Preventing and Fixing Problems on the DG/UX® System

069–701145–02

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.02, August 1999
Ordering No. 069–701145

Notice

Preventing and Fixing Problems on the DG/UX® System
069–701145–02

Revision History:	Effective with:
Original Release – March 1998	DG/UX System Release 4.20
First Revision – March 1999	DG/UX System Release 4.20MU04
Second Revision – August 1999	DG/UX System Release 4.20MU05

This manual replaces 093–701141. A vertical bar in the margin of a page indicates substantive technical change from the previous revision.

Preface

This manual presents tips and techniques for preventing and fixing typical problems you may have on your DG/UX® system. It summarizes methods for collecting information about your system that will aid you in the troubleshooting process or that you may furnish to the Data General Support Center for their continued investigation. This manual also covers the typical error alert mechanisms — error messages, log files, system hangs, and halts — and diagnostic tools — process monitoring commands, disk space maintenance commands, and the file system checker (**fsck**).

This manual does not cover comprehensively all types of problems that you can experience on the DG/UX system. It does not cover third-party applications, networks, DG/UX system installation, and details about particular types of hardware, such as modems and printers. Consult the appropriate manual listed below for detailed troubleshooting information.

The audience for this manual is the system administrator who should have some system administration training or knowledge. Such a person should be familiar with the hardware and software configuration and with managing users on the system. The reader of this manual should know how to use the UNIX® shell and the DG/UX system administration utilities (**sysadm**).

DG/UX manuals containing troubleshooting information

The following DG/UX manuals contain troubleshooting information.

Configuring and Managing a CLARiiON® Disk-Array Storage System — DG/UX® Environment — 014–002323

Installing and Managing Printers on the DG/UX® System — 069–701143

Installing, Operating, and Maintaining the CLARiiON™ Tape–Array Storage System Model 7931— DG/UX® or AOS/VS II Environment — 014–002359–00

Installing the DG/UX® System — 069–701140

Legato Networker's Administrator's Guide — 069-100495

Legato Networker's User's Guide — 069-100496

Managing Mass Storage Devices and DG/UX® File Systems — 069–701144

Managing Modems and UUCP on the DG/UX® System — 069-000698

Managing ONC/NFS and Its Utilities on the DG/UX® System — 069–701136

Managing TCP/IP on the DG/UX® System — 069–701137

Programming with TCP/IP on the DG/UX® System — 069–701135

Using the DG/UX CLARiiON Manager™ Interface — DG/UX® Environments — 069–000792

Using TCP/IP on the DG/UX® System — 069–701134

X Windows User's Guide — 069-100392

DG/UX performance and debugging manuals

The following manuals contain information about performance monitoring and debugging software.

Analyzing DG/UX® System Performance — 069–701142

Using the DG/UX® Kernel Debugger — 069–701138

Using the Multi-extensible Debugger (Mxdb) for DG/UX® Systems — 093-000710

AV/Alert manual

The following manual supports the optional software package used for detecting hardware problems that are forwarded automatically to the Customer Support Center for diagnosis.

Using AViiON® Diagnostics and the AV/Alert℠ Diagnostic Support System — DG/UX R4.10 Environment 014–002512

DG/UX 5.4 file system technical brief

This technical brief explains the internals of the DG/UX file system in detail.

The DG/UX® 5.4 File System — 012–004054

How this manual is organized

The following list gives an overview of what you will find in this manual:

Chapter 1	**Service agreements**
Chapter 2	**Avoiding problems**
Chapter 3	**Maintaining sufficient file system space**
Chapter 4	**Developing a policy for handling halts**
Chapter 5	**Log files**
Chapter 6	**Performance problems**
Chapter 7	**Diagnosing a problem**
Chapter 8	**Common problems and error messages**
Chapter 9	**Recovering from power failures, hangs, and halts**
Chapter 10	**Fixing corrupt file systems with fsck**
Chapter 11	**Frequently asked questions**
Appendix A	**Tuning the system and network logging facilities**
Appendix B	**Fsck messages**
Appendix C	**Using stand-alone sysadm**
Appendix D	**System configuration data worksheets**
Appendix E	**Data General Software Trouble Report (STR) form**

Format conventions

We use the following format conventions in this manual:

Convention	Meaning
boldface	Indicates text (including punctuation) that you type verbatim.
boldface	All DG/UX commands, pathnames, and names of files, directories, and manual pages appear in this typeface.
monospace	Represents a system response (such as a message or prompt), a file or program listing, or a menu path.
italic	Represents variables for which you supply values; for example, the name of a directory or file, your username or password, and explicit arguments to commands.
...	Means you can repeat the preceding argument as many times as appropriate.
#, $ and %	Represent the system command prompts for the superuser, and the Bourne and C shells, respectively. Note that your system might use different symbols for the prompts.
↵	Represents the Enter key. (On some keyboards this key is called Return or New Line.)
< >	Angle brackets distinguish a command sequence or a keystroke (such as <**Ctrl-D**>, <**Esc**>, and <**3dw**>) from surrounding text. Don't type the angle brackets.

Contacting Data General

Data General wants to assist you in any way it can. Please feel free to use the contact information below.

Manuals

If you require additional manuals, please 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, telephone support is available with your hardware warranty and with Support Plus and Hotline Software Support service contracts. 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.

Data General on the World Wide Web

Data General's comprehensive information library provides Internet users with access to virtually all of Data General's publicly available information and to a variety of feature articles and white papers on critical issues in computing. Browse through product and service catalogs, our Solutions Directory, partner and customer profiles, and other publications.

Internet users can reach Data General's web server at **http://www.dg.com**. In addition, you can send us e–mail at aviion@dg.com.

Data General's Customer Support Center (CSC) provides Internet users with access to a Service Request Menu, Electronic Search Program, a Bulletin Board, Monthly Newsletters, Weekly Bulletins,

Maintenance Updates, patches, and important information on a variety of operating systems. An active support contract may be required for certain features.

Internet users can reach the CSC web server at **http://www.csc.dg.com**.

Joining our users group

NADGUG, the Independent Data General Users Group, is a unique network of Data General users who wish to discover more about working computer environments and how to use them. As a professional, user–run organization, NADGUG members make contacts with colleagues, share expertise and perspectives, and influence the companies that provide products and services. The User Group is your connection to expert users of Data General products, including AViiON open systems, CLARiiON advanced storage solutions, PCs, and ECLIPSE MV systems. Getting just one solution from a fellow member can more than pay for the cost of belonging to NADGUG. To join or for more information, call 1–800–253–3902 or 512–335–2286.

End of Preface

Contents

Chapter 4 – Developing a policy for handling halts

Chapter 5 – Log files

Chapter 6 – Performance problems

Chapter 7 – Diagnosing a problem

Chapter 8 – Common problems and error messages

Chapter 9 – Recovering from power failures, hangs, and halts

Chapter 10 – Fixing corrupt file systems with fsck

Chapter 11 – Frequently asked questions

Tables

Table

Figures

Summary of changes

Changes for Release 4.20MU05

The following changes were made for Release 4.20MU05

Chapter 4	Added information on enhanced dump support for full memory dumps. In addition, information on storage of halt dumps on virtual disks acknowledges that users can dump to unstriped aggregations of simple partitions if all pieces, up to a total of 8, are on the same physical disk.
Chapter 11	Cut a problem note documented under “Restoring a file system from tape.” This problem has been remedied. In addition, added a note about the capability to boot directly from a mirror.

1 Service agreements

You probably obtained your service agreement when you purchased your DG/UX® system. When troubleshooting your system, be familiar with the features of your agreement. Also, you may purchase extensions to your agreement and optional services over the lifetime of your hardware equipment and software products.

Basic service agreements

The basic types of service agreements, ranging from most to least comprehensive, are:

OMNiiSERVICE℠
PASS (ProActive Software Service)
Support Plus
HSS (Hotline Software Support)
SSS (Software Subscription Service)
DASH (Direct Access to Software Help) Electronic Services

OMNiiSERVICE

OMNiiSERVICE is a comprehensive set of hardware and software support services, which are designed for customers with AV 5500, 8500, and 9500 Enterprise environments. You already must have the Support Plus service agreement before purchasing the additional service offered by OMNiiSERVICE. Refer to Table 1–3 for Support Plus features and benefits.

Features and benefits of OMNiiSERVICE follow:

Table 1–1 OMNiiSERVICE features and benefits

Features	Benefits
General	
Assigned support coordinator in the Customer Support Center (CSC)	Coordinates customer request for service.
Local branch manager semi-annual on-site review	Maintains close relationship with customer to review service history and to make future plans.
Quarterly reports of both hardware and software calls	Helps you optimize system operations by pinpointing trouble spots.
Hardware	
99.5% guaranteed uptime	Maximizes your productivity and system's high availability.
Priority response and dispatch	Critical system trouble calls receive almost immediate attention from CSC personnel and account engineering on-site, if necessary.

Continued

Table 1–1 OMNiiSERVICE features and benefits

Hardware	
7 x 24 coverage	Authorized for three callers at your site; toll-free telephone support for the critical system (CPU and cabinet, master consoles, and mass storage devices) seven days a week, around-the-clock; problem diagnosis and assistance. Printers, terminals, personal computers, and external network devices are excluded.[1]
Software	
7 x 24 coverage of DG/UX	Extends phone support to seven days a week, twenty–four hours a day for DG/UX software only. No bundled products are supported — such as Gnu C, TCP/IP, NFS, X Window System, or OSF/Motif.
Proactive patch notification	Gives customers more time to plan and implement changes and fixes.
Local or remote operating system installation or upgrade	Eases responsibility for performing installation and upgrades.

1 You may purchase extra coverage for additional callers. Order model W001AZN65N for each additional caller desired.

PASS — ProActive Software Service

PASS is aimed at large customer sites with complex installations, requiring advanced notice of software updates and revisions, and more personal attention. You already must have the Support Plus service agreement before purchasing the service extensions offered by PASS. Refer to Table 1–3 for Support Plus features and benefits.

Features and benefits of PASS follow:

Table 1–2 ProActive Software Service features and benefits

Features	Benefits
Designated account specialist who makes an initial site visit	Improves understanding of your needs.
Proactive communication	Alerts you to releases, problems, and solutions.
Proactive patch notification	Gives customers more time to plan and implement changes and fixes.
Quarterly reports	Help you to plan the service you need.

Support Plus

Support Plus is recommended for most customers, especially new customers or customers purchasing new software products. Support Plus extensions are required in a multiple-systems environment; see later section for details. Features and benefits of Support Plus follow:

Table 1–3 Support Plus features and benefits

Features	Benefits
Authorized for three callers at your site, for all products — toll-free telephone software support, Monday – Friday, 8 am – 5 pm, customer's local time, for all products. For selected products — seven days a week, 8 am – 5 pm, customer's local time.[1]	Provides immediate availability of software specialists to respond to all software problems and usage questions.
Remote Software Support allows software specialists direct access to a customer's system and situation.	Facilitates quick problem diagnosis and electronic delivery of fixes.
On-site assistance for problem isolation at Data General's discretion.	Confidence that on-site help is available, resulting from consultation with Data General.

Continued

Table 1–3 Support Plus features and benefits

Features	Benefits
DASH Plus Electronic Services	"Around-the-clock" access to help information on system operation and notice of fixes.
Software problem reporting	The Customer Support Center (CSC) will write, submit, and track Software Trouble Reports (STRs) on your behalf.
Software Subscription Service provides automatic distribution of software and documentation updates, and periodic technical newsletters.	Automatic receipt of improvements and fixes to assure smooth system operations.

1 You may purchase extra coverage for additional callers. Order model W001AZN65N for each additional caller desired.

SSS — Software Subscription Service

SSS is aimed at the more highly technical customer who may not need telephone support.

It provides automatic distribution of software and documentation updates and periodic technical newsletters.

SSS is required in a multiple-systems environment. Refer to the section on "Software service extensions" for information on service upgrades.

HSS — Hotline Software Support

HSS is available on selected AViiON® software products only, excluding the DG/UX system. Its features and benefits follow:

Table 1–4 Hotline Software Support features and benefits

Features	Benefits
Authorized for three callers at your site; toll–free telephone software support, Monday – Friday, 8 a.m.- 5 p.m., customer's local time, for selected products; problem diagnosis and assistance and use and operation of software.[1]	Provides immediate availability of software specialists to respond to all software problems and usage questions.
Remote Software Support allows software specialists direct access to a customer's system and situation.	Facilitates quick problem diagnosis and electronic deliveries of fixes when available.
On-site assistance for problem isolation at Data General's discretion.	Confidence that on-site help is available when needed.
DASH Plus Electronic Services	"Around-the-clock" access to help information on system operation and notice of fixes.
Software Problem Reporting	The Customer Support Center (CSC) will write, submit, and track Software Trouble Reports (STRs) on your behalf.

1 You may purchase extra coverage for additional callers at your site. Order model W001AZN65N for each additional caller desired.

