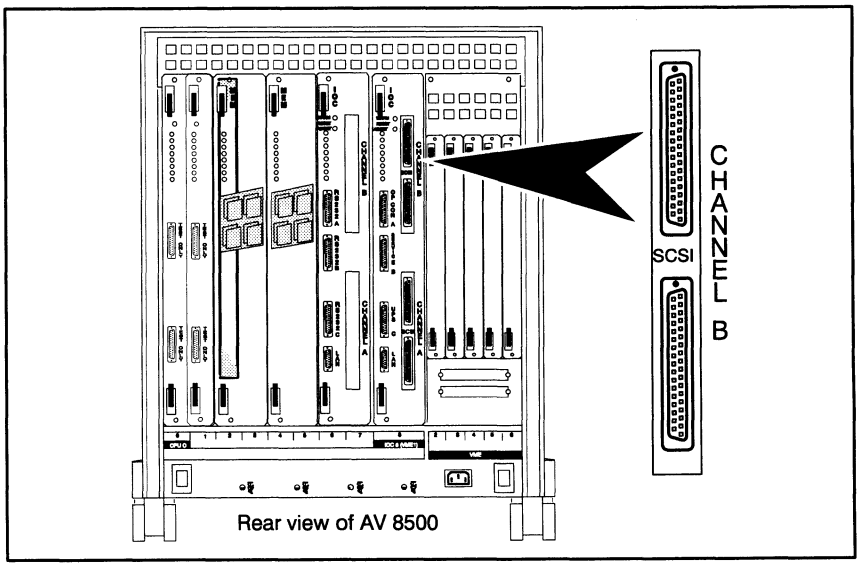
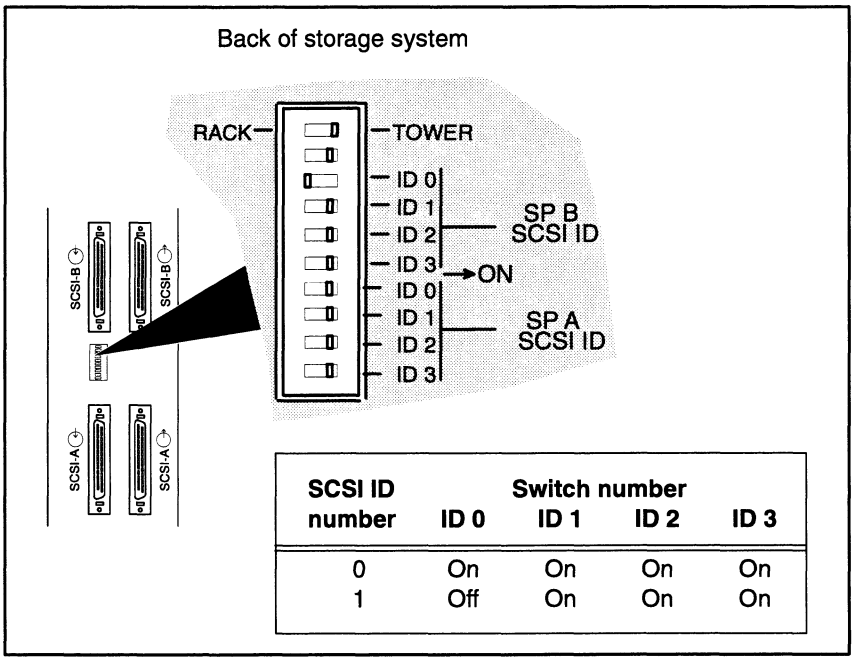


Figure 5–5 SCSI-2 adapter channel B on AV 8500

After plugging in the cabling, the Acme staff had to make sure the SCSI ID numbers for the CLARiiON storage system's SPs are set correctly for their configuration. The SCSI ID for SP A is set to 0, and the ID for SP B is set to 1. Figure 5–6 shows how Acme set the SCSI switch packs on the rear of the CLARiiON system.



SCSI ID number	Switch number ID 0	ID 1	ID 2	ID 3
0	On	On	On	On
1	Off	On	On	On

Figure 5–6 Setting the CLARiiON disk-array's SP Board SCSI IDs for a dual-adapter-with-dual-SP configuration

The CLARiiON disk array's hardware is now set up.

Binding together disk modules into a RAID group

Acme will be using five of the disk modules in the CLARiiON disk array to hold their ORACLE database. To ensure that the database is as available as possible, they decided to bind these modules together in a RAID-5 group.

Acme followed these steps to bind together five disk modules into a RAID-5 group:

1. **Invoke GridMgr.**

   The system displays the GridMgr Main Menu.

```
SP A                        GridMgr Main Menu
SP SCSI ID 0

                    1. Presentation Utility
                    2. Bind Physical Units
                    3. Unbind Physical Units
                    4. View Unsolicited Event Log
                    5. Change Parameters
                    6. View Cache Statistics

               Enter ? or <number>? for HELP.
               Enter Choice:
```

Since Acme is using SP A on the CLARiiON system as the primary route to the physical disks, the RAID group will be set up for this SP. If SP A fails, Acme can use the DG/UX system's multi-path disk I/O to automatically switch control of the disks to SP B.

2. Select menu choice 2, "Bind Physical Units."

The system displays an advisory message.

3. After reading the message, press Enter.

The system displays the Bind Physical Units Menu.

```
SP  A
SP SCSI ID 0                  Bind Physical Units Menu

          1. RAID-5 Group (Individual Access Array)
          2. RAID-3 Group (Parallel Access Array)
          3. RAID-1 Mirrored Pair
          4. RAID-0 Group (Nonredundant Individual Access Array)
          5. RAID-1/0 Group (Mirrored RAID-0 Group)
          6. Individual Disk Unit
          7. Hot Spare

Select a valid number, <number>? for HELP or ^ to go to the MAIN MENU
Enter Choice:
```

4. Select menu choice 1, "Bind a RAID-5 Group."

The system displays the Bind RAID-5 Groups grid.

```
SP A            Bind RAID-5 Group (Independent Access Array)
SP SCSI ID 0


                    ________          ________
                   | UNB **|   A    | UNB **|
                   | UNB **|   B    | UNB **|
                   | UNB **|   C    | UNB **|
                   | UNB **|   D    | UNB **|
                   | UNB **|   E    | UNB **|
                     1                  0

                    ________          ________
                   | UNB **|   A    | UNB **|
                   | UNB **|   B    | UNB **|
                   | UNB **|   C    | UNB **|
                   |_______|   D    | UNB **|
                   |_______|   E    | UNB **|
                     3                  2
```

This grid enables Acme to select the disk modules in the CLARiiON array that they will bind together into the RAID-5 group. To achieve the maximum level of high availability, Acme will bind modules A1, B1, C1, D1, and E1. These five modules are all on separate SCSI internal buses in the CLARiiON storage system, keeping the group from losing more than one disk module to an internal SCSI bus failure.

5. **Use the arrow keys to highlight the modules to be bound in the RAID-5 group. Press Enter to select a module.**

After Acme selects the appropriate modules, the Bind RAID-5 Groups grid looks like the following.

```
SP A            Bind RAID-5 Group (Independent Access Array)
SP SCSI ID 0


                    ________          ________
                   | UNB **|   A    | UNB **|
                   | UNB **|   B    | UNB **|
                   | UNB **|   C    | UNB **|
                   | UNB **|   D    | UNB **|
                   | UNB **|   E    | UNB **|
                     1                  0

                    ________          ________
                   | UNB **|   A    | UNB **|
                   | UNB **|   B    | UNB **|
                   | UNB **|   C    | UNB **|
                   |_______|   D    | UNB **|
                   |_______|   E    | UNB **|
                     3                  2
```

6. **Press "B" to bind the modules together.**

The system displays the Bind Options window.

```
                    Bind Options

Physical unit number (0-1F hex)? [default]

Maximum rebuild time (hours)? [4]
Sectors per stripe element? [128]

Enable caches - None, Read, Write, Both? [None]
Confirm bind options (Y/N)? [N]
```

7. **Accept the defaults for the Bind Options. Enter "Y" at the Confirm prompt.**

   The window closes, and the system displays the following prompt.

   ```
   Bind these disk modules?
   ```

8. **Enter "Y."**

   The GridMgr software changes the UNB on the selected modules to BIN and starts creating the RAID-5 group. Binding the group takes about 30 minutes. At the end of the process, the system changes the BIN on the selected modules to ENA.

Setting up the CLARiiON tape-array's high-availability features

The Acme technical staff set up the following high-availability features on their CLARiiON tape-array storage system:

- Set up dual SCSI-2 adapter channels
- Set up redundant tape group

The following sections describe how to set up these features.

Setting up the SCSI-2 adapter channels

To set up their CLARiiON storage system for maximum high availability, the Acme staff connected their CLARiiON system as shown in Figure 5–7.

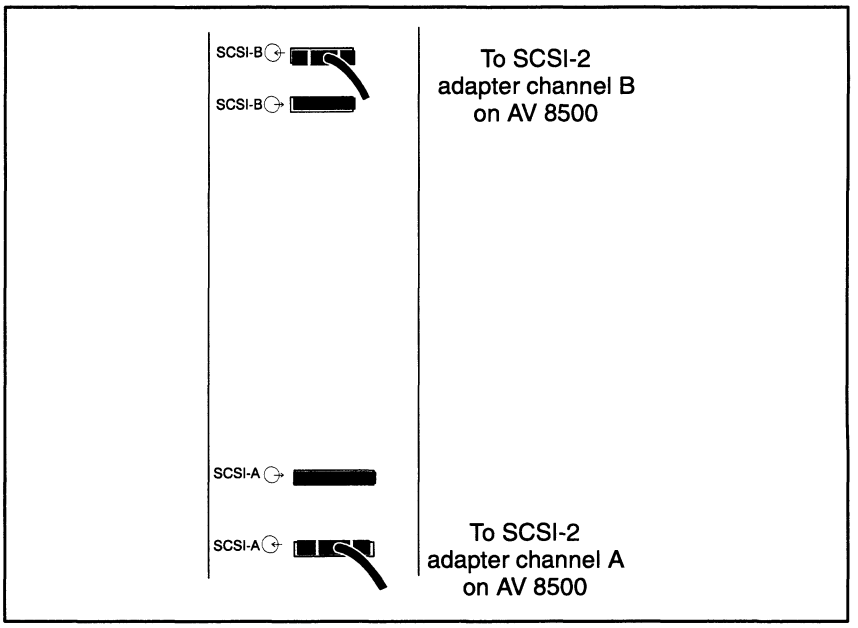


Figure 5–7 Cabling diagram for the extended device configuration

This cabling configuration sets the CLARiiON tape-array up in a dual-adapter-configuration providing the maximum level of high availability. The cables are connected to the back of the AV 8500 in a manner similar to the CLARiiON disk-array.

Setting up the redundant tape group

To set up a redundant tape group, Acme first needs to remove the array's TAP board and make sure switch number two on the board is set to on. Figure 5–8 shows the TAP board's SCSI ID switches.

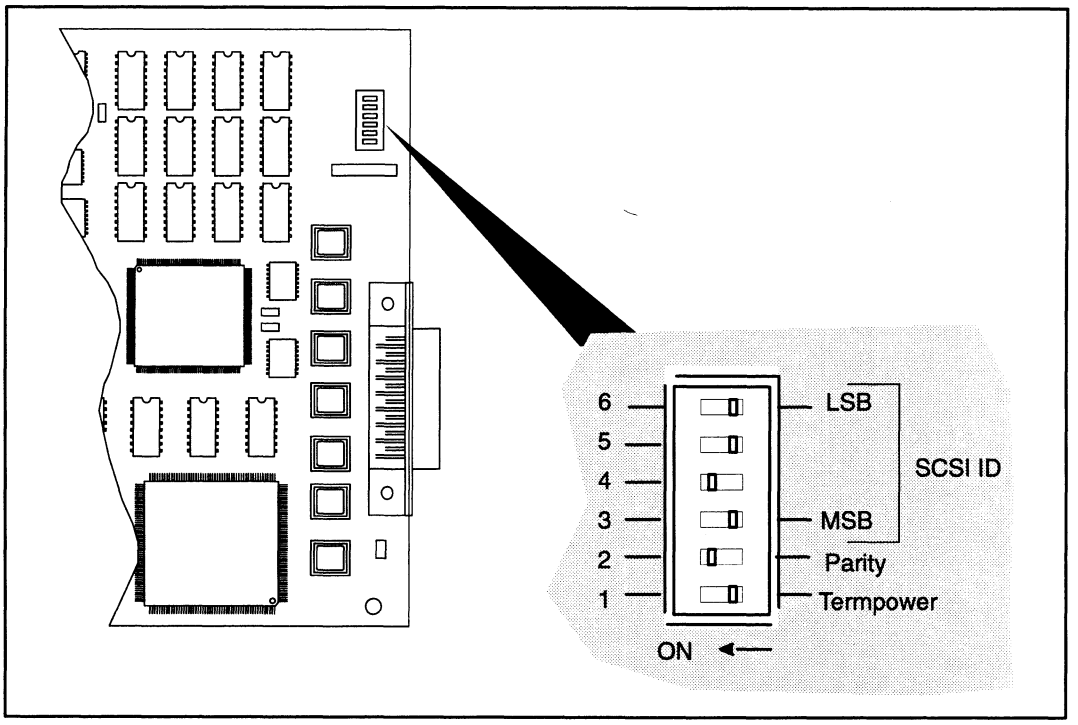


Figure 5–8 Setting the CLARiiON tape-array's TAP board SCSI ID switches

After the TAP board switch is set correctly, Acme must make sure that the device name for the tape group has been built into the DG/UX system kernel. Once this is done, the tape units are automatically bound together into a redundant tape group during the first write operation to the tapes.

Setting up the DG/UX operating system's high-availability features

The Acme technical staff set up the following DG/UX system high-availability features on their system:

- Set up operatorless dumps to disk
- Set up automatic reboot
- Set up watchdog timer
- Set up multi-path LAN I/O
- Set up multi-path disk I/O

The following sections describe how to set up these features.

Note that in addition to these steps, the staff needs to set up the disk and tape drives on their CLARiiON storage systems in the DG/UX system. For a complete description of these procedures, refer to the CLARiiON system documentation.

Setting up dumps to disk

Acme followed these steps to set up operatorless dumps to disk:

1. **Create a virtual disk to hold the system dump.**

   Execute the following operation on **sysadm**:

   Device –> Disk –> Virtual –> Create

   Answer the Create operations queries as follows:

```
New Virtual Disk Name: sys_dump ↵
Striped? [no] ↵
Create File System? [yes] no ↵
Select Space by: [Size alone] disk ↵
Disk to Partition From: sd(ncsc(1),0,4) ↵
Length of Piece in Blocks: (1-576563) 200000 ↵
Starting Block (optional): ↵
Do you want to specify another piece for this virtual disk? [no] ↵
```

   The system creates the **sys_dump** virtual disk on the designated physical disk.

2. **Set the default dump device to the sys_dump virtual disk**

   Execute the following command to set the default dump device to the **sys_dump** virtual disk:

```
# dg_sysctl -f "vdm_dump(sd(ncsc(1),0,4),sys_dump)" ↵
```

3. **Enable automatic system dumps.**

   Execute the following command to make the DG/UX system take an automatic system dump if the system panics:

```
# dg_sysctl -d auto ↵
```

   This completes the set up. If Acme's system fails, the DG/UX system will automatically perform a system dump to the **sys_dump** virtual disk.

Setting up automatic reboot

Execute the following command to set up automatic rebooting:

```
# dg_sysctl -r auto ↵
```

If Acme's system fails, the operating system will automatically reboot.

Setting up the watchdog timer

Acme's new AV 8500 contains a hardware watchdog timer. However, their technical staff does not have to do any set up on either the AViiON or in the DG/UX system for the hardware watch dog timer. The DG/UX system automatically builds the **wdt** device into the kernel. This enables the system to recover from operating system hangs.

However, Acme also wants to use the watchdog timer to protect their system from user level system hangs through the **failovermon** monitoring process. The **failovermon** monitor can communicate with the **wdt** to determine whether there is a hang in a user application. If such a hang exists, the watchdog timer can detect it and panic the system in 90 seconds or less. When the watchdog timer detects the hang, it initiates an automatic system dump and automatic reboot.

Acme followed these steps to set up the user level watchdog timer:

1. **Set up the failovermon process on production.**

   Execute this command to set up the **failovermon** monitoring process:

   ```
   # failovermon -o add -i 30 production ↵
   ```

   This command sets up a **failovermon** process on the server production to check for user level hangs every 30 seconds.

2. **Start the monitoring process.**

   Execute this command to start the **failovermon** monitoring process on production:

   ```
   # failovermon -o start production ↵
   ```

   Acme's system is now protected from hangs at both the operating system and user application levels.

Setting up multi-path LAN I/O

Next, the Acme staff want to protect their system from both LAN interface failures and I/O channel failures. Their AViiON server contains two **dgen** LAN interfaces used to connect to the network. Acme wants to set up their system to automatically switch I/O to the backup LAN interface, if the primary interface fails.

First, Acme needs to be sure to build their DG/UX kernel with both the network device drivers, **dgen(0)** and **dgen(1)**, and the

multi-path LAN driver, **mpl()**. After making sure the kernel contains the correct device drivers, Acme can use **sysadm** to set up multi-path LAN I/O.

Acme executed the following operation in **sysadm** to set up and start multi-path LAN I/O:

Availability –> Multi-Path I/O –> LAN –> Add

Acme answered the Add operation's queries as follows:

```
Primary LAN I/O Path: [dgen0] ↵
Backup LAN I/O Path: dgen1 ↵
Start Multi-Path LAN I/O on Reboot? [yes] ↵
Start Multi-Path LAN I/O Now? [yes] ↵
OK to perform operation? [yes] ↵
```

This operation sets up the multi-path LAN I/O relationship, starts multi-path LAN I/O, and ensures it will be started every time the system boots.

Setting up multi-path disk I/O

Finally, Acme wants to protect their system from any I/O path failures to the RAID-5 group containing their database. To do this they need to set up multi-path disk I/O, which will automatically protect their I/O path from failures to the following components:

- SCSI-2 adapters
- SCSI cables and connectors
- SPs in the CLARiiON disk array

Acme's AV 8500 has two SCSI-2 adapters, each connected to a separate SP in their CLARiiON array. They decide to set up a backup path to the disk group containing their database. This backup path protects from any failure on the path between the first SCSI-2 adapter and SP A in the CLARiiON disk array by using a path to the RAID-5 group through the second SCSI-2 adapter and SP B.

Figure 5–9 shows the different I/O paths Acme will be using for multi-path disk I/O.

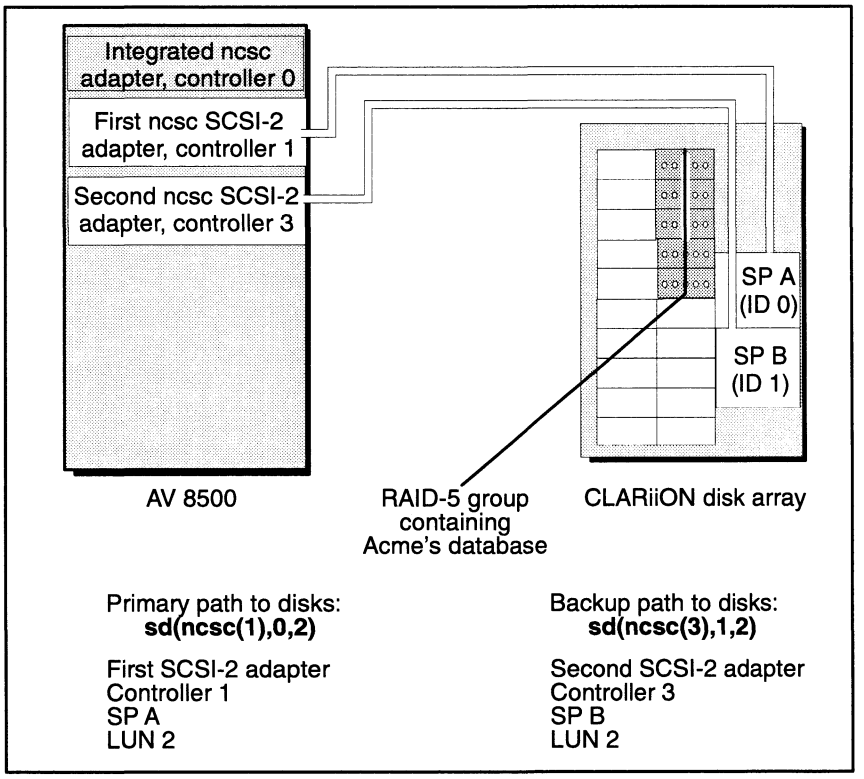


Figure 5–9 Multi-path disk I/O routes

Acme executed the following operation in **sysadm** to set up and start multi-path disk I/O:

Availability –> Multi-Path I/O –> Disk –> Add

Acme answered the Add operation's queries as follows:

```
Primary Disk I/O Path: [sd(ncsc(1),0,0] sd(ncsc(1),0,2) ↵
Backup Disk I/O Path: sd(ncsc(3),1,2) ↵
Backup Disk I/O Path: ↵
Backup Disk I/O Path: ↵
Start Multi-Path Disk I/O on Reboot? [yes] ↵
Start Multi-Path Disk I/O Now? [yes] ↵
OK to perform operation? [yes] ↵
```

This operation sets up the multi-path disk I/O relationships, starts multi-path disk I/O, and ensures it will be started every time the system boots.

Setting up the AV/Alert diagnostic support system

The Acme technical staff performed the following tasks to set up the AV/Alert system on their system:

- Install a dynamic password
- Register and verify the system

The following sections describe how to set up these features.

Installing the password

Before Acme can use the AV/Alert service, they have to install a dynamic password for the system. This password must be obtained from a Data General customer service representative and is changed periodically.

Acme followed these steps to install a dynamic password:

1. **Invoke the SVCMGR software.**

The system displays the following message.

```
Licensed Material - Property of DGC
Data General Proprietary Diagnostics

This diagnostic material contains information which is
proprietary...

(C) DATA GENERAL CORPORATION 1992, 1993
ALL RIGHTS RESERVED
This copyright notice does not constitute or evidence
publication or public disclosure. Press New Line to proceed.
```

2. **Press Enter to clear the message.**

The system displays the SVCMGR Main Menu.

```
Service Manager (Rev 4.0)
=========================

1.       AV/Alert System Status
2.       AV/Alert Setup
3.       AV/Alert Messages
4.       Error Management
5.       Online Diagnostics

Enter selection ('q' to quit; ^ previous menu):
```

3. **Select menu choice 2, "AV/Alert Setup."**

The system displays the AV/Alert Setup Menu.

```
AV/Alert Setup Menu
===================

1.       Dynamic Password
2.       Remote Access Parameters
3.       Modem Parameters

Enter selection ('q' to quit; ^ previous menu):
```

4. **Select menu choice 1, "Dynamic Password."**

   The system displays the Dynamic Password menu.

```
         Dynamic Password

1        Install
2        Verify

Enter selection ('q' to quit; ^ previous menu):
```

5. **Select menu choice 1, "Install."**

   The system displays the AV 8500's Maintenance Access Code and a prompt for the dynamic password.

```
Maintenance Access Code:  ABCDE23523
Enter New Level 1 Password ->
```

6. **Telephone the Data General AV/Alert Support Center.**

   Acme called their Customer Support Center at 1–800–DG–HELPS. They asked the customer support representative for a Level 1, dynamic password and provided their maintenance access code and hardware service contract number. The representative provided the 24-digit password.

7. **Enter the dynamic password.**

   Acme entered the password at the prompt as follows.

```
Maintenance Access Code:  ABCDE23523
Enter New Level 1 Password -> xxxxxxxxxxxxxxxxxxxxxxxx
```

8. **Press Enter.**

   The system displays the Dynamic Password menu again.

9. **Select menu choice 2, "Verify."**

   The system displays the following message verifying the password is valid.

```
FE Service Contract is Valid
Contract ends on 23/05/95

Press any key to continue ...
```

Now that the password is installed, Acme enables remote access to their system, so a Data General support engineer can finish setting up AV/Alert.

Registering and verifying AV/Alert service

Once the AV/Alert software is set up on their system, Acme needs to register a copy of their system configuration with the AV/Alert Support System. This ensures that configuration analysis runs properly for Acme's system if any error occurs.

Acme followed these steps to register and verify the AV/Alert system:

1. **Invoke the SVCMGR Main Menu.**

   The system displays the menu.

```
Service Manager (Rev 4.0)
=========================

1.       AV/Alert System Status
2.       AV/Alert Setup
3.       AV/Alert Messages
4.       Error Management
5.       Online Diagnostics

Enter selection ('q' to quit; ^ previous menu):
```

2. **Select menu choice 3, "AV/Alert Messages."**

   The system displays the AV/Alert Messages Menu.

```
AV/Alert Messages
=================

1.       Test Call
2.       System Hardware Configuration

Enter selection ('q' to quit; ^ previous menu):
```

3. **Select menu choice 1, "Test Call."**

   SVCMGR displays the following message.

```
Warning: A call to the Customer Support Center will be
made.  Do you wish to continue? (Y/N) [N]:
```

4. **Enter "Y."**

   The system sends an MI test packet to the AV/Alert Customer Support Center computer. This confirms that MI calling is working properly for Acme's system.

5. **Select menu choice 2, "System Hardware Configuration."**

   The system displays a list of Acme's system hardware configuration with a prompt asking whether to send the configuration.

6. **Enter "Y" to confirm the configuration.**

   The system sends the hardware configuration to the Customer Support Center.

   AV/Alert is now installed on Acme's system and fully operational.

Putting it all together

After setting up the high-availability features on their AViiON and CLARiiON system components, the Acme staff set up their ORACLE order processing application with its associated data integrity and recovery features. After this was finished, their new Data General system was ready to go to work.

Saving the day

This section presents some examples of how the high-availability features of Acme's new computer system could handle some common system problems.

Disk failure

If one of the disk units in the CLARiiON disk-array storage system fails, order processing would continue without interruption.

In this case, the following events occur:

1. One of the disk units in the RAID-5 group fails.
2. The other four disk units in the RAID group use their parity information to reconstruct the data on the lost disk. The ORACLE application continues without interruption.
3. The AV/Alert system detects that the disk unit has failed and automatically calls the Diagnostic Decision Support (DDS) system at the Customer Support Center.