DASH — Direct Access to Software Help

DASH is a menu-driven facility, accessible by modem or the Internet, that offers a variety of self-help customer services, some of which are free to all customers, with others offered through service agreements only. Refer to the *DASH Electronic Services Users Manual* for details on accessing and using DASH.

Free to all customers

DASH electronic services provide customers with electronic access to the free Basic Services listed in Table 1–5, seven days a week, 24 hours a day, using either of these modes of communication:

- Toll-free modem access — **1–800–DASH CSC**
- Internet — **128.222.159.141** or **csc101.us.dg.com**

Table 1–5 DASH Basic Electronic Services features and benefits

Features	Benefits
Bulletin board	Customers can read, post to, and respond to items on the board, which include these categories: Announcements, DG/UX, Networking, Languages, and Hardware.
Electronic mail	Internal to the DASH system, this mail system allows customers to communicate with each other.
STR lookup	Allows customers to check status of any STR by number.
Training and publications	Allows customers access to lists of courses and publications offered by Educational Services.

Available through service agreement

In addition to the DASH Basic services, DASH Plus provides the following features and benefits free of charge, to Data General AViiON hardware service agreement customers and Support Plus customers.

Table 1–6 DASH Plus Electronic Services features and benefits

Features	Benefits
Weekly bulletins	Exposure to customer and product issues.
Electronic search facility, allowing you to query a database of common symptoms and solutions for DG product problems	Provides an immediate solution to the most commonly reported software problems and their resolutions.
Electronic submission of problems and questions about software	Provides next business-day response to software inquiries from the same specialists who provide telephone support.
Electronic patch and maintenance update ordering	Electronic delivery of patches to customer computer advised at discretion of Support Center.

Comparison of service agreement features

A comparison of all basic service agreement features follows:

Table 1–7 Comparison of service agreement features

Feature	SSS	HSS	Support Plus	PASS	OMNiiSERVICE
Branch manager makes semi-annual on-site visit					●
99.5% guaranteed uptime					●
Priority response and dispatch at General's discretion				●	●
Designated account specialist				●	●
Proactive patch distribution				●	●
Activity reporting				●	●
Initial copies of software media and documentation	●		●	●	●
Automatic distribution of software and documentation updates	●		●	●	●
Periodic newsletter	●	●	●	●	●
File Software Trouble Report (STR)	●	●	●	●	●
STR tracking		●	●	●	●
Telephone support		●	●	●	●
Remote support		●	●	●	●
On-site assistance		●	●	●	●
DASH Plus electronic services		●	●	●	●
DASH basic		●	●	●	●

Software service extensions

We offer these extensions to your service contract:

Multiple system coverage
Additional caller coverage
DG/UX System 7 x 24 coverage

Multiple system coverage

If you have multiple systems, and your primary system is covered by Support Plus or SSS, you must purchase additional service for each secondary system.

If you have Support Plus, you must purchase two additional service features for your secondary system:

- the phone support component of the HSS plan, and
- the Right to Copy (RTC) and run updated software on the secondary system. To run updated software on any system, you must purchase RTC or a service that includes RTC on each system on which the updated software will be run.

Select the appropriate model number to order additional phone service:

for personal computers	W002AS265N
for workstations	W002ASQ65N
for deskside servers	W002ASX65N
for large servers	W002ASY65N

If you have only SSS for your primary system, you must purchase the Right to Copy (RTC) service to run updated software on your secondary systems. To run updated software on any system, you must purchase RTC or a service that includes RTC on each system on which the updated software will be run.

A summary of service extensions follows:

Table 1–8 Multiple service extensions for secondary systems

Primary System	Secondary System
Support Plus	Additional system's HSS + RTC
SSS	RTC

Additional callers

If you already have the Support Plus or HSS agreement, you may purchase extra coverage for additional callers at your site above the three granted. Order model W001AZN65N for each additional caller required.

DG/UX 7 x 24

You may purchase around-the-clock phone support for the DG/UX operating system only (no bundled products such as Gnu C, TCP/IP, NFS, X Window System, or OSF/Motif). For each system on which 7 x 24 coverage is desired, order model W003AZN65N.

Billable services

If you require a service that is not covered by your agreement, Data General can provide the service, either on-site or remote, for an additional fee. Such services include installation and configuration of the DG/UX operating system; loading of software and databases; programming, such as writing a script or an application interface; complete application or system development; network design, planning, implementation, and support; as well as on-site residencies.

SEPAC

SEPAC (Systems Evaluation and Performance Analysis Center) offers a broad range of performance analysis, tuning, and capacity analysis software and services. Fundamental to these services is the add-on software package DG/UX Real-time Performance Monitor (UX/RPM) which allows system administrators to quickly view and measure key performance statistics. Easy to use, UX/RPM collects statistics that help to determine potential bottlenecks and user processes that may affect performance.

You may elect consultation services, whereby a consultant dials in to your system and uses UX/RPM to monitor your system's activity for a specified time period. After the monitoring period, the consultant prepares a report, offering recommendations for improved performance.

Training

You may purchase and enroll in courses that you choose from the Data General Educational Services curriculum. Select from a variety of training formats: lecture and laboratory training in the Data General education centers, on-site training at your facility, computer-based training, and video-based training.

Some of the current DG/UX training courses include:

- DG/UX systems operations
- Using and administering a basic DG/UX system
- DG/UX system administration for AViiON systems
- Advanced DG/UX system administration
- DG/UX System tuning
- Open systems troubleshooting
- DG/UX system programming

For a full list of courses that currently are available, call 1–800–343–8842, option 5, and speak with an educational consultant.

Business recovery

To insure your hardware and software components against disaster, you may purchase a business recovery agreement. In the event of a disaster, this contract provides you with a full system replacement, shipped on a priority basis.

End of Chapter

2 Avoiding problems

System problems often catch us unprepared, causing us to scramble for explanations and help. Not all, but some, problems can be avoided by following good system administration practices and understanding your DG/UX configuration. This chapter offers hints for avoiding problems and highlights some useful commands and files that describe the configuration.

If you contact the Customer Support Center for help, you will need to supply some of the following information. If you complete a troubleshooting worksheet, located in Chapter 7, you will use the commands in this chapter to collect useful information.

Tips for good system administration

This section lists tips for good system administration, which are divided into three categories: things to do, things not to do, and things to know.

Things to do

▶ **Use the installation planning worksheets and keep system administration records.**

Keeping a record of your configuration (that you completed at installation time) gives you valuable information when you need it, such as during upgrades and times when you need to troubleshoot your system. Such records allow you to track changes to the configuration over time and provide useful historical information for future system administrators.

Since the system is sometimes down when troubleshooting, online system information will be inaccessible. So, keep a notebook to record useful information, such as hardware and software changes, hangs and halts (with halt codes), and service calls. Also, record carefully the exact commands and parameters used for backing up and restoring file systems. Document maintenance activities with dates and names of those involved. Record system configuration data; refer to Tables 2–1 and 2–2 for a list.

- **Back up your system on tape according to a backup schedule.**

  Refer to *Managing the DG/UX® System* for details on developing a backup schedule. A catastrophic event, such as a disk drive failure, may require that you restore your system entirely from backup. Also, back up your system before you upgrade the DG/UX system or apply a patch or maintenance update in case you need to reinstate the system to its previous revision.

  In addition to backing up your system on a regular schedule, you should also use the **systemtape** utility to build a bootable tape of your **/** and **/usr** file systems in case these file systems are accidentally corrupted. For information on using the utility, refer to the **systemtape**(1M) manual page; for information on when you may need to restore file systems from a bootable tape, refer to Chapter 11.

- **Notify users when you plan to upgrade the operating system so that they can conclude their work before logging off.**

  Also, consult with users before altering the hardware environment. A user may rely on a device that you intend to remove from service.

- **Consider the implications before you remove any hardware or software from your configuration.**

  If you physically remove a device, you should also deconfigure it and then rebuild the kernel, removing the device from the system file. If you disable an application, make sure that you likewise remove attendant files, if necessary. Notify users of plans to alter the configuration.

- **Make sure that the release you plan to install or upgrade is compatible with the current release of the operating system.**

  Carefully read the release or update notice to determine the requirements for installing the new release or update. Also, before you attempt an installation or upgrade, make sure that your applications will be compatible with the new release of the operating system.

- **Always read the release notice and the documentation.**

  Read the release notice that accompanies DG/UX system releases and updates. It alerts you to changes from the previous revision. Be aware that the content of a customer document changes from revision to revision. If a new document ships with a DG/UX system update, discard its predecessor and use the currently shipped version.

When consulting with the Customer Support Center about how to use a DG/UX feature or how to fix a problem, they will first refer you to the documentation if the feature or problem is explained there. Try locating the desired information in the documentation before contacting the Support Center.

▶ Access DG/UX documentation on CD-ROM.

> **IMPORTANT** For information on hardware requirements and how to set up the documentation viewing software, refer to the documentation for the DG/UX documentation CD-ROM.

As an option to reading traditional printed documentation, you may prefer to read documents on line. A major advantage is that you may find the desired information without having to know the title of the book that contains it. You may perform a full text query, searching for a particular subject (or text string) that may be in multiple documents. After identifying all the documents (and chapters) that contain the desired subject, you can proceed directly to that information.

▶ Use on-line manual pages.

The DG/UX system provides on-line manual pages, which give technical descriptions of all attributes for each command, system call, or special file in the system. A manual page is accessible by typing at the shell: **man** *command*; for example **man admfilesystem**. Refer to *Using the DG/UX® System* for more information on using manual pages.

▶ Monitor system and network log files.

It is a good practice to regularly monitor log files to verify your system's healthy operation. Examining logs may bring your attention to a problem before it is manifested in an error message or in some other way. Identifying a problem early may circumvent a serious problem later.

Besides monitoring the information in log files, you must also keep track of their sizes. Various system facilities send messages to these files on a continual basis, causing them to grow. Some facilities control the size of their logs and take care of their own cleanup, while others do not. To maintain sufficient disk free space, monitor your log files that require cleanup. Refer to Chapter 5 for information on log files.

▶ Develop a halt policy and define a default dump device destination.

You must set up a default dump destination during normal system operation so that the system knows where to direct a memory dump under a halt condition. Detected by the kernel, the halt indicates an internal system software malfunction or inconsistency. Typically, when the system halts, it is necessary to take a system dump (copy memory to a designated device) to send to the Customer Support Center for their analysis.

If a default dump destination is not defined, your system will not be able to create a dump. The lack of a system dump can be a serious obstacle to finding and fixing system problems.

You must develop a halt policy in advance of the halt situation. Typically, you will decide whether or not to take a dump automatically when the halt occurs. If you opt to take a dump, you can define the local dump device to which the memory dump is sent: virtual disk or tape device. OS clients may dump to a network-defined device.

You may specify a dump device using these methods:

- DG/UX system installation
- **dg_sysctl** command
- watchdog timer kernel parameter selection
- SCM menu

During installation, you may specify a virtual disk to capture a kernel memory dump upon a halt condition. For systems with a requirement for high availability, we recommend selecting the virtual disk dump device. See *Installing the DG/UX® System* for details on selecting a dump virtual disk.

In addition to selecting the dump virtual disk, you may select other halt options through the **dg_sysctl** command, the SCM, and the watchdog timer. You may check the current default halt behavior by typing **dg_sysctl** at the shell prompt. You may check the autoboot and boot path values set through the SCM by typing **f** at the SCM prompt. You may check on whether the watchdog timer is configured in your kernel by typing **sysdef** at the shell prompt, as superuser. For details on setting a dump device, see Chapter 4.

▶ **Keep Software Trouble Report (STR) forms handy.**

Go to Appendix E for a copy of the STR form. Make copies of this form to have on hand if and when you need to file them.

You will also find an STR form that is suitable for printing on a line printer in the file **/usr/release/STR_form** on your DG/UX system. Print an STR form while your system is operational and make copies for future use. Remember that you may not be able to access online STR forms when your system is experiencing difficulties.

STR forms are also available from the nearest Data General office, from your sales representative, or from the Customer Support Center. Depending on your service agreement, you can submit STR reports to the Center or directly to your sales representative. See Chapter 9 for details on filing an STR.

▶ **Collect process data and CPU and memory usage data regularly.**

Establish a policy for monitoring system and user processes and CPU and memory usage. Performance data collected during normal operation will provide a useful baseline against which to compare abnormal system performance data.

Particularly useful statistics are gathered with the **nps** (process status) and **nsar** (system activity reporting) commands. The **sysadm** utility also provides a menu-interface for **nps**, and AV Syscope provides an interface to the **nsar** command. Refer to *Managing the DG/UX® System* and *Using the AV SysScope® Performance Monitor* for information on these interfaces.

Refer to Chapter 6 for complete details on fixing performance problems.

▶ **Select a file recovery method before you mount file systems.**

A fast recovery file system check is preferable primarily for file systems where rapid recovery and high availability are crucial. The fast option reduces downtime by using file system logging. Logging keeps records about file system updates, which permits a quick reconstruction of file systems from the log's records. In the event of a file system failure, the file system checker can use the log to speed the process of verifying and restoring file system integrity. A disadvantage of logging is that it slows down write performance in the file system during normal operation.