4. DSS sends a Data General service representative with a replacement disk unit to Acme Novelties.
5. The service representative replaces the failed disk unit at an appropriately scheduled maintenance time. The replacement is performed while the system is up. There is no interruption to order processing.
6. The CLARiiON disk-array automatically rebuilds the data on the replacement drive.

   At no time during this process was Acme's application or data unavailable.

Disk I/O path failure

If one of the components in the I/O path to the disks containing Acme's database fails, order processing would continue without interruption.

In this case, the SCSI-2 adapter fails, and the following events occur:

1. The primary SCSI-2 adapter fails.
2. The DG/UX system recognizes that the adapter has failed and automatically switches over to the backup disk I/O path. Acme's order processing continues without interruption.
3. The AV/Alert system detects that the adapter has failed and automatically calls the Decision Support System (DSS) at the customer support center.
4. DSS sends a Data General service representative with a replacement adapter to Acme Novelties.
5. The service representative takes the system down at an appropriately scheduled maintenance time, replaces the failed adapter, and brings the system back up.
6. When the system comes up, the DG/UX system automatically uses the primary disk I/O path.

   The only time Acme's application or data were unavailable during this process was when the failed controller was replaced. Multi-path disk I/O would also have switched to an alternate path if SP A in the CLARiiON disk array had failed.

LAN failure

If a LAN controller card fails, order processing can again continue without interruption.

The following events would occur:

1. The primary LAN controller card fails.
2. The DG/UX system recognizes that the controller has failed and automatically switches over to the backup controller. Acme's order processing continues without interruption.
3. The AV/Alert system detects that the controller has failed and automatically calls the Decision Support System (DSS) at the customer support center.
4. DSS sends a Data General service representative with a replacement controller to Acme Novelties.
5. The service representative takes the system down at an appropriately scheduled maintenance time, replaces the failed controller, and brings the system back up.
6. When the system comes up, the DG/UX system automatically uses the new primary LAN controller card.

   The only time Acme's application or data were unavailable during this process was when the failed controller was replaced.

Host failure

If Acme's server, production, fails, then order processing would be interrupted until Acme's staff brought the server back up. Having a single host was the major problem area left in Acme's high-availability protection. Chapter 6 tells how the company solved this problem.

End of Chapter

6 Setting up failover configurations

This chapter tells you how to set up systems in a failover configuration. It covers the following areas:

- How to set up failover
- Planning a failover configuration
- Examples of setting up failover systems

How to set up failover

Setting up a failover configuration involves several steps. These steps can be grouped into three major categories:

- Hardware setup
- DG/UX installation and setup
- Failover setup

Figure 6–1 shows the process for setting up two AViiON servers and a CLARiiON disk-array storage system in a new failover configuration.

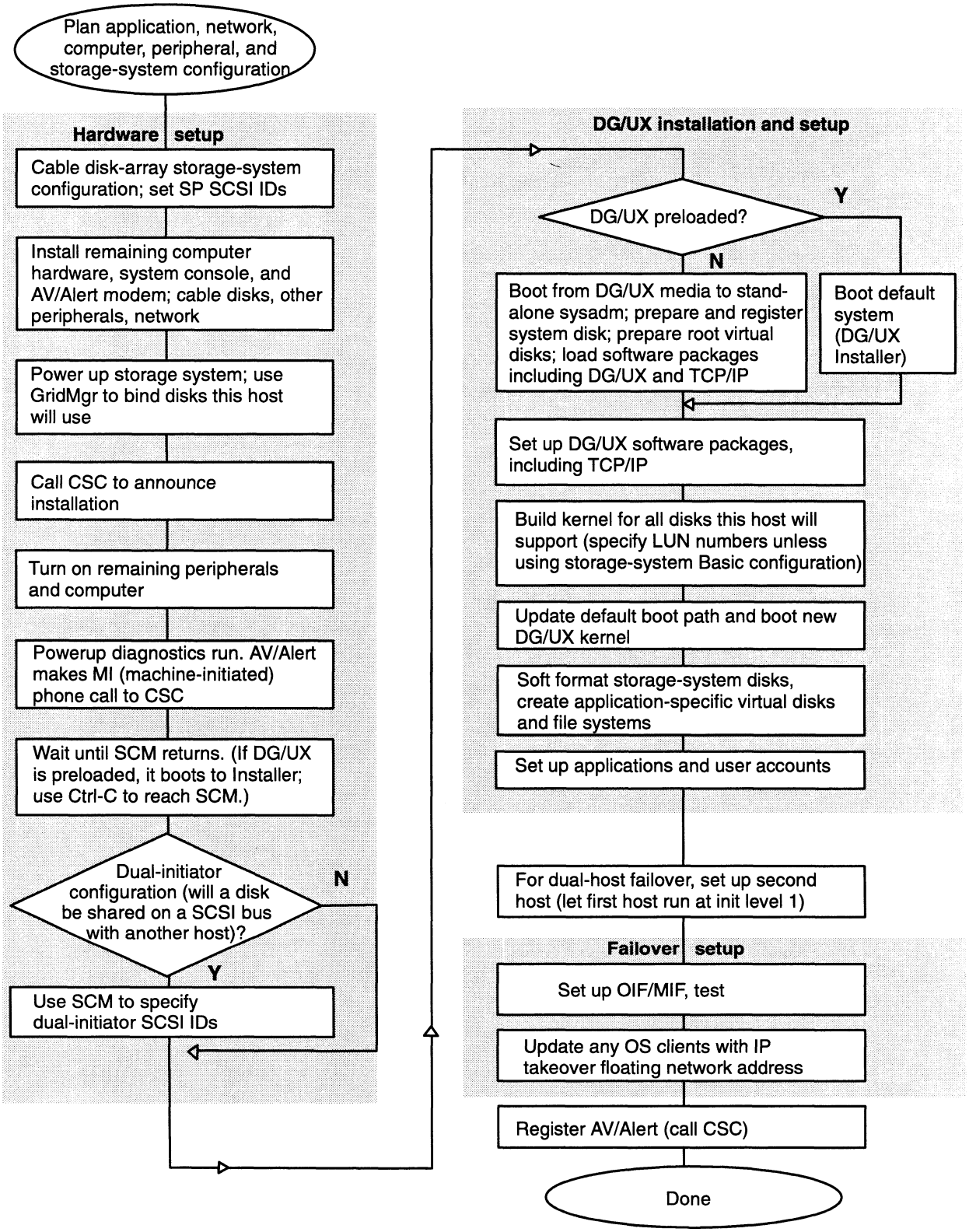


Figure 6–1 Failover set up process

See Appendix B for additional information on upgrading or installing the DG/UX system on servers in an existing failover configuration.

Planning a failover configuration

Before you set up a system in a failover configuration, you should have planned in some detail how you are going to proceed. This section contains the following information to help with that planning:

- Using the high-availability information worksheet
- Special considerations

Using the high-availability information worksheet

To help plan your failover configuration, you can use the worksheet shown in Figure 6–2. You should fill out a different worksheet for each host in a failover configuration. Following the figure are explanations for each of the fields in the worksheet.

High availability information worksheet for one host

Host information: Hostname: (1) _____

Computer type: (2) _____ Number of processors: (3) ____ Main memory: (4) ____ Mbytes

AV/Alert: ☐ (5) Possible downtime/maintenance/repair period: Day: (6) _____ Time: _____ a.m./p.m

DG/UX revision: (7) _____ Number of users: (8) _____

Network controller and software: (9) _____ Network controller and software: _____

Network controller and software: _____ Network controller and software: _____

Applications: (10) _____

Failover type planned: (11) OIF ☐ MIF ☐ IP takeover ☐ Multi-path LAN I/O ☐ Multi-path disk I/O ☐

Multi-path LAN I/O primary path: (12) _____ Backup path: (13) _____

Multi-path disk I/O primary path: (14) _____

Backup paths: (15) _____ _____ _____

Peripheral information:

Number of CLARiiON disk-array storage systems: (16) 20–slot____ 10–slot____

Storage system 1: Storage-system configuration: (17) _____

Physical disk no.: (18) ____ Size: (19) ___Gb Primary route: (20) _____

File system name: (21) _____ Start address: (22) _____ Size: (23) _____

File system name: _____ Start address: _____ Size: _____

File system name: _____ Start address: _____ Size: _____

Other host accessing disk: (24) _____ Failover route: (25) _____

Physical disk no.: _____ Size: ___Gb Primary route: _____

File system name: _____ Start address: _____ Size: _____

File system name: _____ Start address: _____ Size: _____

File system name: _____ Start address: _____ Size: _____

Other host accessing disk: _____ Failover route: _____

Physical disk no.: _____ Size: ___Gb Primary route: _____

File system name: _____ Start address: _____ Size: _____

File system name: _____ Start address: _____ Size: _____

File system name: _____ Start address: _____ Size: _____

Other host accessing disk: _____ Failover route: _____

Number of CLARiiON tape-array storage systems: (26) ____

Figure 6–2 Fields in the high availability information worksheet

(1) For Hostname, enter the name of the host this worksheet describes. If there is another host sharing disks for high availability, use a worksheet for each host.

(2) For Computer type, enter the type of AViiON server; for example, AV 5500

(3) For Number of processors, enter the number of processors in the host; for example, 2.

(4) For Amount of main memory, enter the number of Mbytes in the host's main memory; for example, 64.

(5) For AV/Alert, check the box if the host will have AV/Alert diagnostic support system set up and running.

(6) For Possible downtime/maintenance/repair period, enter any days and times during the week when the host could be removed from service for any needed maintenance or repairs.

(7) For DG/UX revision, enter the revision of the DG/UX system running on the server; for example, 5.4R3.10.

(8) For Number of users, enter the number of users supported by the server; for example, 95.

(9) For Network controller and software, enter the type of network controller and the communications software running on the host; for example dgen0 and TCP/IP. Note that there are four of these fields in case the host has more than one network controller.

(10) For Applications, enter the user applications running on the host; for example, AV Object Office.

(11) For Failover type planned, check the boxes for each type of failover you want to use on this host.

(12) For Multi-path LAN I/O primary path, enter the network controller that is the primary LAN connection for which you have set up

multi-path LAN I/O; for example, dgen0. If you are not using multi-path LAN I/O, leave this field blank.

(13) For Backup paths, enter the network controller for the backup LAN I/O path; for example, dgen1. If you are not using multi-path LAN I/O, leave this field blank.

(14) For Multi-path disk I/O primary path, enter the device specification for the primary path to a disk for which you have set up multi-path disk I/O; for example, sd(ncsc(1,7),2,1). If you are not using multi-path disk I/O, leave this field blank.

(15) For Backup paths, enter the device specification for any backup paths to a disk for which you have set up multi-path disk I/O; for example, sd(ncsc(1,7),3,1). If you are not using multi-path disk I/O, leave this field blank.

(16) For Number of CLARiiON disk-array storage systems, enter the number of 20-slot and 10-slot disk arrays attached to this host.

(17) For Storage-system configuration, enter the configuration in which the host and the CLARiiON disk-array storage system are set up; for example, dual-initiator-dual-bus. See *Configuring and Managing a CLARiiON® Disk-Array Storage System — DG/UX™ Environment* for more information on these configurations.

(18) For Physical disk no., enter the number of the physical disk in the CLARiiON array on which the file system resides; for example, 0. Since your CLARiiON may contain a number of disks that will not be used for failover, you may want to just enter information for failover disks on this worksheet. *Configuring and Managing a CLARiiON® Disk-Array Storage System — DG/UX™ Environment* contains additional worksheets for planning the overall configuration of your disk array.

(19) For Size, enter the capacity of the disk in gigabytes; for example, 2 Gbs.

(20) For Primary route, enter the device specification for the primary route to this disk; for example, sd(ncsc(1,6),0,0).

(21) For File system name, enter the name of the first file system on the disk; for example, root. These is space for up to three file systems

for each disk. If a disk contains more than three file systems, use the file system fields for the disk below the one you are recording and additional worksheets as necessary.

(22) For Start address, enter the address on the disk where the file system starts.

(23) For Size, enter the size of the file system in blocks.

(24) For Other host accessing disk, enter the name of any other host set up in a dual-initiator configuration with this host.

(25) For Failover route, enter the route that any other host set up in a dual-initiator configuration with this host can use to access the disk; for example, sd(ncsc(1,7,0,0).

(26) For number of CLARiiON tape-array storage systems, enter the number of tape arrays attached to this host.

You may need more room than is on the worksheet for areas such as multi-path disk I/O paths and file systems stored on CLARiiON storage systems. In this case, use additional worksheets for the host. Also, use an additional worksheet if more than one CLARiiON disk array is attached to the host.

An example later in this chapter provides more information about using the worksheet. Appendix A contains a blank worksheet you can copy for your high-availability planning.

Special considerations

During the planning for your failover system, there are areas that require special consideration. This section covers some of these areas.

The regain_pulse script

When a system uses MIF to fail services over to a backup host, this can be very disruptive to any users accessing applications or data stored on the failover disks. To avoid a similar disruption when the primary host recovers, you may not want to have the **regain_pulse** script automatically give services back to the primary host. Instead, you can have the **regain_pulse** script do nothing when the backup host detects that the primary host is back up. You can then use OIF

to give services back to the primary host at a more convenient or less disruptive time.

Hosts with automatic recovery features

As covered in Chapter 2, AViiON 8500 and 9500 servers have many built-in high-availability features. These include the ability to automatically reconfigure and reboot the server when some components fail. While there may be some performance degradation, a server missing a component could still perform well enough after rebooting to negate the need for MIF.

If you are setting up a failover configuration containing an AViiON 8500 or 9500 server, you need to account for the automatic recovery features of these hosts in your planning. For example, if the primary host in your failover configuration is an AV/9500, you may want to either just set up OIF or set the time period the backup host waits before initiating MIF to be long enough to give the primary host a chance to recover itself.

This is especially true if your backup host is not as powerful as the primary host. For example, suppose your primary system is an AV 9500 with eight processors and your backup system is an AV 8500 with four processors. If a CPU in the AV 9500 fails, the system will deconfigure the failed processor, automatically reboot, and continue operations with seven processors. In this case, the AV 9500 would perform better in its degraded state than the backup AV 8500.

UPS systems

If you have a UPS system attached to the hosts set up in a failover configuration, you have to take steps to ensure that the DG/UX system takes the backup system down first in the event of a power outage. If the primary system is taken down first, the backup system may initiate the MIF process before it can be brought down. To prevent this from happening, you need to set up some mechanism to ensure the backup system is taken down first in the event of a power outage.

For example, you could use an AViiON workstation to control how the hosts in a failover configuration are brought down during a power outage. The following script shows how you might accomplish this:

```
# Three hosts participate in this UPS shutdown script.  The UPS is
# physically connected to an AV310 (ws1), which is communicating with the
# two "failover" hosts (av1 & av2) via tcp/ip.  The /etc/inittab entry
# for the UPS Daemon was modified to point to this script when shutdown
```

```
# is required by the UPS.  When ws1 detects a required UPS shutdown, it
# notifies all users on ws1 of the problem and that they MUST log off.
# Next, users on av1 are notified of the problem, and finally, users
# on AV2 are notified.  The script sleeps for 30 seconds and the process
# is repeated.  When 1 minute has expired, a signal is sent to av2 to
# shutdown.  The script sleeps for 15 seconds and sends the shutdown
# signal to av1.  Note that av2 (the backup system) must be shut down
# first to prevent the failover monitors from initiating a machine
# initiated failover (MIF).  Another brief sleep and the workstation
# (ws1) shuts down.

rsh ws1 "/usr/sbin/wall << END

***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***  The System's UPS has detected a power loss.  ***WARNING***
***WARNING***          You must LOG OFF NOW!!!              ***WARNING***
***WARNING***  The system will shut down in 1 minute...     ***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***

END"&
rsh av1 "/usr/sbin/wall << END

***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***  The System's UPS has detected a power loss.  ***WARNING***
***WARNING***          You must LOG OFF NOW!!!              ***WARNING***
***WARNING***  The system will shut down in 1 minute...     ***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***

END"&
rsh av2 "/usr/sbin/wall << END

***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***  The System's UPS has detected a power loss.  ***WARNING***
***WARNING***          You must LOG OFF NOW!!!              ***WARNING***
***WARNING***  The system will shut down in 1 minute...     ***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***

.....................................................................
END"&
sleep 30
rsh ws1 "/usr/sbin/wall << END
```

```
***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***  The System's UPS has detected a power loss.    ***WARNING***
***WARNING***          You must LOG OFF NOW!!!                 ***WARNING***
***WARNING***  The system will shut down in 30 seconds...      ***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***          This is the last warning...             ***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***

END"&
rsh av1 "/usr/sbin/wall << END

***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***  The System's UPS has detected a power loss.    ***WARNING***
***WARNING***          You must LOG OFF NOW!!!                 ***WARNING***
***WARNING***  The system will shut down in 30 seconds...      ***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***          This is the last warning...             ***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***

END"&
rsh av2 "/usr/sbin/wall << END
END"&
rsh av2 "/usr/sbin/wall << END

***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***  The System's UPS has detected a power loss.    ***WARNING***
***WARNING***          You must LOG OFF NOW!!!                 ***WARNING***
***WARNING***  The system will shut down in 30 seconds...      ***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***          This is the last warning...             ***WARNING***
***WARNING***                                                  ***WARNING***
***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***WARNING***

END"&
sleep 30
rsh av2 "/sbin/init 5"&
sleep 15
rsh av1 "/sbin/init 5"&
sleep 15
init 5
```

Examples of setting up failover systems

This section contains two examples of how to set up failover systems. The first example continues the example from Chapter 5. The second example does not go into as much detail, but shows how the components of a failover configuration can determine how that configuration is set up.

Example 1

This section presents an example of how to set up and operate Data General's high-availability components in a multiple system environment. It continues the example from Chapter 5, as Acme expands their computer system and sets up failover.

Acme's new computer system had enabled the company to increase its order processing by more than twenty percent. However, the increased order volume had made the company even more reliant on their computer system.

Acme could no longer afford any significant computer downtime. Order volume had grown to the point that, should the computer system go down, the company could no longer process all of the orders they received manually. A computer outage could cause the company to lose customers with a corresponding loss of revenue.

Acme decided to expand the high-availability capabilities of their computer system. They decided to add the following additional components to their system:

- AV 8500 server to use as a development and failover machine
- Machine-Initiated Failover
- IP takeover and NFS failover
- TUXEDO transaction processing monitor

These new features would give Acme the capability to survive any failure of their primary production server. Should that server fail, the backup server could quickly and completely replace the primary server, enabling order processing to resume after a short delay.

Figure 6–3 shows how Acme's system will look after these new components have been set up.

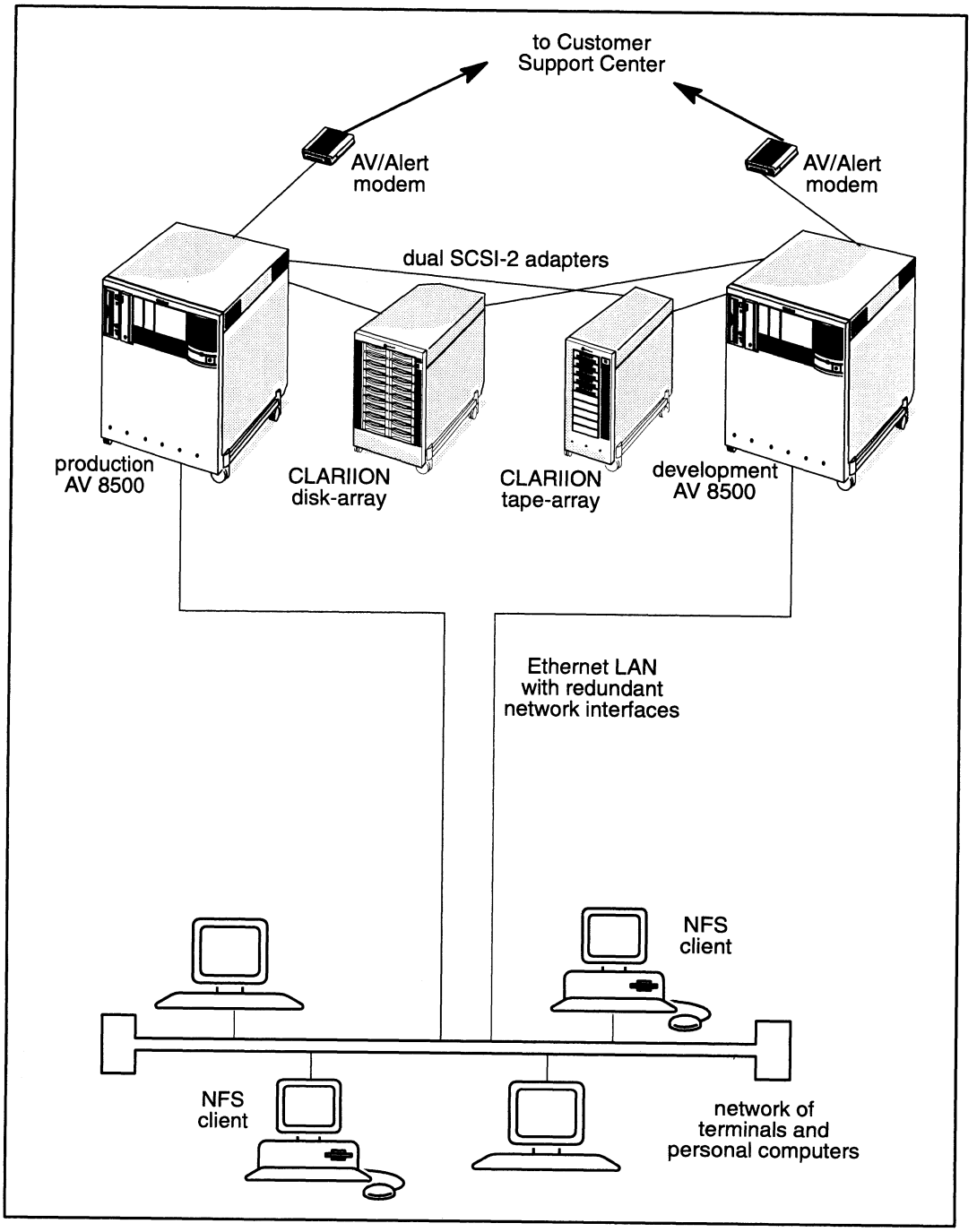


Figure 6–3 Acme Product's expanded computer system for example 1

Planning Acme's failover configuration

As part of the planning process for Acme's expanded system, their technical staff filled out a high availability information worksheet for each host. Figure 6–4 shows the partially completed worksheet for the host production.

High availability information worksheet for one host

Host information: Hostname: production

Computer type: av/8500 Number of processors: 2 Main memory: 64 Mbytes

AV/Alert: ☒ Possible downtime/maintenance/repair period: Day: _____ Time: _____ a.m./p.m

DG/UX revision: 5.4R3.10 Number of users: 95

Network controller and software: dgen0, TCP/IP Network controller and software: dgen1, TCP/IP

Network controller and software: _____ Network controller and software: _____

Applications: acme, AV obj office

Failover type planned: OIF☒ MIF☒ IP takeover☒ Multi-path LAN I/O☒ Multi-path disk I/O ☐

Multi-path LAN I/O primary path: dgen0 Backup path: dgen1

Multi-path disk I/O primary path: sd(ncsc(1,6),0,2)

Backup paths: sd(ncsc(3,6),1,2) _____

Peripheral information:

Number of CLARiiON disk-array storage systems: 20–slot 1 10–slot ____

Storage system 1: Storage-system configuration: Dual-initiator-dual-bus

Physical disk no.: 0 Size: 2 Gb Primary route: sd(ncsc(1,6),0,0)

File system name: root Start address: 700 Size: 50000

File system name: usr Start address: 57000 Size: 300000

File system name: _____ Start address: _____ Size: _____

Other host accessing disk: _____ Failover route: _____

Physical disk no.: 1 Size: 2 Gb Primary route: sd(ncsc(1,6),0,1)

File system name: acme Start address: 700 Size: 150000

File system name: _____ Start address: _____ Size: _____

File system name: _____ Start address: _____ Size: _____

Other host accessing disk: development Failover route: sd(ncsc(1,7),0,1)

Physical disk no.: 2 Size: 8 Gb Primary route: sd(ncsc(1,6),0,2)

File system name: acme_data Start address: 700 Size: 5000000

File system name: _____ Start address: _____ Size: _____

File system name: _____ Start address: _____ Size: _____

Other host accessing disk: development Failover route: sd(ncsc(1,7),0,2)

Number of CLARiiON tape-array storage systems: 1

Figure 6–4 Sample worksheet

The Acme technical staff filled out a similar worksheet for the host development.

Setting up the hardware

After completing the planning for their system, the Acme technical staff needs to integrate the new AViiON server into their system. In addition to the hardware setup described in this section, Acme needs to set up development in their Local Area Network.

Acme set up the following high-availability system components:

- Set up the AViiON servers and CLARiiON disk-array storage system in a dual-initiator-dual-bus configuration
- Set up CLARiiON tape-array storage system in a dual-bus cabinet-sharing configuration

The following sections describe how to set up these features.

Setting up the failover configuration — To set up their CLARiiON disk array in a failover configuration with both of their AViiON servers, the Acme staff connected their CLARiiON storage system as shown in Figure 6–5.

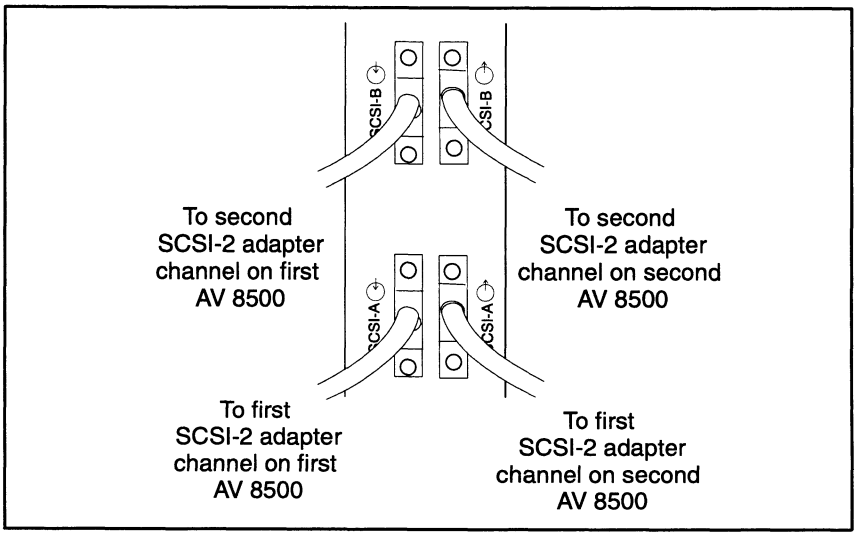


Figure 6–5 CLARiiON disk-array SCSI-2 cabling for a dual-initiator-dual-bus configuration

This cabling configuration sets the CLARiiON disk array up in a dual-initiator-dual-bus configuration providing the maximum level of high availability when the disk array is connected to two hosts.