You select a form of file system recovery when you add file systems to the **/etc/fstab** file. If you choose fast file recovery, you may also wish to tune these kernel parameters: RUNFSCK, FSCKFLAGS, and ROOTLOGSIZE. See Chapter 10 for information on running the file system checker, **fsck**, and see *Managing Mass Storage Devices and DG/UX® File Systems* for information about adding file systems to **fstab**.

▶ **Mount file systems in the proper order.**

"Parent" file systems should be mounted before "child" file systems. For example, if you have file systems for both **/var** and **/var/spool**, mount **/var** before **/var/spool**. Unmount file systems in the opposite order from which they were mounted to avoid possible loss of file system records.

▶ **Identify software and hardware experts at your site.**

There may be a wealth of knowledge at your site about how the DG/UX system behaves and operates that you have not yet tapped. You may consult other system administrators, network administrators, or any person who is knowledgeable about the system. Use your internal contacts as an aid to identifying and solving system or operational problems.

▶ **Join Data General's Users Group.**

Join the North American Data General Users Group (NADGUG), the largest independent organization of Data General users. In addition to making valuable contacts, you 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 NADGUG, call **1–800–253–3902** or **1–508–443–3330**.

▶ **Use Data General's World Wide Web server.**

Data General's comprehensive information library provides Internet users with access to virtually all of Data General's publicly available information and to a variety of feature articles and white papers on critical issues in computing. Browse through product and service catalogs, our Solutions Directory, partner and customer profiles, and other publications.

Internet users can reach Data General's web server at **http://www.dg.com**. In addition, you can send us e–mail at

aviion@dg.com.

Data General's Customer Support Center (CSC) provides Internet users with access to a Service Request Menu, Electronic Search Program, a Bulletin Board, Monthly Newsletters, Weekly Bulletins, Maintenance Updates, patches, and important information on a variety of operating systems. An active support contract may be required for certain features.

Internet users can reach the CSC web server at **http://www.csc.dg.com**.

Internet users can reach the Common Sense Connection through **http://www.dg.com** or **gopher.dg.com**. You must be connected to a network and have access to a WWW browser such as **Mosaic** to access the WWW. In addition, you can send us e-mail at **commonsense@dg.com**.

▶ **Subscribe to the Data General users' newsgroup.**

The **dg–users@ilinx.wimsey.bc.ca** newsgroup offers a forum for users of Data General products to discuss common interests and issues. To subscribe to the newsgroup, send your request by electronic mail to **dg–users–request@ilinx.wimsey.bc.ca**. You must be connected to a network to send e-mail and to receive mailings from this newsgroup.

Things not to do

IMPORTANT Some accidental actions can destroy data. The following tips warn against these inadvertent errors.

▶ **Do not edit directly the /etc/passwd, /etc/inittab, or /etc/fstab files.**

Instead, use **sysadm** or an appropriate shell command to operate on these files. Typographical errors in these files can cause unrecoverable problems, which may require you to restore selected files from backup or entirely re-install the operating system.

▶ **Do not execute the rm * command in the / directory.**

This command will remove files, links, and possibly directories (signified by the asterisk wildcard), causing much trouble. Restoration of files and links will be troublesome and time-consuming, requiring that you reconstruct symbolic links and rebuild the kernel. If you accidentally shut down the system before realizing your error, you must reinstall the DG/UX system entirely. Unless you can restore exact replicas of tuned configuration files from backup, all tuned variables in configuration files will be lost.

Things to know

▶ **Be able to identify the form of the DG/UX distribution that you receive and install on your system: release, update, patch, maintenance update (MU), extension, and supplement.**

Most customer service agreements, covered in Chapter 1, include the automatic distribution of DG/UX system software on tape or CD-ROM and documentation (hardcopy and CD-ROM). Specific types of distributions, however, are limited to the requesting customer. The DG/UX software comes in these forms, which are clearly labelled on the release medium and the accompanying notice.

release A group of components composing a product, that when correctly loaded onto a system will result in a fully functional system. You may either install a release of DG/UX for the first time or you may "overload" an existing system, overwriting the previous DG/UX system completely or partially. Releases are distributed automatically to all customers who have an SSS, Support Plus, PASS or OMNiiSERVICE service agreement.

update A group of components composing a product that is designed to overload a product release. Overloading may selectively replace or completely replace the previous DG/UX system. Both methods of overloading are transparent to the user. Updates automatically are distributed to all customers having an SSS, Support Plus, PASS or OMNiiSERVICE service agreement.

patch One or more modules that replace modules on the DG/UX system. A patch fixes a specific problem that a customer reports through a software trouble report (STR). Fixes made available through patches are included in subsequent releases (and updates) of the product. Patches are distributed to only the requesting customer.

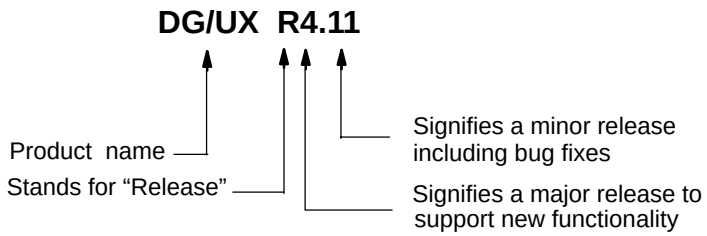
maintenance update (MU) A collection of finalized patches for a specific DG/UX release or update that is regularly scheduled and widely distributed to the customer base to reduce the time and effort needed to install individual patches. Successive MUs for a particular revision of DG/UX are cumulative. The content of the MU is included in the next release or update of the DG/UX operating system. An MU is distributed automatically to customers on PASS and OMNiiSERVICE agreements and to all other customers who request it.

extension An extension is a special "one-of-a-kind" fix to a specific problem that a customer reports. An extension is like a patch except that it will not become part of the next release (or update) of the product. An extension is distributed to only the requesting customer.

supplement A supplement is like a patch except that it is released through Data General Manufacturing. In most cases, a supplement supports specific hardware whose ship date does not coincide with other scheduled software releases, or it supports a layered software product that cannot ship without a change to the DG/UX operating system software. The content of the supplement is included in the next release, update, or MU of the DG/UX operating system. A supplement is distributed to only those customers who receive the new hardware or order the layered product.

▶ Understand the DG/UX product numbering scheme.

The DG/UX operating system release numbering scheme was designed to accurately identify the version of the product. The official name for the current product is:



Comparing the previous DG/UX number with the current DG/UX number shows what aspect of the operating system has changed. For example, there is one difference between the previous and the current DG/UX release numbers:

```
DG/UX R4.11
DG/UX R4.20
```

The minor release field was incremented to indicate that bug fixes were included in the major release. The DG/UX R4.20 release introduces bug fixes to that major release.

Knowledge of your current release level is important particularly when installing releases, updates, maintenance updates, or patches on your system. You must consider the compatibility of the release you are installing with the current release. Carefully read the notice that accompanies the release medium for information on prerequisites. To find out the current release level, check the entry in the **/etc/issue** file or use the **uname –a** command.

▶ **Know the shell command equivalents to DG/UX system administration (sysadm) operations.**

The DG/UX documentation explains how to manage the system through its interactive, menu-driven system administration facility known as **sysadm**. Also provided are the **adm** commands (on which **sysadm** is based) and shell commands that are incorporated into the **adm** command set. Use of either method is a matter of personal preference. However, to expedite the troubleshooting process, you may prefer using an **adm** or shell command, thus eliminating the overhead of initializing **sysadm**. The nature of the problem you're having, in fact, may prevent you from using **sysadm**.

As a preventive measure, you may wish to collect **adm** command equivalents during a time when system operations are normal (there are no problems). Collecting these commands ahead of time will prepare you for a potential troubleshooting situation down the road.

> **IMPORTANT** **Adm** and shell commands, rather than **sysadm** operations, are used throughout this book for illustrating preventive and troubleshooting procedures.

An easy way to find the **adm** command equivalent to a **sysadm** operation is to set system verbosity through the **sysadm** interface.

1. In **sysadm**, follow this path:

   ```
   Session -> Parameters -> Set
   ```

2. Accept the default intermediate level of expertise by pressing Enter.

   ```
   Level of Expertise: [Intermediate]  ↵
   ```

3. Select the verbosity level that produces the **adm** command equivalent for each **sysadm** operation you perform.

   ```
   Verbosity: [Informational] Command invocations ↵
   ```

   Once this feature is selected, the **adm** command equivalent is written to the screen and is logged to **/var/adm/log/sysadm.log** as you execute a **sysadm** operation. Print the **sysadm** log to keep in your permanent records.

   IMPORTANT The verbosity parameter is restored to its default value — Informational — each time you restart a **sysadm** session. Reset it to the desired value, as needed.

▶ Keep hardware and software support hotline numbers handy.

Be prepared for potential problems requiring assistance by having customer support center toll-free numbers handy. Data General's Customer Support Center's toll-free number is: **1–800–DG–HELPS**. The hours of service available depend on your particular service agreement. Refer to Chapter 1 for a review of the terms of your agreement.

Also, know the toll-free numbers of third-party software and hardware technical support centers for companies whose products you run in a DG/UX environment. Data General cannot help you with problems relating to software and hardware provided by other companies. Record these numbers in your system records.

IMPORTANT Data General supports some third-party products. Check your service agreement and product notices, and check with your sales representative for this information.

▶ Understand how your applications operate.

If you have problems with either third-party or Data General applications that run on the DG/UX system and you seek the help of the Customer Support Center, be able to discuss the organization and behavior of the application with them. Read the application's documentation and accompanying notices.

Commands that report configuration data

IMPORTANT To perform some administrative tasks, you must have appropriate privilege. On a traditional DG/UX system, appropriate privilege means having an effective UID of 0 (**root**). On a system with DG/UX information security, appropriate privilege means having one or more specific capabilities enabled in the effective capability set of the user. See the manual pages for **appropriate_privilege**(5) and **cap_defaults**(5).

Table 2–1 summarizes commands for collecting useful configuration data about your configuration. You may record this information in worksheets provided in Appendix D to keep in your system administration records.

Table 2–1 Commands to collect information about your system

To find out:	Use this command:
System	
Operating system name, host name, operating system release, operating system version, machine hardware name, and host processor type	**uname –a**
System architecture, clock speed, processor type, model number, hardware caches, physical memory, free memory, free swap (for systems running DG/UX 5.4R43.00 and later revisions)	**dg_sysreport**
Processor speed, number of processors, firmware revision, Ethernet address, and physical memory size	power-up output
Run level	**who –r**
When your system was last booted	**who –b**
When the kernel linked to **/dgux** was last built	**ls –ail** *kernel-that-was-booted*
Vendor stamp and the system identifier NOTE: A system ID is not supported on all Intel platforms	**systemid**

Continued

Table 2–1 Commands to collect information about your system

System	
Pre-defined behaviors set for a system halt; see Chapter 4 for details	**dg_sysct**l
User license limit	**admuserlicense –o list –c**
Disks (Physical and Virtual)	
Statistics about configured disks	**admpdisk –o list**
List of configured devices	**sysdef** *kernel-that-was-booted*
Statistics about virtual disks	**admvdisk –o list**
File Systems	
Statistics about mounted file systems	**admfilesystem –o list –m**
Amount of file system free space in kbytes NOTE: A user with appropriate privilege can access an additional 10 percent more space than reported.	**df –k**
Amount of file system free space in blocks, including the additional 10 percent available to a user with appropriate privilege.	**df –t**
Used and free disk space, including 10 percent additional space accessible only to a user with appropriate privilege	**admfsinfo –o diskuse –l**
Number of disk blocks used by a file or directory	**du –s**
Network	
If network device is configured NOTE: **sysdef** command reports LAN controller name	**ifconfig** *LAN-controller*
Names of network devices	**netstat –i**

Continued

Table 2–1 Commands to collect information about your system

Network	
If particular remote host is operational	**ping** *hostname*
Internet and Ethernet address (if connected to a network)	**arp** *hostname*
Domain name (if connected to a network and NIS (Network Information Services) package is loaded and set up)	**domainname**
Packages	
What packages have been loaded on an OS client	**admpackage –o list –c** *OS-client*
Statistics about packages that are loaded on your host	**admpackage –o list –v**

Examples of commands used to collect configuration information with output follow:

IMPORTANT Some output has been abbreviated.