Once the CLARiiON array is connected to the AViiON servers, Acme needs to change the SCSI ID on the CLARiiON storage system's SP board. For the new configuration, they must set the SCSI ID on both SPs to 0. Figure 6–6 shows how Acme set the SCSI ID switches.

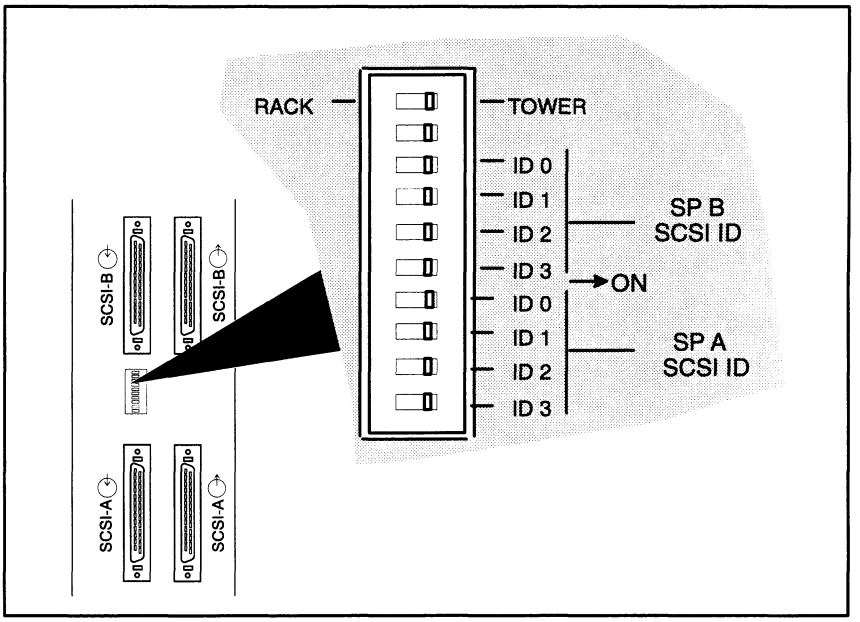


Figure 6–6 Setting the CLARiiON disk-array's SP Board SCSI IDs for a dual-initiator-dual-bus configuration

The CLARiiON disk-array hardware is now set up.

Setting up the CLARiiON tape-array storage system — To set up their CLARiiON tape array with both AViiON servers, the Acme staff connected the array as shown in Figure 6–7.

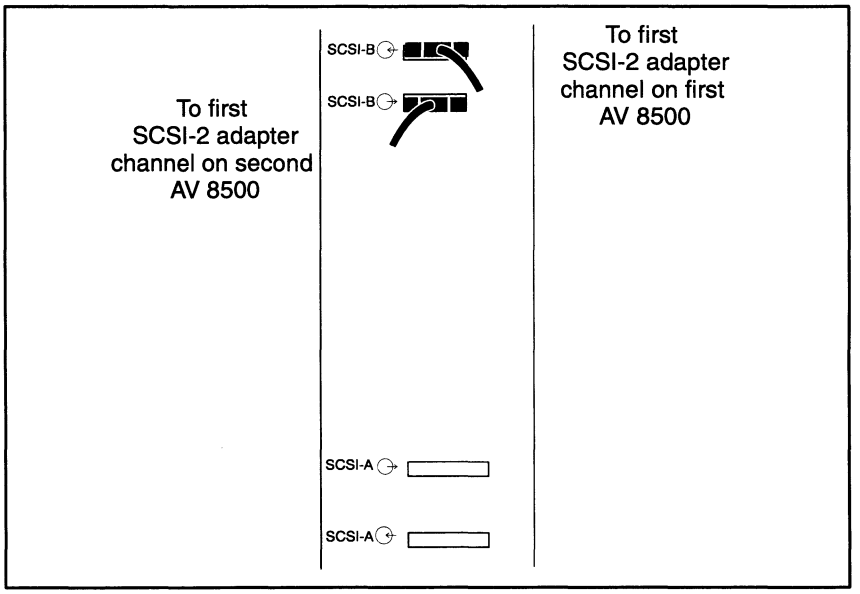


Figure 6–7 CLARiiON tape-array SCSI-2 cabling for dual-bus cabinet-sharing configuration

This cabling configuration sets the CLARiiON tape array up in a single-bus cabinet-sharing configuration. The CLARiiON tape-array hardware is now set up.

Specifying the dual-initiator SCSI IDs

After setting up their hardware, Acme needs to set the SCM on both hosts to specify the dual-initiator SCSI IDs. Since the hosts will be sharing a SCSI bus, each server must have a unique SCSI-ID. The primary device in a dual-initiator configuration uses 6 for the SCSI adapter ID. The backup device uses 7 for the SCSI adapter ID.

Acme followed this procedure on the host production:

1. **Enter "f" at the System Control Monitor (SCM) prompt.**

   The system displays the following menu:

```
View or Change System Configuration

1  Change default boot paths
2  Setup dual-initiator SCSI IDs
3  Modify port parameters
4  View system configuration
5  Modify system parameters
6  Return to previous screen

Enter choice->
```

2. **Select choice 2 from the View or Change System Configuration menu**

   The system displays the following screen:

```
1                          11
2                          12
3                          13
4                          14
5                          15
6                          16
7                          17
8                          18
9                          19
10                         20

Enter choice->
```

3. **Select choice 1.**

   The system displays the following prompt:

```
Type controller specification ->
```

4. **Enter the dual-initiator SCSI ID for production:**

```
Type controller specification -> ncsc(1,6)
```

The system displays the following screen:

```
1  ncsc(1,6)              11
2                         12
3                         13
4                         14
5                         15
6                         16
7                         17
8                         18
9                         19
10                        20

Enter choice->
```

This sets up the dual-initiator SCSI ID for production. Acme repeats this procedure on development, entering `ncsc(1,7)` for that host's ID.

Setting up the expanded DG/UX system high-availability features

The Acme technical staff set up the following expanded DG/UX system high-availability features on their system:

- Operator Initiated Failover (OIF)
- Machine Initiated Failover (MIF)
- IP takeover
- NFS failover

The following sections describe how to set up these features.

In addition to these steps, the staff needs to set up the disks in the CLARiiON disk array that will be controlled by their new AViiON computer in the DG/UX system. They similarly need to set up the CLARiiON tape array on the new server. For a complete description of these procedures, refer to the CLARiiON storage system documentation.

Setting up Operator Initiated Failover — Before Acme can set up their system for MIF, they have to set up OIF. Acme must perform the following tasks to set up OIF:

- Edit the system file on both hosts to define the failover disks.
- Write application script.

- Set up and synchronize the OIF databases.

The following sections describe how Acme performed these tasks.

Editing the system file — Before Acme can set up OIF, they must edit the system file to define the the disks they want to fail over.

In Acme's case, they will be failing over the RAID-5 group that contains their database, and the disk that contain their application. These disks are usually controlled by Acme's primary server, production. However, if production should fail, development will take over these disks and restart the order processing application. The backup server, development, also controls some disks in the CLARiiON disk array, but these will never be needed for failover.

Acme followed these steps to set up the failover disks in the system files:

1. **Build a new kernel on production.**

   Execute the following **sysadm** operation on production:

   System –> Kernel –> Build

   Accept the default for all prompts. When **vi** runs, look for the device entries for production's system disk, the RAID-5 group, and the application disk. These entries look like the following:

```
sd(ncsc(1,6),0,0)
sd(ncsc(1,6),0,1)
sd(ncsc(1,6),0,2)
sd(ncsc(1,6),0,3)
```

   Edit these entries to remove devices controlled by development. In this case, `sd(ncsc(1,6),0,3)` is the system disk for development and is removed from production's system file.

   After editing the file, the device entries looked like the following:

```
sd(ncsc(1,6),0,0)
sd(ncsc(1,6),0,1)
sd(ncsc(1,6),0,2)
```

   Save the file and exit from **vi**. Build the kernel and link it with **/dgux**.

   Note that since Acme had already set the SCSI ID for production at the SCM, they did not need to leave the ID in the system file entries. However, if they had not been able to set the SCSI at the SCM, Acme would have had to explicitly set it in the system file as shown above.

2. **Bring production down to the SCM prompt.**

3. **Build a new kernel on development.**

   Repeat the tasks in step 1 for production. However, for the backup server remove the devices controlled by the primary server.

   After editing production's file, the device entries looked like the following:

   ```
   sd(ncsc(1,7),0,3)
   ```

   Save the file, then build and link a new kernel.

4. **Bring development down to the SCM prompt.**

5. **Reboot production.**

6. **Reboot development.**

   After completing these steps, Acme's two AViiON servers are set up in the failover configuration.

   Writing the application script — Before setting up the servers for OIF, Acme needs to write an application script for the failover process. This script will automatically restart Acme's order processing application as part of the OIF process.

   The script must start with an interpreter line such as the following:

   ```
   #!/bin/sh
   ```

   Acme writes a script in the file **/usr/opt/acme/acme_up** on the primary server, production.

   Setting up OIF — After setting up the disks and writing the application script, Acme is ready to set up OIF. Each disk that Acme wants to fail over from the primary server to the backup server must be set up in the **/etc/failover** databases on the servers. In this case, the disks containing the file systems with Acme's application and database will be set up for failover.

   Acme followed these steps to set up OIF:

1. **Add the disk containing Acme's database to the OIF databases.**

   Execute the following **sysadm** operation on production:

   Availability –> Disk Failover –> Giveaway –> Add

Answer the Add operation's queries as follows:

```
Local Physical Disk: sd(ncsc(1,6),0,1) ↵
Remote Physical Disk: [sd(ncsc(1,6),0,1)] sd(ncsc(1,7),0,1) ↵
Hostname to Fail Disk over to: development ↵
Synchronize database? [no] ↵
Application Command Line: ↵
OK to perform operation? [yes] ↵
```

The system adds an entry for the database disk to the **giveaway** database on production.

2. **Add the disk containing Acme's application to the OIF databases.**

   Execute the Add operation again. Answer the queries as follows:

```
Local Physical Disk: sd(ncsc(1,6),0,2) ↵
Remote Physical Disk: [sd(ncsc(1,6),0,2)] sd(ncsc(1,7),0,2) ↵
Hostname to Fail Disk over to: development ↵
Synchronize database? [no] yes ↵
Application Command Line: /usr/opt/acme/acme_up ↵
OK to perform operation? [yes] ↵
```

   This entry is added to the **giveaway** database on production. Since this is the last disk Acme wants to add to the database, they also chose to synchronize the database. This sets up the **takeaway** database on development with this information. Acme also added the path to the **acme_up** application script, so their applications would be restarted when OIF executes.

   OIF is now set up on Acme's system.

   Setting up Machine Initiated Failover — After setting up OIF, Acme is ready to set up MIF. Once MIF is in place, Acme's system will be able to automatically recover if their primary server, development, fails.

   Acme must perform the following tasks to set up MIF:

- Set up the failover scripts and a script to stop their application gracefully
- Set up the communication paths
- Add and start the failover monitor

   The following sections describe how Acme performed these tasks.

   Setting up the scripts — Acme needs to set up the following scripts for MIF:

- **/etc/failover/failovermon_lost_pulse**
- **/etc/failover/failovermon_regain_pulse**
- **/usr/opt/acme/acme_down**

The **failovermon_lost_pulse** defines the actions development should take when it can no longer contact production over either of the defined communication paths. The script typically initiates an OIF Take with Trespass operation to failover the disks. The script could also restart Acme's order processing application, but this is not necessary as the **acme_up** script in OIF already does this.

The **failovermon_lost_pulse** on development looks like the following:

```
#!/bin/sh
#
# failovermon_lost_pulse shell script
#
# This script uses a take with trespass to move the disks used
# for the Acme order processing application to the system
development.
# This script is used when failovermon(1M) has detected that host
# production has failed.
#
DISKLIST="sd(ncsc(1,7),0,1) sd(ncsc(1,7),0,2)"
#
admpdisk -o list -r "$DISKLIST" > dev/null 2>&1
if [ $? -ne 0 ]
then admfailoverdisk -o take -T -h production "$DISKLIST"
else
echo "Disks "$DISKLIST" are already registered." > /dev/console
fi
# end of script
```

The **failovermon_regain_pulse** defines the actions development should take when it regains communication with production. This script uses the new **acme_down** script to gracefully shutdown Acme's order processing application, then gives control of the failover disks back to production.

The **failovermon_regain_pulse** on development looks like the following:

```
#!/bin/sh
#
# failovermon_regain_pulse shell script
#
# This script is run when the host development detects that host
# production has come back on line. It takes down the Acme software
# and returns the physical disks for the Acme order processing
# application back to host production. We sleep at the beginning to
# ensure that production has finished running its /etc/rc3.d script
# for failover.
#
DISKLIST="sd(ncsc(1,7),0,1) sd(ncsc(1,7),0,2)"
sleep 30
/usr/opt/acme/acme_down
admfailoverdisk -o give -h production "$DISKLIST"
# end of script
```

Acme writes **/usr/opt/acme/acme_down** script to gracefully shut down their order processing application, so the failover disks can be given back to the primary server. As with the **acme_up** script, the script must start with an interpreter line such as `#!/bin/sh`.