System

```
# uname -a ↵
dgux homer R4.11 generic AViiON mc88110
```

where:

dgux is the system name
homer is the host name
R4.11 is the operating system release
generic is the operating system version
AViiON is the machine hardware name
mc88110 is the host's processor type

The following typical output is generated during power-up of your computer:

```
(C) DATA GENERAL CORPORATION 1989 ...  1997
40 MHz AV Series Computer
Dual Processor
Firmware Revision 08.08
Keyboard Language is U.S. English
Local Ethernet address is 08:00:1B:1F:03:77
Initializing [32 Megabytes]

Testing.....
    0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
Passed
```

where:

40 MHz is the processor speed
dual processor refers to the system having two CPUs
Firmware revision is **08.08**
Local Ethernet address is **08:00:1B:1F:03:77**
Physical memory size is **32 megabytes**

```
# who -r ↵
.       run-level 3  Feb 24 13:20    3    0    S

# who -b ↵
.       system boot  Feb 24 13:20
```

When rebooting the system, it is useful to know which kernel is a hard link to **/dgux**. The **–i** option lists kernel inode numbers.

```
# ls -ail /dgux* ↵
28236 -rwxr-xr-x   2 root   other   4195798 Mar 15 08:40 /dgux
28225 -rwxr-xr-x   1 root   other   4429830 Mar 28 13:52 /dgux.chip
 4037 -rwxr-xr-x   1 root   sys     3579128 Nov 29 13:30 /dgux.installer
 4164 -rw-rw-rw-   1 root   other   4616373 Mar 28 16:17 /dgux.ph
28236 -rwxr-xr-x   2 root   other   4195798 Mar 15 08:40 /dgux.wednesday
```

Look for the matching inode numbers, which are listed in the first column. In the example, since the inode numbers for **/dgux** matches the inode number of **/dgux.wednesday**, you can conclude that **/dgux.wendesday** is linked to **/dgux**. The kernels **/dgux.ph** and **/dgux.chip** were possibly built for a special purpose, while **/dgux.installer** is shipped with DG/UX and is sometimes used for system recovery.

```
# systemid ↵
Vendor stamp: 512 (Data General DG/UX)
System id:    071ff7dc
```

IMPORTANT A system ID is not supported on all Intel platforms.

```
# dg_sysctl ↵
AUTOBOOT=halt
BOOTPATH=sd(cisc(),3)root:/dgux
AUTODUMP=ask
DUMPDEV=st(cisc(),4)
DUMPLEVEL=kernel
POWEROFF=skip
```

```
# admuserlicense –o list –c ↵
Current license is for 2 users
Current user count is  1 users
```

Physical and Virtual Disks

```
# admpdisk –o list ↵
Disk name          State   Reg? Format Total blocks  Free blocks
sd(insc(0),0,0)    avail      y vdisks      1295922      1230279
```

If you booted a kernel not hard linked to **/dgux**, you must specify the kernel file name.

```
# sysdef ↵
# Configured devices
#
kbd()
grfx()
lp()
duart(0)
inen()
wdt()
sd(insc(),0)
st(insc(),4)
st(insc(0),*)
...
```

```
# admvdisk –o list ↵
Name              Volume Temp  Size Type
root                V      40000 partition on sd(insc(0),0,0)
swap                V     100000 partition on sd(insc(0),0,0)
test                V        100 partition on sd(insc(0),0,0)
usr                 V     300000 aggregation of 3 pieces
usr_opt_X11         V     140000 partition on sd(insc(0),0,0)
```

File Systems

```
# admfilesystem –o list –m ↵
File System                            FS          NFS     Dmp Fsck
Source           Mount directory      Type   RW  Mount    Frq Pass
--------------   ------------         ----   --  -----    --- ----
/dev/dsk/root    /                    dg/ux  rw            0    0
/dev/dsk/usr     /usr                 dg/ux  rw            0    0
/dev/dsk/usr/leaf       /usr/leaf     dg/ux  rw            0    1
/dev/dsk/usr_opt_x11    /usr/opt/X11  dg/ux  rw            0    1
viper:/usr/local        /usr/local      nfs  ro soft,intr  0    0
custdoc:/usr/opt/iview  /usr/opt/iview  nfs  rw soft,intr  0    0
dg–rtp:/usr/opt/fram /usr/opt/frame3.1  nfs  rw soft,intr  0    0
```

```
# df –k / /usr ↵
filesystem     kbytes    used   avail capacity  mounted on
/dev/dsk/root   20000   15554    2519    86%    /
/dev/dsk/usr   150000  118525   17019    87%    /usr
```

```
# df –t / /usr ↵
/    (/dev/dsk/root):      8893 blocks       4635 files
                 total:     40000 blocks       5760 files
/usr (/dev/dsk/usr ):     62951 blocks      33449 files
                 total:    300000 blocks      43200 files
```

```
# admfsinfo –o diskuse –l ↵
                 Free     Total    Pct     Free     Total    Pct
  Directory    Inodes    Inodes   Used   Blocks    Blocks   Used
------------  --------  --------  ----  --------  --------  ----
/                4633      5760   19%      8524     40000   78%
/usr            33449     43200   22%     62951    300000   79%
/usr/opt/X11    16875     20160   16%     12954    140000   90%
```

```
# du –s dgux ↵
6181    dgux
```

Network

```
# ifconfig cien0 ↵
cien0: 128.999.2.1 flags=443<UP,BROADCAST,RUNNING,STARTED>
          broadcast=128.224.2.255  netmask=0xffffff00  metric=0
```

```
# netstat –i ↵
Name  Mtu   Network        Address    Ipkts   Ierrs Opkts   Oerrs Collis
loop0 4136  loopback–net   localhost    1327   0     1327    0     0
inen0 1500  spin–lan       monarchy  3880485   0     2879318 0     13031
```

```
# ping natasha ↵
natasha is alive
```

```
# arp natasha ↵
ARP entry for natasha found in inen0:
Hostname   Internet Address   Hardware Address           Status
natasha    128.999.10.33      08:00:1b:20:06:b7        temporary
```

For a host on your subnet that you have just connected with (using the **ping** command, for example), the **arp** command reports both an Internet and Ethernet address.

```
# arp monarchy ↵
Currently no ARP entry for monarch (128.222.1.3)
```

For either a host on your subnet that you have not recently connected with or a host that is not on your subnet, the **arp** command reports an Internet address only.

```
# domainname ↵
ux_kernel
```

Packages

```
# admpackage –o list –v ↵
dgux      DG/UX Operating System  Release 4.00 February 1994
dgux      DG/UX Operating System   Release 4.10  August 1994
x.man     DG/UX Op Sys manual pages   Release 4.0  February 1994
dgux.man  DG/UX Op Sys manual pages   Release 4.10  August 1994
frame     FrameMaker               Release 3.1        March 1992
gcc       DG/UX Gnu C              Release 2.4.5.6 February 1994
gcc       DG/UX Gnu C              Release 2.5.8.1   August 1994

# admpackage –o list –c homer ↵
networker
```

Files that contain configuration data

Table 2–2 summarizes files that contain useful information about your configuration.

Table 2–2 Files that contain information about your system

File name	Description
/**dgux**	Link to kernel executable; not to be read or edited
/var/Build/dgux.*hostname*	Kernel executable; not to be read or edited
/var/Build/system.*hostname*	System configuration file generated from the **/var/system.params.***hostname* and **/var/system.device.***hostname* files; not to be edited
/var/Build/system.params. *hostname*	Kernel parameters file that you may edit
/var/Build/system.device. *hostname*	Kernel devices file that you may edit
/usr/src/uts/aviion/Build	Directory that is a link to **/var/Build**; not to be read or edited
/dev/pdsk (block mode) **/dev/rpdsk** (character mode)	Directory containing disk drive nodes; readable
/dev/rmt (character mode)	Directory containing tape drive nodes; readable
/dev/dsk (block mode) **/dev/rdsk** (character mode)	Directory containing virtual disk nodes and names; readable
/etc/inittab	Specifies the processes to be invoked at each run level
/etc/rc*x***.d**	Contains rc scripts that execute at different run levels, where *x* identifies the run level, that you may edit
/etc/fstab	File systems mounted at run level 1 or higher; readable, but not to be edited

Continued

Table 2–2 Files that contain information about your system

File name	Description
/etc/mnttab	File systems that currently are mounted; readable, but not to be edited
/etc/exports	File systems that are exported (accessible by remote computers attached to a network) at boot time; readable, but not to be edited
/etc/xtab	File systems that currently are exported; readable, but not to be edited
/etc/hosts	Host and Internet address pairs; readable
/etc/issue	Current operating system revision level; readable

You may locate these files and read them, if allowed, on your system.

End of Chapter

3 Maintaining sufficient file system space

Having free file system space on a disk prevents degraded system performance and allows you to continue creating and writing files in the file system. You are alerted to a space problem with these error messages:

```
file system full
no space left on device
out of contiguous disk space
```

File space is consumed by users as well as by processes that are part of application programs and DG/UX system services. Sometimes interim or temporary files created by a user or process are not deleted. In some cases, processes create and add messages and information to log files that, over time, grow in size.

When a file system becomes 80% full, I/O performance begins to decline. When a file system becomes 90% full, only the user with appropriate privilege can perform operations that involve allocating more space in the file system, such as creating new files or directories or increasing the size of a file. Any process without appropriate privilege that tries to create a file or write to a file in a 90%-full file system will fail with an error.

> **IMPORTANT** You may adjust the default 10% file system overhead that gets allocated to only users with the appropriate privilege when you create the file system with the **mkfs** command with **–m** free space option. For information on changing the file system overhead value, see the **tunefs**(1M) manual page.

Sufficient disk space is also necessary for a successful operating system upgrade. If you do not have enough space, the upgrade will fail. Although the upgrade procedure contains a file system cleanup script, the process will go faster if you have sufficient disk space before you start.

Therefore it is important to periodically:

- Ask all users to delete any of their files that they do not need. Copy a file that might be needed at a later date onto some other medium before deletion.
- Monitor disk space with the **ls –l**, **admfsinfo, df**, and **du** commands.
- Clean up the log files that are created and added to by processes. For more information on log files, see Chapter 5.
- Review the contents of the system temporary file directories and delete unnecessary files. For information on file system cleanup, see "Cleaning up files to increase free space" in this chapter.

Finding out the sizes of files in a directory

An easy method to check sizes of files in a directory is to perform a long listing of a directory with the **ls** command. An example follows:

```
# ls -l ↵
total 1186
-rw----r--   1 leary    doc         8686 May 11  1992 92-perform
-rw----r--   1 leary    doc         5525 May 21  1993 93-perform
-rw-rw-r--   1 leary    doc          595 Aug 10 11:47 Intel-SCSI
-rw----r--   1 leary    doc         2281 Mar 15  1995 anotherform
-rw----r--   1 leary    doc         3835 Jan 17  1995 timecards
-rw-------   1 leary    doc        33125 Jun 19  1992 taxes
-rw----r--   1 leary    doc          313 Nov 14  1994 notes
-rw----r--   1 leary    doc       101392 Jan 10  1995 lasttry
-rw-------   1 leary    doc          884 Sep 30  1992 dreams
```

The fifth column in the output shows file length in bytes. Comparing file sizes, you can determine which ones are good candidates for deletion (using the **rm** command) or moving elsewhere (using the **mv** command). Simple mathematics shows that the file named **lasttry** is almost 324 times larger than the file **notes**.

Finding out the number of used and free blocks

To find out the number of blocks used and free and the number of inodes (file slots) used and free, use the following command format:

admfsinfo -o diskuse -l *file-system*

Example:

```
# admfsinfo -o diskuse -l / ↵
```

IMPORTANT An easy way to check free space automatically is to insert the following line into your startup file, either **.login** for C shell users and **.profile** file for Bourne and Korn shell users:

```
admfsinfo -o diskuse
```

This command lists the number of inodes (file slots) and blocks in the / file system. As the percentage of blocks used approaches 80 percent, you must perform cleanup. An excerpt of typical output follows:

```
                Free     Total    Pct     Free     Total    Pct
 Directory     Inodes    Inodes   Used   Blocks    Blocks   Used
-----------  --------  --------  ----  --------  --------  -----
/               4624      5760   19%      8971     40000    77%
```

You may either clean up these file systems or expand them. To find out what files in / can be deleted, use the **admfsinfo** command shown in the sections titled “Finding out the sizes of files in a file system” and “Identifying large files.”

You may expand the **root** (/) or **/usr** file system only if the physical disk housing each contains sufficient space for another partition. You need to be able to boot from either of these file systems, and you cannot boot from a file system that spans multiple physical disks. You may perform this operation while the **/** and **/usr** file systems are mounted and in use.

For how to increase or change the size of a DG/UX file system, see *Managing Mass Storage Devices and DG/UX® File Systems*.

Finding out the amount of file system free space

To find out how much free space a file system has, use either of the following command formats:

df –k [*file–system ...*]

df –t [*file–system ...*]

The **–k** option reports free space in kbytes. Ten percent more space is reserved for only a user with the appropriate privilege.

The **–t** option reports free space in blocks, and includes the 10 percent extra space allocated for only a user with the appropriate privilege.