Setting up the communications paths — To set up MIF, Acme must first set up the alternate communications paths for the two servers. Unlike the OIF setup, these operations are performed on the backup server, development. The TCP/IP paths from development to production are defined in the **/etc/hosts** database. The entries in this database associate the names "production" and "production-alt" with their Internet Protocol (IP) addresses. The name production-alt refers to a secondary network interface on production that development can use as an alternate path to communicate with that host.

Execute the following **sysadm** operation on development to set up the communications paths:

Availability –> Disk Failover –> Alternate Paths –> Add

Answer the Add operation's queries as follows:

```
Primary Host Name: production ↵
Alternate Remote Host Name: production-alt ↵
Check Path? [no] yes ↵
OK to perform operation? [yes] ↵
Failover altcommpath entry for production has been added.
Alternate path production-alt to production is accessible.
```

The alternate communications paths for MIF are now set up.

Adding the failover monitor — The final step Acme takes to set up MIF is adding and starting the **failovermon** monitoring process. This process runs on development. It attempt to communicate with production over the defined communications paths at a regularly defined interval. If the monitor loses contact with production over all paths and cannot reestablish communications after a defined number of retries, it execute the **failovermon_lost_pulse**.

Execute the following **sysadm** operation on development to add and start the failover monitor:

Availability –> Disk Failover –> Monitors –> Add

Answer the Add operation's queries as follows:

```
Host to Monitor: production ↵
Seconds Between Heartbeats: [0] 15 ↵
Number of Retries: [0] 1 ↵
Lost Pulse Action: [/etc/failover/failovermon_lost_pulse] ↵
Regain Pulse Action: [/etc/failover/failovermon_regain_pulse] ↵
Start Monitor On Reboot? [no] yes
Start Monitor Now? [no] yes ↵
OK to perform operation? [yes] ↵
Failover monitors file entry for production has been added.
Failover monitor for production has been started.
```

MIF is now set up and running on Acme's system.

Setting up IP takeover — To ensure that development can take over additional functions of production, Acme also sets up IP takeover on their system. With IP takeover, Acme assigns an IP address that will float with the disks in the CLARiiON disk array that are set up for failover. If production fails, this floating IP address can be transferred from production to development. This enables Acme's employees to use login and NFS services of those disks through the backup server.

Acme must perform the following tasks to set up IP takeover:

- Set up the floating IP address
- Set up and start IP takeover
- Add IP takeover command lines to the failover scripts
- Set up NFS failover

The following sections describe how Acme performed these tasks.

Setting up the floating IP address — Before setting up IP takeover, Acme must assign an IP address that will float with the failover disks in their CLARiiON disk array. This address must be defined in the **/etc/hosts** databases on both production and development. Acme uses the name "giveip."

Setting up and starting IP takeover — After setting up the floating IP address on both of their servers, Acme is ready to set up IP takeover. Since production has primary control of the failover disks, IP takeover is set up on that server. Acme will be using the **dgen0** network interface on each server for IP takeover.

Execute the following **sysadm** operation on production to set up IP takeover:

Availability –> IP Takeover –> Add

Answer the Add operation's queries as follows:

```
Host Name: [development] ↵
Floating Address Name: giveip ↵
Local Network Interface: dgen0 ↵
Remote Network Interface: dgen0 ↵
Start Floating IP Address on System Reboot? [yes] ↵
Start Floating IP Address Now? [no] yes ↵
Synchronize database? [no] yes ↵
OK to perform operation? [yes] ↵
```

This entry is added to the **/etc/failover/failoverip** database on production. Since Acme synchronized the databases, the entry is also added on development.

IP takeover is now set up and running on Acme's system.

Adding IP takeover command lines to the failover scripts — To make IP takeover part of MIF, Acme needs to add a command line to each of the failover scripts on development.

Acme added the following command line to the end of the **failovermon_lost_pulse** script on development:

```
admfailoverip -o take -h production -T
```

Acme added the following command line to the end of the **failovermon_regain_pulse** script on development:

```
admfailoverip -o give -h production
```

These command lines make transfer of the floating IP address on Acme's system part of the MIF process.

Setting up NFS failover — After setting up MIF and IP takeover, Acme is ready to set up NFS failover. There are no **sysadm** operations or administrative commands associated with NFS failover. After MIF and IP takeover are set up, NFS failover is also set up and running as a by-product of these operations. However, Acme does need to do two things to complete the set up of NFS failover.

First, Acme needs to make sure that all of the file systems used in the NFS failover process are exported.

Second, each NFS client in Acme's network needs to add the **giveip** file system as a remote file system using a **sysadm** sequence such as the following:

File System –> Remote Filesys –> Add

```
Mount Directory: /usr/opt/acme_orders↵
Remote Host Name: giveip↵
Remote Mount Directory: /usr/opt/orders_fs↵
Write Permission: [Read/Write]↵
NFS Mount Type: [Hard] ↵
Interruptible? [yes] ↵
Retry in background? [yes]↵
Mount the file system? [yes]↵
OK to perform operation? [yes]↵
```

Adding the remote file system on each NFS client mounts the file system and enters its name in file **/etc/fstab**, from which the DG/UX system will mount it automatically on future client startups.

Since all of Acme's users access their application and database through NFS, it now no longer matters whether these file systems are controlled by production or development. Users can access the file systems through the **giveip** regardless of which host controls the disk containing the file system. The only thing users might notice is an error message from NFS while failover is taking place, but they will have access to the file systems again once that process is complete.

Setting up TUXEDO

To ensure that user requests are directed to the backup AViiON server when the primary server fails, Acme also purchased the TUXEDO transaction processing monitor. After redesigning their application to take advantage of TUXEDO's capabilities, Acme had a complete and seamless failover system.

For a complete description of how to set up this product, refer to the TUXEDO documentation.

Saving the day

This section presents an example of how the high-availability features of Acme's expanded computer system could quickly recover if the primary host fails.

Host failure — If Acme's primary server fails, the backup server takes over the role of the failed server.

In this case, the following events would occur:

1. The primary server, production, fails.
2. The **failovermon** monitoring process on the backup server, development, loses contact with production.
3. The **failovermon** process tries once to reestablish communications with production on all of the defined communications paths.
4. When **failovermon** cannot reestablish communication with production, it executes the **/etc/failover/failovermon_lost_pulse** script.
5. The script initiates the OIF takeover of the disks in the CLARiiON disk array controlled by production. The script also initiates IP takeover to transfer the floating IP address of these disks to development. As part of this process, development begins providing NFS service to the NFS clients accessing **giveip** file systems.
6. The OIF process executes the **/usr/opt/acme/acme_up** script.
7. The **/usr/opt/acme/acme_up** script restarts Acme's order processing application. The TUXEDO transaction processing monitor restores transactions to the state they were in when production failed.
8. Once Acme's application has restarted, Acme's employees can continue order processing on development. Those employees using **telnet** or **rlogin** to access the server must re-establish their connections, but continue to call the same host, **giveip**.
9. Acme's technical staff determines what caused production to fail, corrects the problem, and reboots the server.
10. The **failovermon** process on development reestablishes communications with production and executes the **/etc/failover/failovermon_regain_pulse** script.
11. The script executes the **/usr/opt/acme/acme_down** script to gracefully shut down the order processing application. It then transfers control of the failover disks, floating IP address, and NFS mount points back to production.
12. The OIF process executes the **/usr/opt/acme/acme_up** script on production.
13. The **/usr/opt/acme/acme_up** script restarts Acme's order processing application on production. The TUXEDO transaction processing monitor restores transactions to the state they were in when the disks were transferred back to production.

14. Order processing proceeds as before.

    Acme's system administrators did not have to do anything to start the failover process. The DG/UX system's failover operations, in conjunction with TUXEDO, automatically, completely, and seamlessly transferred Acme's order processing to their backup server. Acme's applications and data were only unavailable for a short time.

Example 2

This section presents an example of how the components that make up a company's computer system can help determine what sort of failover features are included in the system. This example does not go into the level of detail of the previous one. Instead, it presents an overview of the situation and the final system configuration. This example considers how Acme Products might have set up their computer system differently if the company had bought different AViiON models from Data General.

Suppose that when Acme replaced its old, proprietary computer system, they had opted for an AV 5500 server instead of an AV 8500 server. In this case, installation and set up of the new computer system would still proceed similar to the process described in Chapter 5.

Now suppose that when Acme decided to expand their computer system, they decided to add an AV 9500 to their existing AV 5500 and CLARiiON storage systems. How would this change the set up their expanded system? The following sections provide details about these differences.

Installation and set up

Acme would not install or set up their new AV 9500 themselves. Instead, Data General engineers would perform these tasks for the new server. Data General engineers would also maintain the new server.

Also, the roles of the two servers would be different. Due to its processing power, the new AV 9500 would become the primary server for the entire company. The AV 5500 it replaces could be reserved exclusively for application development and testing by the programming staff and similar functions.

Failover set up

Given the many high-availability features built into the AV 9500 and the associated CLARiiON disk array, as well as DG/UX

features such as multi-path disk I/O, Acme would not need to set up MIF for their system. In fact, there would be little reason to set up any sort of failover from the AV 9500 to the AV 5500.

However, Acme could gain by setting up failover from the AV 5500 to the AV 9500. The AV 5500 does not have the recovery features of the AV 9500, so the failure of a component such as a CPU would cause that host to fail until the component is replaced. Given this, Acme could set up OIF to give their system administrators the capability to transfer some of the functions of the AV 5500 to the AV 9500 when necessary.

Since the AV 9500 is the main production machine, Acme probably would not want to set up MIF. The sudden transfer of the AV 5500's processing needs to the AV 9500 might cause an unacceptable slow down in order processing if done at the wrong time. With OIF set up, the system administrators could check the load on the AV 9500 and see if it could handle the additional processing needs of the AV 5500's users.

CLARiiON disk-array storage system configuration

Since failover is not a major part of Acme's high-availability needs in this case, they could set up their CLARiiON disk-array in a cabinet-sharing configuration instead of a dual-initiator-dual-bus configuration. This would let their two hosts share the disk array without the added overhead of a more complicated configuration.

See *Configuring and Managing a CLARiiON® Disk-Array Storage System — DG/UX™ Environment* for more information on these configurations.

The expanded system

Figure 6–8 shows part of Acme's expanded computer system for this example.

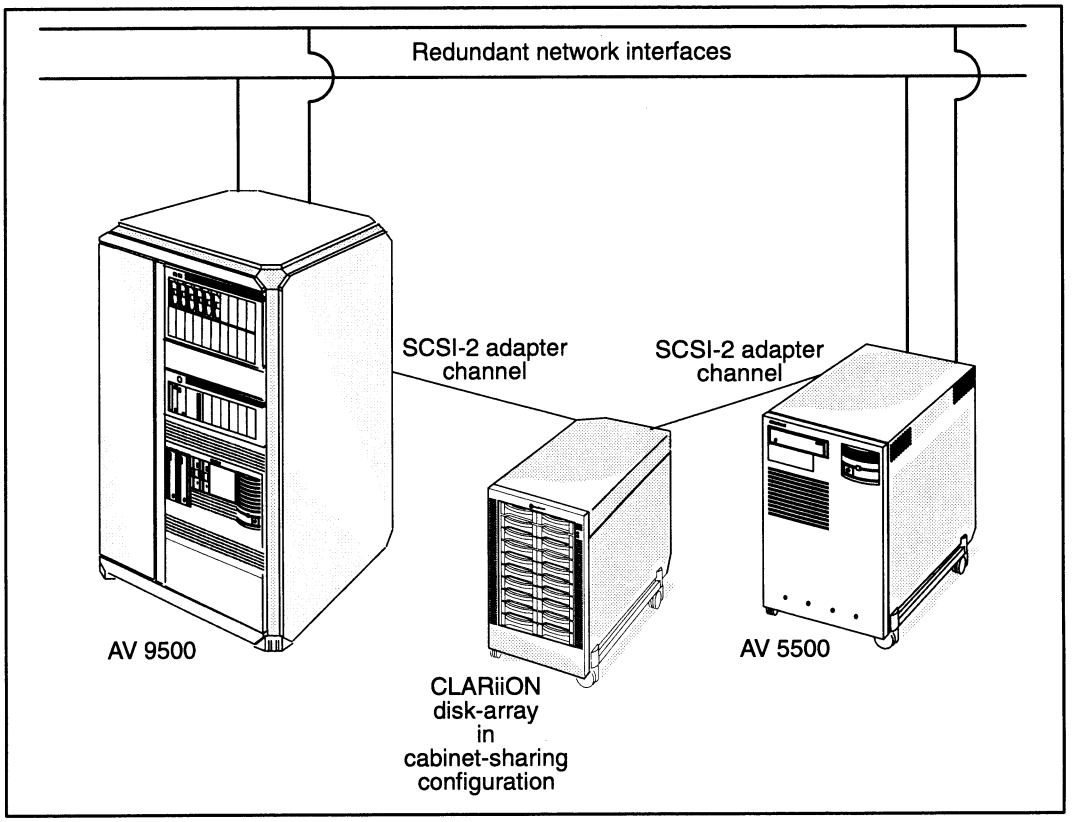


Figure 6–8 Acme Product's expanded computer system for example 2

End of Chapter

A High-availability worksheet

This appendix contains a copy of the worksheet shown in Chapter 6.

If you need additional sheets, please copy the worksheet and use the copies. A sample completed copy of the worksheet is shown in Chapter 6.

High availability information worksheet for one host

Host information: Hostname: ______________

Computer type: __________ Number of processors: _____ Main memory: _____ Mbytes

AV/Alert: ☐ Possible downtime/maintenance/repair period: Day: _______ Time: _______ a.m./p.m

DG/UX revision: _____________ Number of users: ________

Network controller and software: ________________ Network controller and software: _______________

Network controller and software: ________________ Network controller and software: _______________

Applications: ____

Failover type planned: OIF ☐ MIF ☐ IP takeover ☐ Multi-path LAN I/O ☐ Multi-path disk I/O ☐

Multi-path LAN I/O primary path: ______________ Backup path: __________________

Multi-path disk I/O primary path: ______________

Backup paths: __________________ __________________ __________________

Peripheral information:

Number of CLARiiON disk-array storage systems: 20–slot____ 10–slot____

Storage system 1: Storage-system configuration: _______________________

Physical disk no.: _____ Size: ___Gb Primary route: __________________

File system name: ______________ Start address: __________ Size: __________

File system name: ______________ Start address: __________ Size: __________

File system name: ______________ Start address: __________ Size: __________

Other host accessing disk: ______________ Failover route: __________________

Physical disk no.: _____ Size: ___Gb Primary route: __________________

File system name: ______________ Start address: __________ Size: __________

File system name: ______________ Start address: __________ Size: __________

File system name: ______________ Start address: __________ Size: __________

Other host accessing disk: ______________ Failover route: __________________

Physical disk no.: _____ Size: ___Gb Primary route: __________________

File system name: ______________ Start address: __________ Size: __________

File system name: ______________ Start address: __________ Size: __________

File system name: ______________ Start address: __________ Size: __________

Other host accessing disk: ______________ Failover route: __________________

Number of CLARiiON tape-array storage systems: ____

End of Appendix

B Installing the DG/UX system on hosts in a dual-initiator configuration

Since the hosts in a failover configuration are set up in a dual-initiator configuration, you need to take some special preparations before installing or upgrading the DG/UX system on those hosts. The following sections provide details about these preparations.

Note that in the instructions that follow, the host on which you boot or install the DG/UX system is referred to as the "local" host. The other host in the local host's dual-initiator configuration is referred to as the "remote" host.

Setting SCSI bus operating parameters

Before booting the DG/UX installer program, set SCSI bus operating parameters on the local host using a firmware System Control Monitor (SCM) menu. For instructions about this procedure, refer to the 014-series operating manual for your AViiON hardware model. You can do this procedure only from the firmware SCM prompt that displays when no software is running on your computer:

```
SCM>
```

Check the SCSI bus parameters every time you install or upgrade the DG/UX software. Note that you must set the parameters before booting a DG/UX installer kernel or autoconfigured DG/UX custom kernel. If for some reason you cannot set the parameters first, you may shut down the remote host or disconnect the shared SCSI bus from the local host before booting such a kernel.

IMPORTANT: Do not boot a DG/UX installer kernel or autoconfigured DG/UX custom kernel in an active failover configuration before setting the SCSI bus operating parameters. Unset or improper parameters will panic the DG/UX system on the remote host.

Upgrading an existing system

While the SCSI bus operating parameters should have been set before the DG/UX system was first booted on the local host, you should still check the SCSI bus operating parameters to make sure they are correct. Again, refer to the 014–series operating manual for your AViiON hardware model.

Booting a preloaded system

If the operating system on your AViiON computer you is preloaded (the DG/UX system software was loaded onto your system disk at the factory), you must interrupt the boot process to reach the SCM menus before the DG/UX installer kernel starts. To do this, you can enter Ctrl–C any time after the AViiON powerup diagnostics have completed and the DG/UX bootstrap is running.

The powerup diagnostics are complete when the system displays the following self-test information:

```
Testing...
    0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
Passed
```

If your server does not complete the self test or displays error messages, do not attempt to complete the installation. Refer to your hardware manual for troubleshooting information.

Once the powerup diagnostics are complete, the system displays the following DG/UX bootstrap message:

```
DG/UX System Release 5.4R3.10 Bootstrap
Loading image ..............................
```

You may enter Ctrl–C any time after the DG/UX bootstrap displays the first line of the message. However, you must interrupt the bootstrap before the system completes this operation. If the system displays the following message, the bootstrap is complete and you can no longer interrupt the system boot:

```
Configuring devices ...
```

Once you are at the SCM prompt, you can check your system's SCSI bus operating parameters.

IMPORTANT: If you fail to interrupt the bootstrap before it completes, turn off your computer immediately. Since the DG/UX system does not begin writing to disk devices until after it has finished configuring all devices, you will not corrupt any data on your disks. Wait a few seconds until your computer has reset, then try again.

Failing disks over to the remote host

If you are upgrading the DG/UX system on a host in an existing failover configuration, make sure the remote host takes ownership of the dual–initiator disks before you start. This minimizes the downtime of the failover configuration as a whole.

You may skip this step if any of the following conditions exist:

- you do not use failover,
- you are adding a new system to an existing failover configuration,
- or you are certain the local host does not own any dual-initiator disks.

To check whether the local host owns a dual-initiator disk, use the **sysadm** operation Availability –> Disk Failover –> Giveaway –> List. see whether **sysadm** displays a message similar to the following:

```
No failover giveaway entries found
```

If **sysadm** display this message, the local system does not own any dual-initiator disks. Proceed to the next step.

If **sysadm** does list failover **giveaway** entries, the local host owns one or more dual-initiator disks. Transfer these disks to the remote host through the **sysadm** operation Availability –> Disk Failover –> Give. See Chapter 4 for detailed information about giving failover disks to another host.

What next?

After you complete your DG/UX installation or upgrade, follow the instructions in Chapter 4 to set up failover on the local system or return ownership of the failover disk to the local system.

End of Appendix

C Sharing tape drives

Tape sharing involves one or more tape drives accessible by two different routes (paths). Generally, the routes run to controllers in two hosts but sometimes they run to different controllers within the same host. Tape sharing is primarily useful because it lets you use tape drives more efficiently; both systems can use one tape drive. A shared tape can be on a shared SCSI bus in a dual-initiator configuration.

Figure C–1 shows two hosts with a tape-array storage system in a dual-initiator configuration.

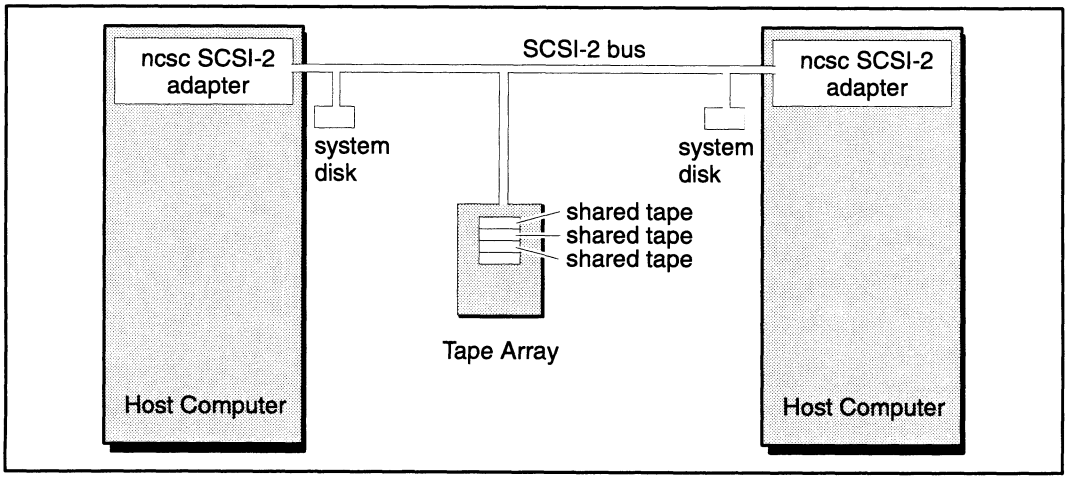


Figure C–1 Two hosts with a tape array in a dual-initiator configuration

Setting up the shared SCSI bus

This section tells how to set up two computers to share a SCSI bus in a dual-initiator configuration for sharing tapes in a peripherals unit like a tape array. This configuration works only on Data General AViiON systems that support SCSI-2 adapters (such as **ncsc**). For both hosts to use a shared bus, each must specify a different SCSI address for its SCSI-2 adapter channel. For example, one host uses tape unit 2 though the tape unit name **st(ncsc(1,7),2,0)** and the other host uses tape unit 2 through the tape unit name **st(ncsc(1,6),2,0)**. The 7 and 6 in the unit names are the SCSI adapter SCSI IDs; since they are different, they allow each host to initiate its own requests to the bus.

The basic requirement for a shared SCSI bus configuration is that every device on the bus must have a different SCSI ID. This means the SCSI-2 adapters themselves must have different SCSI IDs.

CAUTION: *If two or more devices on the SCSI bus have the same SCSI ID, either or both systems may hang or behave erratically. Be sure that each device on the bus, including each SCSI adapter, has a unique SCSI ID.*

To set up two systems to share a SCSI bus, follow these steps:

1. For each device that will be on the shared SCSI bus (other than the SCSI adapters themselves), set the jumpers to indicate a unique SCSI ID. Do not set any devices to SCSI ID 6 or 7; reserve these numbers for the SCSI adapters later.

   For example, you want to set up hosts **system1** and **system2** to share a SCSI bus. The SCSI bus will support a combined storage subsystem with a system disk for each host and three tape drives, as shown earlier in Figure C–1. Each device on the bus will have a unique SCSI ID, as follows.

Device	SCSI ID
1.4 Gbyte disk (system disk)	0
1.4 Gbyte disk (system disk)	1
QIC-320 tape (not shared)	2
QIC-320 tape (to be shared)	4
QIC-320 tape (to be shared)	5

2. Install the hardware: connect the devices to either of the two systems in the standard fashion, such that you can boot and set up both systems independently. Do not connect the two systems' SCSI buses together at this time.

   In the example configuration, you connect the systems so that **system1** has one disk and two tape drives, and **system2** has one disk drive and one tape drive, as follows.

system1	system2
`sd(ncsc(),0,0)`	`sd(ncsc(),1,0)`
`st(ncsc(),2,0)`	`st(ncsc(),5,0)`
`st(ncsc(),4,0)`	

3. Select one of the systems to set up first. Power up and boot the system. If necessary, install the DG/UX system software. This is the system whose SCSI-2 adapter will have the SCSI ID of 6. You set the adapter's SCSI ID at the SCM as discussed below.

4. Execute the **sysadm** sequence `System -> Kernel -> Build` to build a new kernel. For every tape drive you want the hosts to share, make sure the device entry appears in both host's system files.

   Make sure the entries in the system file include the correct SCSI ID for the device as well as the SCSI ID for the local SCSI adapter, SCSI ID 6. Build and install the kernel.

   In the example configuration, boot **system1** and execute System -> Kernel -> Build. When **vi** runs, you see the following entries in the system file:

```
sd(ncsc(),0,0)
st(ncsc(),2,0)
st(ncsc(),4,0)
```

   You want let the hosts share the tape drive attached to **system2**, so you add an entry for it to **system1**'s system file:

```
st(ncsc(),5,0)
```

   Then add the SCSI adapter address and SCSI ID in all of these entries. The SCSI-2 adapter number is 1 (the 0 SCSI-2 adapter number is used by the host's internal SCSI bus), and its SCSI ID is 6. The list in the system file now looks like this:

IMPORTANT: If your system does not enable you to set the dual-initiator SCSI ID from the SCM, you must explicitly set it in the system file.

```
sd(ncsc(1,6),0)
st(ncsc(1,6),2)
st(ncsc(1,6),4)
st(ncsc(1,6),5)
```

5. After making the changes you want to the kernel configuration, exit from the text editor (for **vi**, enter Esc, then ZZ).

6. Proceed with the Build operation and install the new kernel. If an error occurs, **sysadm** will display an error message. For more information on building a kernel, see Chapter 4.

7. Shut down the first system. From the SCM, set the system's dual-initiator SCSI ID to **ncsc(1,6)**. Be sure that you set the SCSI ID to the same one you used in the system file. Also, change the SCM's default boot path so that it includes the correct SCSI adapter SCSI ID. For example, if the default boot path had been

sd(ncsc(),0), change it to **sd(ncsc(1,6),0)**. If you also intend to set your system's boot path using **dg_sysctl**(1M) after you have brought the system back up, remember to specify the correct SCSI adapter SCSI ID there as well.

CAUTION: Be careful to specify the correct SCSI adapter SCSI ID in the boot path. If the boot path you use to boot your system specifies the SCSI ID of the SCSI adapter installed in the other system, the boot will fail and the SCSI bus will hang. If the SCSI bus hangs, an attempt by either system to access the SCSI bus will hang the system. When this happens, recover by resetting the hardware and rebooting.

8. Power off the system.
9. Power on and boot the second system. If necessary, install the DG/UX system software. This system's SCSI adapter will have SCSI ID 7.
10. Execute `System -> Kernel -> Build` and perform essentially the same steps that you performed for the first system: add device entries for the tape drives connected to the other system that you want to share; then in each device entry set the SCSI ID for the SCSI adapter. On this system, the SCSI adapter SCSI ID is 7.