Examples:

```
# df –k / /usr ↵
filesystem     kbytes    used   avail capacity  mounted on
/dev/dsk/root   20000   15554    2519    86%    /
/dev/dsk/usr   150000  118525   17019    87%    /usr
```

IMPORTANT To convert kbytes to blocks, multiply by 2. 20,000 kbytes equal 40,000 blocks.

```
# df -t / /usr ↵
/    (/dev/dsk/root):      8893 blocks       4635 files
               total:     40000 blocks       5760 files
/usr (/dev/dsk/usr ):     62951 blocks      33449 files
       total:    300000 blocks      43200 files
```

This display shows that the **/** and **/usr** file systems are approaching fullness. You either may clean up these file systems or expand them. To find out what files in / can be deleted, use the **du** command covered in the section titled "Finding out the sizes of directories in a file system" and the **admfsinfo** command in the section titled "Identifying large files."

You may expand the **root** (/) or **/usr** file system only if the physical disk housing each contains sufficient space for another partition. You need to be able to boot from either of these file systems, and you cannot boot from a file system that spans multiple physical disks. You may perform this operation while the **/** and **/usr** file systems are mounted and in use.

For how to increase or change the size of a DG/UX file system, see *Managing Mass Storage Devices and DG/UX® File Systems*.

Finding out the sizes of directories in a file system

To display the number of blocks occupied by all of the files (and recursively, all of the subdirectories) in a file system, use the following command format:

du *file-system*

Example:

```
# du /usr | egrep '[1-9][0-9][0-9][0-9]+' > /tmp/usr ↵
```

The **du** (disk usage) command reports the names of directories in the specified file system that are at least 1000 blocks in size. The quoted string specifies a search pattern that is equal to or greater than four digits in length. The names of these directories are redirected to the file **/tmp/usr** file. You may then examine these directories for possible cleanup. An excerpt of typical output follows:

```
2140    /usr/local/man/man1
1684    /usr/local/man/man3
1427    /usr/local/man/mann
5719    /usr/local/man
3590    /usr/local/doc
73435   /usr/local
3221    /usr/lost+found
833080  /usr
```

The **du** and **egrep** commands help you locate large files. You may change the search string to locate the desired directory sizes. Each set of brackets represents a single digit. Be sure to delete the **/tmp/usr** file after you finish using it.

IMPORTANT Using the **du** command with no arguments reports the sizes of all files in the current directory.

Using the **du –s** command reports the total size of the current directory.

Identifying large files

To identify large files for possible deletion, use the following command format:

admfsinfo –o find –b *bytes* **–s size –f** *file–system*

IMPORTANT To convert bytes to blocks, divide by 512.

Example:

```
# admfsinfo –o find –b 100000 –s size –f / > /tmp/root & ↵
```

This command finds all files in the **/** file system that are at least 100,000 bytes in length and sorts them according to size, from largest to smallest. The names of these files are directed to the file **/tmp/root**. You may then examine these files for possible cleanup. The following example shows an excerpt of typical output:

```
Owner Size     Last Accessed  File Name
      (in bytes)
----- -------- -------------- ---------
root   3407270 11/06/94 16:26 /dgux.installer
root   3348075 03/03/95 12:37 /dgux
root   3348075 03/03/95 12:37 /dgux.homer
root   3348069 11/07/94 09:37 /dgux.final4
homer   627550 03/27/95 16:51 /tmp/mbox.homer
root    318716 03/17/95 15:02 /sbin/su
bin     250996 03/14/95 10:22 /sbin/fsck
bin     242428 03/20/95 11:32 /sbin/mount
bin     217684 03/27/95 17:19 /sbin/sh
root    191436 11/06/94 16:28 /sbin/ttymon
homer   167677 02/10/95 13:26 /lost+found/#24305
homer   163840 01/11/95 11:37 /tmp/sadm_6161.tmp
bin     160900 03/20/95 11:30 /sbin/umount
root    150632 01/11/95 11:39 /var/adm/streams/error.11–22
root    130420 11/06/94 17:57 /var/ftp/bin/ls
root    111084 11/06/94 16:29 /sbin/halt
root    111084 11/06/94 16:29 /sbin/reboot
bin     105204 03/14/95 10:21 /sbin/init
```

The **admfsinfo** command reports names of files, ordered from largest to smallest, that are equal to or greater than 100,000 bytes (or approximately 195 blocks). You may change the value in the **–b** field to locate the desired file size.

Verify all files before you delete them. From the preceding list, as a user with the appropriate privilege, you can delete the following files:

```
root   3348069 11/07/94 09:37 /dgux.final4
homer   167677 02/10/95 13:26 /lost+found/#24305
homer   163840 01/11/95 11:37 /tmp/sadm_6161.tmp
root    150632 11/22/95 11:39 /var/adm/streams/error.11–22
```

After verifying which kernel is linked to **/dgux**, you may delete the unlinked one; for example, **/dgux.final4**. You may also delete orphan file **/lost+found/#24305**, which was disconnected from its host file. The file **/tmp/sadm_6161.tmp** is a **sysadm** crash file that was generated when the system crashed. If you are not running **sysadm** at the moment, you may delete this file. The **/var/adm/streams/error.11–22** file may also be deleted because the the November 22 date is old. Deleting these files increases free space by 7,480 blocks, computed as follows:

```
The addition of file sizes (in bytes) in the second column
equals 3,830,218 bytes.

3,830,218 bytes / 512 bytes-per-block = 7,480.89 blocks
```

After you increase free space, check the increased percentage of free space with the **admfsinfo** and **df** commands. Be sure to delete the **/tmp/root** file after you finish using it.

Checking for core files

If an application or system utility aborts, it creates a core dump file in the current directory, named **core**. If you encounter a file named **core**, verify that it is, indeed, a core file with this command:

```
# file core ↵
core:core file
```

The output verifies that the file named **core** is a core file. Such a file name could have been identified as an ascii text file.

Use the following command to search for core files throughout your file system:

```
# find / –name core –print > /tmp/root & ↵
```

Be sure to run this command in background mode because it may take a long time to run.

Typical output follows:

```
-rwxr-xr-x   4 bin      bin    3584 Nov  6 17:55 core
```

Document the core dump's location, date, and time in your system administrator's records. Core dumps may be sporadic or regular. If you determine that a core dump is sporadic, you may delete it to reclaim space. If it regularly is written to the same directory, do not delete the core file. If the core file has a current date, preserve the file by moving it to another location (the **mv** command automatically deletes the file from its original location).

Use the **what** command to determine what application or system utility is creating the core file.

```
# what core ↵
core:
        tar.c,v        6.4.1.3
       DG/UX_5.4R3.10__3.0-3.0
        tar.c,v        6.4.1.3
       DG/UX_5.4R3.10__3.0-3.0
        AViiON DG/UX libdgc.so release 5.4R3.10__3.0-3.0
        AViiON DG/UX libc.so release 5.4R3.10__3.0-3.0
        AViiON DG/UX libc.so.1 release 5.4R3.10__5.3-5.0
        AViiON DG/UX libdgc.so.1 release 5.4R3.10__5.3-5.0
        AViiON DG/UX libdgc.so.1 release 5.4R3.10__5.3-5.0
```

The first line in the output shows that **tar.c** dumped core. The remaining listed items, which are shared libraries that are linked in at run time, are used by the **tar** command.

If a third-party application creates a core file, depending on your service agreement, contact either that vendor or Data General for help. For a Data General application or utility creating a core file, if you are covered by either a Support Plus, PASS, or OMNiiSERVICE support service, report the problem to the Customer Service Support Center. Otherwise, file an STR and write the core file to tape for submission to the Customer Support Center for diagnosis (see Chapter 9).

Cleaning up files to increase free space

Cleaning up files means either to delete or truncate them to increase free space. Truncation means deleting some or all of the file's contents, but retaining the file. You typically truncate log files, which are discussed in the next chapter. Look for the following files to clean up.

- Large files in **/tmp**, **/var/tmp**, or **/usr/tmp**

  Use the l**s –l** command to check the sizes of files.

  The easiest way to keep temporary file directories clean is by running a janitor-type job regularly using the **cron** utility. The prototype **crontab** file for **root** includes jobs that clean out **/tmp** and **/var/tmp** periodically. For more information on the **cron** utility and the prototype **root** jobs, see *Managing the DG/UX® System*.

- Saved editing sessions, such as files in directories beneath **/var/preserve**

- Extra kernels and system files (whose names start with **dgux**) in **/** and **/var/Build**

  You may have multiple kernels resulting from several kernel building sessions. The default kernel is the one linked to the DG/UX system, file **/dgux**, which you must keep. You can find out which kernel is linked to **/dgux** by using the following shell command:

```
# ls  -i  /dgux* ↵
4051 /dgux
4049 /dgux.installer
4058 /dgux.final4
4051 /dgux.homer
```

  The output shows that the inode numbers match for **/dgux** and **/dgux.homer**. (An inode is a data structure containing information about a file such as file type, size, date of creation, owner ID, and group ID.) You must keep **dgux.homer**, which is the name of the file linked to file **/dgux**. You may delete **/dgux.final4**. Do not delete **/dgux.installer**, which is needed for error recovery.

- Out-of-date site-supplied executables in directories such as **/local** or **/usr/local**

- Any files unrelated to the operating system

- Obsolete files in the **/usr/admin** and **/usr/lib/gcc–1** directories

IMPORTANT The **/usr/lib/gcc–1** directory does not exist on Intel platforms.

- Failed mail messages that you sent named **dead.letter**
- **/var/mail/***username*
  You may remind the mailbox owners to clean out their mailboxes. You may delete mail accounts for unknown users or users who no longer receive mail on your system.
- Core files
  If an application or system utility aborts, it creates a core dump file, named **core**, in the current directory. For more information on core files, see the section "Checking for core files" in this chapter.
- **lost+found** files in each file system, such as **/** and **/usr**

  A file in **lost+found** is a file fragment that was disconnected from its host file during abnormal conditions such as a halt, shutdown, or malfunctioning application or system facility. For information on **lost+found** files, see Chapter 10.
- Large log files

  For information on checking and cleaning log files, see Chapter 5.
- Large text files can be compressed, saving between 50–60% disk space. See the **compress**(1) manual page for details.
- PostScript® files consume a lot of disk space. You may delete them, since they can be easily re-generated, or archive them.
- Files appended with **.tmp** or **.foo** strongly suggest their transitory nature. You may wish to delete them.

End of Chapter

4 Developing a policy for handling halts

Upon detecting a halt condition, the DG/UX kernel halts all activity on the system and displays a message like the following at the system console:

```
The operating system has detected a serious error and halted.
Please record the following halt code:
"DG/UX R4.10 halt code 2000075"
Do you want to take a kernel memory dump [Y]?
```

During normal system operation, and prior to a halt situation, you must define a policy for handling halts. Such a policy will define a default dump destination device and whether to take a dump automatically upon a halt condition.

The DG/UX system offers these facilities for controlling the system's halt behavior:

- the **dg_sysctl** command
- the watchdog kernel parameter, **wdt()**
- the System Control Monitor (SCM)
- NVRAM database (**NDB**) that is configured through the **admnvram** command

  IMPORTANT DG/UX for Intel systems do not have access to the SCM. Instead, it uses the **NDB**, which is a collection of system parameters used by the DG/UX kernel and bootstraps. Both versions of the DG/UX system, however, use the **dg_sysctl** command and the watchdog kernel parameter.

For information on how to recover from a halt, see Chapter 9.

Checking and changing default halt behavior

You can check the current default system halt behavior using the **dg_sysctl** command. An example follows:

```
# dg_sysctl ↵
AUTOBOOT=halt
BOOTPATH=sd(cisc(0),3,0)root:/dgux
AUTODUMP=ask
DUMPDEV=st(cisc(0),4,0)
DUMPLEVEL=kernel
POWEROFF=skip
```

Refer to the **dg_sysctl**(1M) manual page for more information on the **dg_sysctl** command.

The SCM provides two parameters that control halt behavior: Autoboot and Boot path. If you use both the SCM and **dg_sysctl**, make sure these settings are the same for both.

See your hardware documentation for complete information on the SCM.

To view or change parameters through the SCM, use the **f** (format) command.

```
SCM> f ↵
```

You can set the default boot path using the **admnvram** command format as follows:

admnvram –o set –f '*NDB-device***' –p boot_command_1 =** *bootpath*

where:

–f '*NDB-device*' specifies the name of the device that contains the NDB.

–p boot_command_1 = *bootpath* specifies the device containing the bootstrap which specifies the kernel and the boot options to be loaded.

An example follows:

```
# admnvram –o set –f 'sd(npsc(pci(0),6))' –p boot_command_1 =
'sd(npsc(pci(0),6)) root –f /dgux –i 3'↵
```

This command boots the kernel from **sd(npsc(pci(0),6))** to init 3.

See the **admnvram**(1M) manual page for details on the **admnvram** command.

You can configure the watchdog timer (**wdt**) in the kernel. When the watchdog timer detects a system hang, it automatically halts the system. For AV 8500 and AV 9500 systems with supporting IOC hardware, an automatic dump is generated, using the parameters set through the **dg_sysctl** command.