IMPORTANT: If your system does not enable you to set the dual-initiator SCSI ID from the SCM, you must explicitly set it in the system file.

For example, you execute `System -> Kernel -> Build`. The system file initially contains these entries:

```
sd(ncsc(),1)
st(ncsc(),5)
```

You want to share the tape drive connected to **system1**, so you add the entry for it in your system file:

```
st(ncsc(),4)
```

Then add the SCSI adapter address and SCSI ID in all of these entries. The SCSI adapter address is 0, and the SCSI ID is 7. The list in the system file now looks like this:

```
sd(ncsc(1,7),1)
st(ncsc(1,7),5)
st(ncsc(1,7),4)
```

11. After making the changes you want to the kernel configuration, exit from the text editor (for **vi**, enter Esc, then ZZ).
12. Proceed with the Build operation and install the new kernel. If an error occurs, **sysadm** will display an error message. For more information on building a kernel, see *Managing the DG/UX™ System*.

13. Shut down the second system. From the SCM, set the system's dual-initiator SCSI ID to **ncsc(1,7)**. Also, change the SCM's default boot path so that it includes the correct SCSI adapter SCSI ID. For example, if the default boot path had been **sd(ncsc(),1)**, change it to **sd(ncsc(1,7),1)**. If you also intend to set your system's boot path using **dg_sysctl**(1M) after you have brought the system back up, remember to specify the correct SCSI adapter SCSI ID there as well.

CAUTION: Be careful to specify the correct SCSI adapter SCSI ID in the boot path. If the boot path you use to boot your system specifies the SCSI ID of the SCSI adapter installed in the other system, the boot will fail and the SCSI bus will hang. If the SCSI bus hangs, an attempt by either system to access the SCSI bus will hang the system. When this happens, recover by resetting the hardware and rebooting.

14. Power off the system.

15. With both machines powered off, reconnect the hardware in the desired configuration: using normal SCSI cables, connect the two systems and the devices as a single SCSI bus. Unlike a typical SCSI bus, this bus has no external terminators on it; the adapters in the systems at each end terminate the bus. Refer to your hardware documentation before performing this step.

16. Power on and boot the first system with its new kernel.

17. When the first system has completed booting, power on and boot the second system with its new kernel.

Table C–1 shows the initial configuration of the example systems.

Table C–1 Sample system configuration for tape sharing

Device	SCSI ID Number	Name Relative to system 1	Name Relative to system 2
SCSI adapter on **system 1**	6	**ncsc(1,6)**	Not accessible
SCSI adapter on **system 2**	7	Not accessible	**ncsc(1,7)**
1.4 Gbyte disk (system disk)	0	**sd(ncsc(1,6),0)**	Not accessible
1.4 Gbyte disk (system disk)	1	Not accessible	**sd(ncsc(1,7),1)**
QIC 320 tape drive (shared)	2	**st(ncsc(1,6),2)**	Not accessible
QIC 320 tape drive (shared)	4	**st(ncsc(1,6),4)**	**st(ncsc(1,7),4)**
QIC 320 tape drive (shared)	5	**st(ncsc(1,6),5)**	**st(ncsc(1,7),5)**

At this point, users on the two systems may share both tape drives.

Operating a shared tape drive

Sharing a tape drive is essentially the same in a dual-initiator configuration as on a single system: the tape drive is available for anyone until someone opens it, at which time it remains occupied until the user closes it. The commands that you typically use to open a tape drive include **cpio**, **tar**, and **dump2**. Any attempt to open an already-opened shared tape drive results in the same error that you might see when two users on the same system try to use an unshared tape drive at the same time. Thus users take turns in sharing a tape drive on the shared bus the same way they take turns when sharing a normal one.

If either system boots while the other is using a tape device on the shared SCSI bus, I/O operations to the tape may fail. If a user or application on the running system is using a shared tape device when the other system boots, one of two things will occur: the user or application using the tape drive will receive an I/O error; or the booting system will be unable to configure the tape device. To avoid these problems, make sure no shared tape drives are in use when booting either system.

End of Appendix

Index

Within the index, a bold page number indicates a primary reference. A range of page numbers indicates that the reference spans those pages.

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

TIPS ORDERING PROCEDURES

TO ORDER

1. An order can be placed with the TIPS group in two ways:
   a. MAIL ORDER – Use the order form on the opposite page and fill in all requested information. Be sure to include shipping charges and local sales tax. If applicable, write in your tax exempt number in the space provided on the order form.

   b. Send your order form with payment to:

   Data General Corporation
   ATTN: Educational Services/TIPS G155
   4400 Computer Drive
   Westboro, MA 01581–9973

   c. TELEPHONE – Call TIPS at (508) 870–1600 for all orders that will be charged by credit card or paid for by purchase orders over $50.00. Operators are available from 8:30 AM to 5:00 PM EST.

METHOD OF PAYMENT

2. As a customer, you have several payment options:
   a. Purchase Order – Minimum of $50. If ordering by mail, a hard copy of the purchase order must accompany order.

   b. Check or Money Order – Make payable to Data General Corporation. Credit Card – A minimum order of $20 is required for MasterCard or Visa orders.

SHIPPING

3. To determine the charge for UPS shipping and handling, check the total quantity of units in your order and refer to the following chart:

Total Quantity	Shipping & Handling Charge
1–4 Items	$5.00
5–10 Items	$8.00
11–40 Items	$10.00
41–200 Items	$30.00
Over 200 Items	$100.00

If overnight or second day shipment is desired, this information should be indicated on the order form. A separate charge will be determined at time of shipment and added to your bill.

VOLUME DISCOUNTS

4. The TIPS discount schedule is based upon the total value of the order.

Order Amount	Discount
$0–$149.99	0%
$150–$499.99	10%
Over $500	20%

TERMS AND CONDITIONS

5. Read the TIPS terms and conditions on the reverse side of the order form carefully. These must be adhered to at all times.

DELIVERY

6. Allow at least two weeks for delivery.

RETURNS

7. Items ordered through the TIPS catalog may not be returned for credit.
8. Order discrepancies must be reported within 15 days of shipment date. Contact your TIPS Administrator at (508) 870–1600 to notify the TIPS department of any problems.

INTERNATIONAL ORDERS

9. Customers outside of the United States must obtain documentation from their local Data General Subsidiary or Representative. Any TIPS orders received by Data General U.S. Headquarters will be forwarded to the appropriate DG Subsidiary or Representative for processing.

TIPS ORDER FORM

Mail To: Data General Corporation
Attn: Educational Services/TIPS G155
4400 Computer Drive
Westboro, MA 01581 - 9973

BILL TO:	**SHIP TO:** (No P.O. Boxes - Complete Only If Different Address)
COMPANY NAME _____	COMPANY NAME _____
ATTN: _____	ATTN: _____
ADDRESS _____	ADDRESS (NO PO BOXES) _____
CITY _____	CITY _____
STATE _____ ZIP _____	STATE _____ ZIP _____

Priority Code _____ (See label on back of catalog)

Authorized Signature of Buyer (Agrees to terms & conditions on reverse side)	Title	Date	Phone (Area Code)	Ext.

ORDER #	QTY	DESCRIPTION	UNIT PRICE	TOTAL PRICE

A SHIPPING & HANDLING

☐ UPS	ADD
1–4 Items	$5.00
5–10 Items	$8.00
11–40 Items	$10.00
41–200 Items	$30.00
200+ Items	$100.00

Check for faster delivery

Additional charge to be determined at time of shipment and added to your bill.

☐ UPS Blue Label (2 day shipping)
☐ Red Label (overnight shipping)

B VOLUME DISCOUNTS

Order Amount	Save
$0–$149.99	0%
$150–$499.99	10%
Over $500.00	20%

Tax Exempt # or Sales Tax (if applicable) _____

ORDER TOTAL	
Less Discount See B	–
SUB TOTAL	
Your local* sales tax	+
Shipping and handling – See A	+
TOTAL – See C	

C PAYMENT METHOD

☐ Purchase Order Attached ($50 minimum)
P.O. number is _____. (Include hardcopy P.O.)
☐ Check or Money Order Enclosed
☐ Visa ☐ MasterCard ($20 minimum on credit cards)

Account Number ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Expiration Date ☐☐☐☐

_____
Authorized Signature
(Credit card orders without signature and expiration date cannot be processed.)

THANK YOU FOR YOUR ORDER

PRICES SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.
PLEASE ALLOW 2 WEEKS FOR DELIVERY.
NO REFUNDS NO RETURNS.

* Data General is required by law to collect applicable sales or use tax on all purchases shipped to states where DG maintains a place of business, which covers all 50 states. Please include your local taxes when determining the total value of your order. If you are uncertain about the correct tax amount, please call 508–870–1600.

Form 702
Rev. 8/87

DATA GENERAL CORPORATION
TECHNICAL INFORMATION AND PUBLICATIONS SERVICE

TERMS AND CONDITIONS

Data General Corporation ("DGC") provides its Technical Information and Publications Service (TIPS) solely in accordance with the following terms and conditions and more specifically to the Customer signing the Educational Services TIPS Order Form. These terms and conditions apply to all orders, telephone, telex, or mail. By accepting these products the Customer accepts and agrees to be bound by these terms and conditions.

1. CUSTOMER CERTIFICATION
Customer hereby certifies that it is the owner or lessee of the DGC equipment and/or licensee/sub–licensee of the software which is the subject matter of the publication(s) ordered hereunder.

2. TAXES
Customer shall be responsible for all taxes, including taxes paid or payable by DGC for products or services supplied under this Agreement, exclusive of taxes based on DGC's net income, unless Customer provides written proof of exemption.

3. DATA AND PROPRIETARY RIGHTS
Portions of the publications and materials supplied under this Agreement are proprietary and will be so marked. Customer shall abide by such markings. DGC retains for itself exclusively all proprietary rights (including manufacturing rights) in and to all designs, engineering details and other data pertaining to the products described in such publication. Licensed software materials are provided pursuant to the terms and conditions of the Program License Agreement (PLA) between the Customer and DGC and such PLA is made a part of and incorporated into this Agreement by reference. A copyright notice on any data by itself does not constitute or evidence a publication or public disclosure.

4. LIMITED MEDIA WARRANTY
DGC warrants the CLI Macros media, provided by DGC to the Customer under this Agreement, against physical defects for a period of ninety (90) days from the date of shipment by DGC. DGC will replace defective media at no charge to you, provided it is returned postage prepaid to DGC within the ninety (90) day warranty period. This shall be your exclusive remedy and DGC's sole obligation and liability for defective media. This limited media warranty does not apply if the media has been damaged by accident, abuse or misuse.

5. DISCLAIMER OF WARRANTY
EXCEPT FOR THE LIMITED MEDIA WARRANTY NOTED ABOVE, DGC MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE ON ANY OF THE PUBLICATIONS, CLI MACROS OR MATERIALS SUPPLIED HEREUNDER.

6. LIMITATION OF LIABILITY
A. CUSTOMER AGREES THAT DGC'S LIABILITY, IF ANY, FOR DAMAGES, INCLUDING BUT NOT LIMITED TO LIABILITY ARISING OUT OF CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT OR WARRANTY SHALL NOT EXCEED THE CHARGES PAID BY CUSTOMER FOR THE PARTICULAR PUBLICATION OR CLI MACRO INVOLVED. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO CLAIMS FOR PERSONAL INJURY CAUSED SOLELY BY DGC'S NEGLIGENCE. OTHER THAN THE CHARGES REFERENCED HEREIN, IN NO EVENT SHALL DGC BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING BUT NOT LIMITED TO LOST PROFITS AND DAMAGES RESULTING FROM LOSS OF USE, OR LOST DATA, OR DELIVERY DELAYS, EVEN IF DGC HAS BEEN ADVISED, KNEW OR SHOULD HAVE KNOWN OF THE POSSIBILITY THEREOF; OR FOR ANY CLAIM BY ANY THIRD PARTY.

B. ANY ACTION AGAINST DGC MUST BE COMMENCED WITHIN ONE (1) YEAR AFTER THE CAUSE OF ACTION ACCRUES.

7. GENERAL
A valid contract binding upon DGC will come into being only at the time of DGC's acceptance of the referenced Educational Services Order Form. Such contract is governed by the laws of the Commonwealth of Massachusetts, excluding its conflict of law rules. Such contract is not assignable. These terms and conditions constitute the entire agreement between the parties with respect to the subject matter hereof and supersedes all prior oral or written communications, agreements and understandings. These terms and conditions shall prevail notwithstanding any different, conflicting or additional terms and conditions which may appear on any order submitted by Customer. DGC hereby rejects all such different, conflicting, or additional terms.

8. IMPORTANT NOTICE REGARDING AOS/VS INTERNALS SERIES (ORDER #1865 & #1875)
Customer understands that information and material presented in the AOS/VS Internals Series documents may be specific to a particular revision of the product. Consequently user programs or systems based on this information and material may be revision–locked and may not function properly with prior or future revisions of the product. Therefore, Data General makes no representations as to the utility of this information and material beyond the current revision level which is the subject of the manual. Any use thereof by you or your company is at your own risk. Data General disclaims any liability arising from any such use and I and my company (Customer) hold Data General completely harmless therefrom.

Achieving High
Availability on
AViiON® Systems

093–701133–01

Cut here and insert in binder spine pocket