To enable this feature, ensure that the kernel contains the **wdt()** pseudo device as explained in the section "Enabling the watchdog timer." Also, set up the failover monitor for the software watchdog timer as explained in *Managing the DG/UX® System*.

You may also define an autoboot device either during DG/UX system installation or later with the **dg_sysctl** command.

Setting up an automatic stop with no dump

You can set up an automatic system stop with no operator intervention and no system dump as the default system recovery behavior upon detection of a halt with the following command:

```
# dg_sysctl –d skip –r halt ↵
```

The **–d skip** option tells the system to skip the generation of a dump; the **–r** option sets up the system to stop (with the **halt** command) after a halt.

Setting up an automatic system dump

If you set your system to take a system dump by default upon detection of a halt, you can select either an automatic or interactive dump.

An automatic dump allows you to reduce the recovery time after a halt, but it requires a dump destination device that is ready at all times to receive the system dump in the event of a halt.

Use the following command to enable the automatic dump feature:

```
# dg_sysctl  –d  auto ↵
```

To set up your system to generate a system dump without operator intervention and then stop the system, use the following command, as an example:

```
# dg_sysctl –d auto –f “st(cisc(),5)” –r halt ↵
```

To set up an automatic system dump to a virtual disk, see the section “Dumping to a virtual disk” later in this chapter.

Setting up an interactive system dump

An interactive dump is your system’s default response to a halt. You can set this behavior explicitly by using the following command:

```
# dg_sysctl  –d  ask ↵
```

When configured to perform an interactive dump, the system responds to a halt by displaying the halt code and then the following prompt:

```
Do you want to take a system dump? [Y]
```

If you press Enter, the system prompts for the type of dump and the dump destination device, described in the following sections.

Selecting an entire memory or kernel dump

You can dump either the entire memory of your system or just the kernel memory. A dump of kernel memory is sufficient to diagnose a hang or halt unless the Customer Support Center tells you otherwise. The kernel memory dump is the default type, is faster, and requires about half the space of a complete dump.

To change the default type of dump to a complete dump, use the following command:

```
# dg_sysctl -l all ↵
```

To change the type of dump from a complete to only the kernel image, substitute **kernel** for **all** in the **dg_sysctl** command.

Using Full Memory Dump

The AV 20000 computer supports memory up to 32 GB; dump support is available for this large memory size. If your Data General customer service representative requests a full memory dump (kernel and user memory), this enhanced dump support is available.

If you must perform a full memory dump, consider the following:

- You must have the physical space to hold the physical memory that results from a large system dump.
  - When dumping to a tape drive, you may need multiple cartridges.
  - When dumping to a disk, you must have enough space on one disk; this space should be contiguous if possible.
- Taking a large system dump is very time–consuming. The amount of time required depends on your system's transfer rates and the total amount of physical memory involved. However, even dumping to a disk, which is usually quick, could require an hour or more for the largest memory configurations.

Selecting a tape dump destination device

The dump destination is the tape device, virtual disk, or network interface to which you want the dump written. By default, the device is the value of the DUMP tunable parameter configured in your kernel. You can override the DUMP parameter setting with the **dg_sysctl** command; for example:

```
# dg_sysctl -f "st(ncsc(),5)" ↵
```

You can combine the **–d** and **–f** options to set the automatic dump feature and the dump destination device in a single command line; for example:

```
# dg_sysctl -d auto -f "st(cisc(),4)" ↵
```

Dumping to a virtual disk

You may select a dump virtual disk during DG/UX system installation or during normal operation.

A system with a local disk can dump to a local virtual disk instead of a tape. The advantage of dumping to disk is that it is faster than dumping to tape, resulting in decreased down-time. The disadvantage is that you must reserve for this purpose a virtual disk large enough to contain the system dump image.

> **IMPORTANT** You can dump to a virtual disk only if it resides entirely on a local SCSI disk. You cannot dump to a virtual disk that comprises multiple partitions spanning multiple physical disks or to any virtual disk residing on an SMD or ESDI physical disk, or to virtual disk mirrors. You can store halt dumps on virtual disks that are unstriped aggregations of simple partitions if all pieces, up to a total of 8, are on the same physical disk.

During DG/UX system installation, you may specify a virtual disk dump device to capture a kernel-only memory dump. See *Installing the DG/UX® System* for details on specifying a dump device. Following installation, you may increase the dump disk allocation. Table 4–1 contains information that helps you calculate dump disk size.

Table 4–1 Calculating dump disk size

Type of dump	Memory size	Dump disk size
Kernel-only	Small (32-Mbyte)	1/2 memory (16-Mbyte)
Kernel-only	Large (1-Gbyte)	1/4 memory (256-Mbyte)
Complete memory	Small and large	memory + 5% small: (33.6-Mbyte) large: (1.05-Gbytes)

During normal system operation, you may create the dump disk with the **sysadm** operation `Device –> Disk –> Virtual –> Create`. We recommend that you assign a meaningful name to the virtual disk, such as **sys_dump**. The dump process will overwrite any data, such as a file system, that resides on the virtual disk at the time of the dump. Therefore, you should not create a file system on the disk and attempt to use it for any purpose other than to contain the dump.

> **IMPORTANT** If you specify a dump device during installation, that value automatically will override any value that has already been assigned through the **dg_sysctl** command.

After creating a virtual disk to be used as a dump disk, use the following command syntax to designate it as the dump device.

 # dg_sysctl –f "vdm_dump(*physical_disk_name,virtual_disk_name***)"**

An example follows:

 # dg_sysctl –f "vdm_dump(sd(cisc(0),1,0),sys_dump)" ↵

By default, the system displays a prompt at the system console before writing to the dump virtual disk, allowing you to change the virtual disk or a tape drive. When the system halt procedure prompts you for the dump device, specify the physical and virtual disks using the following command syntax:

vdm_dump(*physical-disk-name,virtual-disk-name***)**

Setting up automatic reboot after a system halt

To minimize your system's downtime, you can set up your system to reboot without operator intervention. Set the automatic boot feature using these operations: the **dg_sysctl** command and the SCM menu or the **admnvram** command, as follows:

 # dg_sysctl –r auto ↵

SCM> **f** ↵

From the SCM main menu, select:

`Change default boot paths –> Auto–reboot/Boot on error –> Enabled`

Use the **admnvram** command to set the boot path in the **NDB**. This path will be used when the system reboots after the power-up diagnostics are run.

admnvram –o set –f '*NDB-device***' –p boot_command_1** = *bootpath*

where:

–f *'NDB-device'* specifies the name of the device that contains the NDB.

–p boot_command_1 = *bootpath* specifies the device containing the bootstrap which specifies the kernel and the boot options to be loaded.

An example follows:

```
# admnvram –o set –f 'sd(npsc(pci(0),6))' –p boot_command_1 =
'sd(npsc(pci(0),6))/dgux –i 3'↵
```

This command boots the kernel from **sd(npsc(pci(0),6))** to init 3.

See the **admnvram**(1M) manual page for details on the **admnvram** command.

> **IMPORTANT** Be sure to perform these operations through the **dg_sysctl** command and the SCM menu or the **admnvram** command, as appropriate, to ensure that the automatic reboot setting is consistent between both facilities. Failure to do both may result in conflicting settings, which could produce an unexpected result.

These operations configure your system to reboot whether or not a system dump is generated. By default, the automatic boot feature reboots the system using the most recently used boot command line initiated at the SCM or the **NDB** prompt before the halt occurred. For example, if you last booted **'sd(cisc(0),0,0) root –f /dgux –i 3'** , the system will reboot using this boot path.

To override this default behavior, use both the **dg_sysctl** command with the **–b** option and the SCM "Change default boot paths" menu or the **admnvram** command, as appropriate, to specify a different boot path.

> **IMPORTANT** Again perform the operation both through the **dg_sysctl** command and the SCM to prevent a possible conflict of settings.

For example, both the following **dg_sysctl** command line and a matching boot path selected through the SCM "Change default boot paths" option or the **admnvram** command sets up the system to send a system dump to the device designated by **–f st(cisc(),5)** and then boot from **–b 'sd(cisc(0),0,0) root –f /dgux –i 3' :**

```
# dg_sysctl –r auto –b 'sd(cisc(0),0,0) root –f /dgux –i 3' –d auto –f
'st(cisc(0),5'↵
```

This example enables auto-reboot (**–r auto**) of the kernel stored as **/dgux** on the **root** virtual disk stored on device **sd(cisc(0),0,0)** to a run level of 3 (**–i 3**) and auto-dumping after a halt (**–d auto**). The dump is stored on **st(insc(0),4)**, a SCSI tape device with a SCSI ID of 4 that is attached to the first (**0**) integrated Ciprico SCSI controller.

When you set up a system to create a dump automatically when a halt occurs, you must make sure that the defined dump device is always ready to receive a dump. Tape drives that are designated dump devices should always contain a tape to which a dump can be written.

```
# dg_sysctl -d auto -f "st(cisc(),5)" -r auto -b
"sd(cisc(),2)root:/dgux"
```

You can set up your system to send you mail every time it reboots. This capability is particularly helpful because it reports reboots that occur in your absence. To enable this feature, edit the **/etc/dgux.params** file. Set the **reboot_notify_START** parameter to true, and set the **reboot_notify_ARG** parameter to one or more local mail addresses. A notification message is sent to each user specified by the **reboot_notify_ARG** parameter each time the system boots.

Changing the default automatic reboot run level

IMPORTANT This operation works only if you define a default boot path for the machine.

When your system reboots automatically, it reboots to the default run level set in the **/etc/inittab** file. Initially the default run level is set to **s** (single-user mode). To change it, edit your **/etc/inittab** file and change this line:

```
def:s:initdefault:
```

so that the second field contains the desired default run level. For example, the following line makes run level 3 the default:

```
def:3:initdefault:
```

Power off policy

Power will continue to be supplied following a system failure. Automatic poweroff following a halt does not apply to an abnormal shutdown. The default value will always be to skip poweroff.

Using the watchdog timer to handle system halts

IMPORTANT The watchdog timer works on AViiON 450, 550, 4500, 5500, 8500, and 9500 systems, and on all AViiON systems that use the failover monitor **failovermon**(1M). For information on the failover monitor, see *Managing the DG/UX® System*.

The watchdog timer performs an automatic system reset upon detection of either a hardware or software system hang. A hardware hang produces these halt codes:

150000044
53000060
323000001

A software hang produces these halt codes:

323000002
53000061

These halt codes trigger a watchdog timer reset:

323000001
323000002
53000060
53000061
150000044

For halt codes 323000001, 323000002, 53000060, and 53000061, the current state of the job processors is not saved with a system reset, so a dump of the memory image is not generated. Halt code 150000044 halts the system and generates a dump of the memory image that can be copied to tape or disk for analysis.

On AV 8500 and AV 9500 systems, a reset starts power-up diagnostics and isolates and deconfigures any faulty field-replaceable hardware components when the system is rebooted. If AV/Alert (a comprehensive diagnostic support system for AViiON family hardware) is enabled, the faulty hardware is also reported automatically to the Customer Support Center.

If a 150000044 halt code was generated by a non-maskable interrupt and the autodump feature is enabled, the system will attempt to generate a system memory dump. If the autodump feature is not enabled, the system will prompt for a memory dump. After generating the dump, if the autoboot feature is enabled, the system reboots without user intervention. If the system does not respond to the non-maskable interrupt, the watchdog timer resets the system, starts power-up diagnostics, isolates, and deconfigures any faulty field-replaceable hardware components on reboot.

Enabling the watchdog timer

The watchdog timer feature is configured in the kernel by default.

CAUTION *Do not configure the watchdog timer on the systems listed below. Use of the wdt () device on these systems can cause a hang while configuring devices during system boot:*

- AV3600
- AV 3600R
- AV 3650
- AV 3650 R
- AV 4700
- AV 4800
- AV4900
- AV 5800
- AV 5900
- AV6600

See the **wdt()** manual page for more information.

To enable this feature, ensure that the kernel contains the **wdt()** pseudo device and set up the failover monitor for the software watchdog timer as explained in *Managing the DG/UX® System*.

To configure the watchdog timer without rebuilding the kernel and rebooting the system, you can dynamically configure the device following the **sysadm** path:

```
Device -> Configure
```

When prompted for the device to configure, specify the watchdog timer, **wdt()**. The watchdog timer will be configured for the duration of the current session. However, rebooting the system will erase this setting. Eventually, you must rebuild the kernel, incorporating the watchdog timer, and reboot the system.

Summary of methods for tuning halt behavior

Table 4–2 summarizes the methods for controlling your system's halt behavior.

Table 4–2 Summary of methods for tuning your system's halt behavior

Parameter	Value (Example)	Where to set
Autoboot	**auto**	**dg_sysctl** command and SCM menus or **admnvram** command
Bootpath	**sd(npsc(pci(0),6))/dgux -i 3**	**dg_sysctl** command and SCM menus or **admnvram** command
Autodump	**auto**	**dg_sysctl** command and SCM menus or **admnvram** command
Dump device	**vdm_dump(sd(ncsc(),1),sys_dump)**	**dg_sysctl** command and **sysadm**
Dump level	**kernel**	**dg_sysctl** command and **sysadm**
Power off state	**N/A**	**dg_sysctl** command
Auto-reboot/Boot on Error	**enabled**	SCM menus or **admnvram** command
Default boot path	**sd(npsc(pci(0),6))/dgux -i 3**	SCM menus or **admnvram** command

Continued

Table 4–2 Summary of methods for tuning your system's halt behavior

Parameter	Value (Example)	Where to set
Watchdog timer (wdt) configured	System –> Kernel –> Build	**sysadm**
Failover monitor set up	Availability –> Disk Failover –> Alternate Paths –> Add	**sysadm**

All values set through the **dg_sysctl** command and the SCM menus or the **admnvram** command, as appropriate, must be consistent.

End of Chapter

5 Log files

A log file collects messages describing errors, abnormal conditions, routine checkpoints, and status that are sent by various system programs. These programs send their log messages to either their own log files, such as **/etc/log/fsck.log**, or to a general system log named **/var/adm/messages**, which is managed by the system error logging facility **syslogd**.

Where a particular program sends its log messages depends on how the program is designed and whether it is configured to take advantage of **syslogd**. Some programs use **syslogd** but have their output redirected to a file other than **/var/adm/messages**. If a program uses **syslogd** in any form, its log file destination and other configuration information is listed in the **syslogd** configuration database **/etc/syslog.conf**. For information on **syslog** and the **syslog.conf** file, see the **syslog.conf**(5) and **logger**(1) manual pages.

For information on starting, tuning, and using **syslogd**, refer to Appendix A. Information on tuning a network logger is also contained in this appendix.

Monitoring and cleaning log files

It is a good practice to regularly monitor log files to verify your system's healthy operation. Examining logs may reveal a problem before it is manifested in an error message or in some other way. Identifying a problem early may prevent a serious problem later.

Besides monitoring the information in log files, you must also keep track of their sizes. Various system utilities send messages to these files continually, causing them to grow. Some utilities control the size of their logs and take care of their own cleanup, while others do not. To maintain sufficient free space, monitor the log files that require cleanup.

Table 5–1 lists the log files, describes them, tells you when to check them, and prescribes a method of cleanup. The log files with "none" in the "Cleanup" column take care of their own cleanup. If the column contains "Truncate," you must reduce the file's length. Some files require special cleanup, which is noted in the column. Special cleanup information is located at the end of the log table series.

IMPORTANT You must have appropriate privilege to truncate log files.

There are two ways to truncate a log file:

- Removing the file's content, retaining the file

  The advantage of this method is that it frees up space without deleting the file. The log file's inode information is preserved for use by the utility that logs to that file.

  To remove the contents of a file while retaining the inode, use the following sample command:

  ```
  # cat /dev/null > fred.log ↵
  ```

  The **cat /dev/null** command overwrites the existing log to produce an empty file.

- Trimming the file's content, retaining the file

  The advantage of this method is that it preserves the newer log information, deletes the older information, and preserves the inode.

  To trim a log file, use the following sample commands:

  ```
  # tail fred.log > /tmp/fred.log ↵
  # cat /tmp/fred.log > fred.log ↵
  # rm /tmp/fred.log ↵
  ```

  The **tail** command preserves the last ten lines of file **fred.log**, by default, directing the output to **/tmp/fred.log**. The **cat** command redirects the content of **/tmp/fred.log** back to **fred.log**. The final command removes the temporary log file. Refer to the **tail**(1) manual page for information on specifying the number of lines to preserve.

Table 5–1 lists the log files, specifies their locations, describes them, tells you when to check them, and prescribes a method of cleanup.

Table 5–1 Log files

Log name description	When to check	Cleanup
/etc/log/fsck.log		
Reports that file system can be mounted and requires no check, file system has recoverable problem, or file system is corrupt and cannot be mounted.	After attempt to access file system fails; after system boots; following run level change; after system crash or disk failure; if **root** is full and you need to free up root space.	truncate

Continued

Table 5–1 Log files

Log name description	When to check	Cleanup
/etc/log/fast_fsck.log		
Same as **fsck**, but is generated by fast version of file system checker.	After attempt to access file system fails; after system boots; following run level change; after system crash or disk failure; if **root** is full and you need to free up root space.	none
/etc/log/filesave.log		
Collects logging information for daily disk-to-disk backup performed with the **filesave** script and weekly disk-to-tape backup performed with the **tapesave** script. Both scripts use the **volcopy** command to perform backup. If you do not use **volcopy** for your backups, this log will not be used.	If **root** is full and you need to free up root space.	truncate
/etc/log/init.log		
Logs progress messages generated by rc scripts that run during boot process.	After system boots; after run level changes.	none
/etc/log/netinit.log		
Reports status messages associated with building network protocol stack on host with two network interfaces.	If a LAN controller does not appear to start correctly.	truncate

Continued

Table 5–1 Log files

Log name description	When to check	Cleanup
/etc/log/nfsfs.log		
Contains output from **mount** command, which shows whether attempts to mount directories failed or succeeded. When mount fails, includes error message. To diagnose network-related failures, see *Managing TCP/IP on the DG/UX® System* or *Managing ONC/NFS™ and Its Facilities on the DG/UX® System*.	After you mount or attempt to mount remote file systems.	none
/etc/log/preserve.rclock		
Serves as lock file, indicating to system software whether particular action was taken at boot time.	No checking is necessary.	none
/etc/lp/logs/lpNet		
Keeps track of print requests directed to remote host.	If users cannot print to remote host.	truncate
/etc/lp/logs/lpsched		
Records lpsched activities.	If having printing problems; if **root** is full and you need to free up root space.	truncate
/etc/lp/logs/requests		
Records information describing lpsched print requests.	If having printing problems; if **root** is full and you need to free up root space.	truncate
/var/adm/acct/nite/fd2log		
Collects error messages from accounting system.	If **root** is full and you need to free up root space.	truncate
/var/adm/acct/nite/wtmp.*MMDD*		

Continued

Table 5–1 Log files

Log name description	When to check	Cleanup
Contains processed version of **/etc/wtmp**, which is used to generate accounting reports.	If **root** is full and you need to free up root space.	truncate
/var/adm/acct/nite/wmtperror*MMDD*		
Collects errors and warnings generated when creating **wtmp.***MMDD*.	If **root** is full and you need to free up root space.	truncate
/var/adm/dgsvcmgr/dgsvcd		
Contains data logged by AV/Alert, including incident numbers assigned to each automatically logged message.	To track logged incidents.	none
/var/adm/log/backup.log		
Collects messages generated through the **sysadm** System –> Backup –> Create dump operation.	Following system backup to verify **dump2** command; to check for possible errors; if **root** is full and you need to free up root space.	truncate
/var/adm/log/idc.log		
Records **idc**(1) activity.	If **root** is full and you need to free up root space.	truncate
/var/adm/log/installman.log		
Collects administrative commands performed and messages generated during installation of DG/UX system.	Following DG/UX installation or upgrade; to verify installation; to check for errors if installation fails; if **root** is full and you need to free up root space.	truncate
/var/adm/messages		

Continued

Table 5–1 Log files

Log name description	When to check	Cleanup
Contains information about system logged by system error logging daemon **syslogd**. Collects informational and error messages for hardware failures, network problems, and daemon starts and stops.	Periodically as good system administration practice; to monitor failover startup or operation actions.	truncate
/var/adm/log/sysadm.log		
Collects administrative commands invoked by **sysadm** operations when verbosity is set to debugging information.	During normal operation, to learn **adm** command equivalents to **sysadm** operations, which may be useful for troubleshooting. Also useful for writing shell scripts to automate system operations; if **root** is full and you need to free up root space.	truncate
/var/adm/sa/*		
Collects information from system activity monitor, **sar** or **nsar.**	Since log data is in a raw format, you do not need to check this log.	none
/var/adm/spellhist		
Collects words not recognized by **spell** utility.	Periodically as a good system administration practice; if **root** is full and you need to free up root space.	truncate; Also see "Trimming the /var/adm/ spellhist file."
/var/adm/streams/error.*MMDD*		
Collects errors and warnings generated by STREAMS facilities.	When having network problems; if **root** is full and you need to free up root space.	See "Cleaning up /var/adm /streams files."
/var/adm/sulog		

Continued

Table 5–1 Log files

Log name description	When to check	Cleanup
Contains history of superuser (su) command usage. As security measure, file should not be readable by others.	To monitor use of **su** command for system security; if **root** is full and you need to free up root space.	truncate
/var/adm/utmp		
Contains information on system's run level. Various system services, such as login, maintain this information.	No checking is necessary.	none
/var/adm/wtmp		
Contains history of system logins and accounting information. This file contains record for each login that occurs.	No checking is necessary.	See "Trimming the /var/adm/ wtmp file."
/var/cron/log		
Contains history of all actions taken by /usr/sbin/cron.	If cron jobs do not start as expected; if cron exits unexpectedly; if root is full and you need to free up root space.	truncate
/var/lp/logs/lpNet		
Records activities of **lpNet**, which handles communication with LP systems running on other systems in network.	If **root** is full and you need to free up root space.	truncate; Also see "Cleaning the LP print service logs."
/var/lp/logs/lpsched		
LP print service keeps log for scheduler-related events.	If you have problems printing.	truncate; Also see "Cleaning the LP print service logs."
/var/lp/logs/requests		

Continued

Table 5–1 Log files

Log name description	When to check	Cleanup
Accumulates information about each completed **lp** request.	If you have problems printing.	truncate; Also see "Cleaning the LP print service logs."
/var/saf/tcp/log		
Collects actions taken by tcp listen port monitor.	If having printer problems; if having failover problems.	truncate
/var/saf/*pmtag*/log		
Collects actions taken by port monitor, *pmtag*.	If having problems with terminals, port monitors, or modems.	truncate
/var/saf/_log		
Records actions taken by **SAC** (Service Access Monitor).	If port monitors, such as the tcp listen port monitor and ttymon port monitors, have problems starting or stopping unexpectedly.	truncate
/var/spool/uucp/log		
Records actions taken by **uucp** facility.	If having problems with uucp; if root is full and you need to free up root space.	truncate; Also see "Cleaning the /var/spool directory."
/var/spool/uucp/log.old		
Created by cron job that performs uucp cleanup, it contains the same type of information as the **/var/spool/uucp/log**, but from an earlier time period.	If having problems with uucp; if root is full and you need to free up root space.	truncate; Also see "Cleaning the /var/spool directory."
/var/setup.d/log/dgux.root		
Contains detailed account of **root** setup.	Following installation or upgrade of this product.	truncate
/var/setup.d/log/dgux.usr		

Continued

Table 5–1 Log files

Log name description	When to check	Cleanup
Contains detailed account of **usr** setup.	Following installation or upgrade of this product.	truncate
/var/setup.d/log/networker.root		
Contains detailed account of **networker** setup in **root**.	Following installation or upgrade of this product.	truncate
/var/setup.d/log/networker.usr		
Contains detailed account of **networker** setup in **usr**.	Following installation or upgrade of this product.	truncate
/var/setup.d/log/nfs.root		
Contains detailed account of **nfs** setup in **root**.	Following installation or upgrade of this product.	truncate
/var/setup.d/log/onc.root		
Contains detailed account of **onc** setup in **root**.	Following installation or upgrade of this product.	truncate
/var/setup.d/log/tcpip.root		
Contains detailed account of **tcpip** setup in **root**.	Following installation or upgrade of this product.	truncate
/var/setup.d/log/tcpip.usr		
Contains detailed account of **tcpip** setup in **usr**.	Following installation or upgrade of this product.	truncate

Log files that need special cleanup

The following log files require special cleanup procedures, which are covered in the remaining sections in this chapter:

/var/adm/spellhist
/var/adm/wtmp
/var/lp/logs/*
/var/spool/*
/var/adm/streams/*

Trimming the /var/adm/spellhist file

If users on your system use the **spell** utility, occasionally check the **/var/adm/spellhist** file to make sure it does not consume excessive space. The **spellhist** file contains words not recognized by **spell**. If your users have no use for the **spellhist** file, you may delete it; however, if they like to track words that appear there, you can at least reduce the size of the file by using **sort** to order file entries and remove duplicate entries. The following example demonstrates:

```
# cd /var/adm ↵
# sort –u spellhist > /tmp/spell ↵
# cat /tmp/spell > spellhist ↵
# rm /tmp/spell ↵
```

The **sort** command sorts the **spellhist** file, eliminating duplicates, and directs the output to **/tmp/spell**. The **cat** command redirects the content of **/tmp/spell** back to **/var/adm/spellhist**. The final command removes the temporary file. Refer to the **sort**(1) manual page for information on specifying different sort keys.

Trimming the /var/adm/wtmp or /var/adm/wtmpx file

When the **wtmp** or **wtmpx file** becomes too large, you may either delete its content, leaving the file, or trim it, leaving only some of the more recent entries.

If you use accounting on your system, realize that removing or reducing this file removes information required by the accounting system to charge for connect time. For more information on the accounting system, see *Managing the DG/UX® System*.

If you are using an m88k system and you trim the file, be aware that this file is a data file (not a text file) made up of 64-character records, and you should not edit it with a text editor. Instead, use the **tail** command to extract entries from the end of the file, the goal being to replace the **wtmp** file with these final entries. Be careful to specify in the **tail** command a number of characters that is divisible by 64, which is the size of a file entry. If any entry in the new **wtmp** file is incomplete, some commands may fail.

For example, to trim the **wtmp** file, leaving only the last 3200 characters, issue these commands:

```
# tail -3200c /var/adm/wtmp > /tmp/wtmp ↵
# cat /tmp/wtmp > /var/adm/wtmp ↵
# rm /tmp/wtmp ↵
```

The **tail** command preserves the last 3200 characters of file **/var/adm/wtmp**, directing the output to **/tmp/wtmp**. The **cat** command redirects the content of **/tmp/wtmp** back to **/var/adm/wtmp**. The final command removes the temporary file.

For ix86 systems, you will generally use the **wtmpx** and **utmpx** files, rather than the **wtmp** and **utmp** files mentioned earlier for m88k systems. The **/var/adm/wtmpx** file record size is 372 bytes in length. Therefore, to properly truncate, you must use a multiple of 372 when specifying the **tail** command.

For example, to trim the **wtmpx** file, leaving only the last 18600 characters of the file, or 50 entries, issue these commands:

```
# tail -18600c /var/adm/wtmpx > /tmp/wtmpx ↵
# cat /tmp/wtmpx > /var/adm/wtmpx ↵
# rm /tmp/wtmpx ↵
```

The **tail** command preserves the last 18,600 characters of file **/var/adm/wtmpx**, directing the output to **/tmp/wtmpx**. The **cat** command redirects the content of **/tmp/wtmpx** back to **/var/adm/wtmpx**. The final command removes the temporary file.

For more information about the **wtmp** and **wtmpx** files and the **tail** command, see the **wtmp**(4) **wtmpx**(4) and **tail**(1) manual pages.

Cleaning the /var/spool directory

Occasionally, check the **/var/spool** directory, which contains directories supporting a number of system services. For example, this directory contains the files and directories supporting the printer (LP) services. An interrupted LP command could abandon a file in the LP requests directory, for example. If such an accident happens often enough, you could waste valuable disk space. When you find lost print jobs, you can delete them or save them for the user who submitted the print job.

Cleaning the LP print service logs

The LP print service has several logs that require occasional trimming: in **/var/lp/logs**, they are **lpNet**, **lpsched**, and **requests**. The LP print service's prototype **crontab** file, **/admin/crontabs/lp.proto**, contains jobs to prevent unlimited log growth. To use them on your system, the user with the appropriate privilege may execute **crontab –e** to edit the **crontab** file. Then add the jobs from the **lp.proto** file. For information on **cron**, see *Managing the DG/UX® System*. For information on the LP print service logs and the **lp.proto cron** jobs, see *Installing and Managing Printers on the DG/UX® System*.

Cleaning up /var/adm/streams files

Delete all error log files in the **/var/adm/streams** directory of a specified age with the following command format:

admstrlog –o delete –a *age–in–days*

You may issue this command from the shell or as a **cron** job. Alternatively, you may use the following **sysadm** operation:

```
Logging –> Network –> Delete
```

Specify the age in days.

Most of the messages contained in these files are informational — when interfaces are stopped, started, attached, detached, and when the kernel intervenes to terminate a connection when multi-path LAN I/O has switched from one interface to another.

End of Chapter

6 Performance problems

Each command and program that runs on the DG/UX system at a particular time is called a process, which is assigned a PID (process identifier). When processing demands approach or exceed the CPU resources available, the system overloads and performance degrades. For details on monitoring DG/UX processes, refer to *Managing the DG/UX® System* and *Analyzing DG/UX® System Performance.*

Three process problems can impede performance:

- hung process
- CPU hog
- memory leak

A hung process is an annoyance, whereas CPU hogs and memory leaks may lead to serious performance problems.

Being conservative with system resources

Here are a few tips for keeping your system in good trim:

- Keep your windows-based programs such as screen savers, clocks, and games to a minimum. When performance starts to degrade, quit programs and restart them when performance improves.
- Log out when you are not actively using the system.
- Keep the number of windows on your screen to a minimum.
- Exit programs completely, including those that are iconified, when you're not using them.
- Keep your mail messages to a minimum. Lengthy mail folders add start-up time to the mail program when invoked.

Hung process

A hung process is an abnormal operation that no longer responds to requests from its controlling program or terminal. Although a hung process does not consume resources, it serves no purpose except to tie up a terminal or a window. You may suspect that a process is hung when you see that nothing is happening. Use the **nps** command to check the status of your active processes. An example follows:

IMPORTANT We encourage you to use the **nps** rather than the **ps** command to check process status. These commands produce the same type of output, but some fields are interpreted differently. Do not use the **nps** and **ps** commands interchangeably. For detailed information on these commands, refer to the respective manual pages.

```
# nps –ostate, uname, pid, ppid, stime, time, cmd ↵
S   UNAME   PID   PPID STIME    TIME CMD
W   homer   5173  5172 09:42:20 0:01 pshell
Z   homer   5177  5173 09:42:20 0:00 android
R   homer   5211  5173 09:42:20 0:00 nps
```

Look for the appearance of "Z," for zombie (also referred to as terminated or defunct), in the leftmost column marked by "S" for process state. Z marks the hung process, **android**, that has already been killed. However, the zombie process is still listed in **nps** output because it is waiting for its parent process to accept delivery of a termination message. Since the parent is not accepting the child's exit status, we can conclude that the parent process, **pshell**, contains a bug.

In this example, the **pshell** command is waiting "W," the **android** command is trying to die "Z," and the **nps** command is running "R." The process identifiers that correspond to each of these processes appear in the column marked by "PID." Refer to the **nps**(1) manual page for details on interpreting each output field.

Although the zombie process has already been killed, it still consumes minimal resources. To recover from this problem, kill the **pshell** parent process. Killing the parent causes the **init** process to inherit the **android** orphan process and to receive the child's termination message, which allows the child to completely terminate.

An example follows of using the PID number of the child's parent to kill the hung process:

```
# kill 5173 ↵
```

For details on methods to kill processes, see the section titled "Solving process problems."

CPU hog

There are two types of CPU hogs:

- a legitimate user process that is competing for limited resources during peak work hours
- a system process that is doing something wrong, which usually points to an application containing a bug. This type of CPU hog is called a runaway process or a rogue process.

The former CPU hog can be handled by rescheduling it. But, the latter must be terminated before it causes serious system harm — such as overfilling the / file system or crashing the system.

Detecting CPU hogs requires that you monitor your system's activities with the **nps** command, which may produce lengthy output. Knowing the types of processes running on your system will help you quickly identify likely CPU hog culprits.

Detecting a legitimate CPU hog

You may suspect a CPU hog when system performance starts to decline. Also, intensive I/O activities may cause a disk or tape drive to click rapidly. To investigate the problem, you would check process status with the **nps** command. Because your system is likely executing many processes, a stream of output probably will scroll up and off your screen. Be sure to redirect your output appropriately.

The following example restricts output to only five fields. See the **nps**(1) manual page for more information on specifying output. Also, the output is restricted to only the CPU hog process. Typical output, however, would report status on all its processes. Ellipses (...) indicate intervening output.

```
% nps –ostate, pid, stime, time, cmd ↵
S      PID  STIME    TIME CMD
R     28376 09:42:20 0:05 zar –cvf /dev/null /
...
% nps –ostate, pid, stime, time, cmd ↵
S      PID  STIME    TIME CMD
R     28376 09:42:20 0:05 zar –cvf /dev/null /
...
% nps –ostate, pid, stime, time, cmd ↵
S      PID  STIME    TIME CMD
R     28376 09:42:20 0:07 zar –cvf /dev/null /
...
% nps –ostate, pid, stime, time, cmd ↵
S      PID  STIME    TIME CMD
R     28376 09:42:20 0:08 zar –cvf /dev/null /
...
% nps –ostate, pid, stime, time, cmd ↵
S      PID  STIME    TIME CMD
R     28376 09:42:20 0:10 zar –cvf /dev/null /
...
```

In this example, the status of PID 28376 is consistently “R,” which means running. Since the starting time of the process, the cumulative execution of the process is 10 seconds. Comparing the processing status of PID 28376 with other PIDs, you may infer that the **zar** command is using excessive CPU.

Possible solutions are to reschedule the process during off-peak hours, to suspend the process and resume it later, or to adjust the process's priority. For details on handling a legitimate CPU hog, refer to the section titled "Solving process problems."

Detecting a runaway process

Monitor your system load with the **nsar** and **ps** commands, checking for CPU time used and percent idle time. A process that uses excessive amounts of CPU may be a runaway process.

> **IMPORTANT** You may also use the optional, separately purchased tool called UX/RPM (the DG/UX Real-time Performance Monitor) to collect and display an ordered list of processes that are using CPU time.

Use the **nsar** command in the following format to report the CPU percentage used by user processing time, system processing time, and idle time.

> **IMPORTANT** We encourage you to use the **nsar** rather than the **sar** command to check system activity. These commands produce the same type of output, but some fields are interpreted differently. Do not use the **nsar** and **sar** commands interchangeably. For detailed information on these commands, refer to the respective manual pages.

nsar -cu *t n*

where: the **c** option reports the number of system calls executed per *t*, the **u** option reports the CPU percentage used. The *t* represents the number of seconds in a snapshot and *n* represents the number of snapshots to take.

An excerpt of typical **nsar** output follows:

Ellipses (...) indicate intervening output.

```
# nsar –cu 5 100 ↵
dgux homer 5.4R3.10 generic AViiON      05/16/95
14:16:51 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
            %usr    %sys   %idle

14:16:56     141      21      14    0.00    0.00    2588     844
               2       4      95

14:17:01      90      11      11    0.00    0.00    6383     658
               2       3      96
...
14:18:32    4466    4422       9    0.20    0.20    1612     573
               6      29      64

14:18:37    6782    6754       7    0.00    0.00     679     703
               9      44      47

14:18:42    6753    6728       7    0.00    0.00     639     576
               9      44      47
...
```

IMPORTANT This output was obtained from a 45 MHz dual CPU system whose single process/thread consumed one Job Processor (JP).

User time (represented as “usr”) is the relative amount of time spent executing instructions in user code. A high user report suggests that one or more user applications are CPU-intensive. System time (represented as “sys”) is the relative amount of time executing instructions in kernel code. Large system numbers indicate lots of disk I/O, system call processing, scheduling, or handling a saturated resource, such as memory.

Idle time is the time spent by the system waiting for something to do. An ideal idle time depends on the number of job processors (JPs) the system has. For a single JP system, an ideal idle time is around 20 percent. For a 16-JP system, idle time will be considerably higher. If system load is low, idle time will be high. If idle time and system load are both high, there is probably a memory problem or there could be a disk or network I/O problem.

If idle time is very low, near 0 percent, there may be a runaway process. Depending on the circumstances, the 0 percent idle time could indicate a fully used CPU, which is good, or it could indicate a saturated CPU, which is probably bad. Idle time at 0 percent for an extended time period usually results in poor system performance.

The output line in the preceding example that is set in bold indicates a sudden surge in CPU consumption. The output shows increases in user and system usage, and a reduction in idle time. Also, the number of system calls and system reads increased dramatically. All these indicators point to a runaway process.

After you determine that there is a potential runaway process, you need to identify it by its PID. Isolating the PID may require careful and methodical analysis of **nps** output. Knowing what applications are running on the system will help you isolate the problem.

The following C shell script takes multiple ten-second snapshots of process status data collected by the **nps** command. Each snapshot is then directed to the **diff** command, providing a comparison of process status at ten-second intervals.

```
#! /bin/csh
nps –ostate, uname, pid, stime, time, cmd >ps1
while (1)
mv ps1 ps2
nps –ostate, uname, pid, stime, time, cmd >ps1
diff ps1 ps2
sleep 10
end
```

Ellipses (...) indicate intervening output.

```
...
19,20c19,20
< R    homer 28333  09:28:43    0:11 bmoc
< R    homer 28341  09:28:55    0:00 nps –o
---
> R    homer 28333  09:28:43    0:01 bmoc
> R    homer 28336  09:28:44    0:00 nps –o
68c68
< W     root   492  Apr 27 ?  135:02 ypserv
---
> R     root   492  Apr 27 ?  135:02 ypserv
19,20c19,20
< R    homer 28333  09:28:43    0:22 bmoc
< R    homer 28346  09:29:06    0:00 nps –o
---
> R    homer 28333  09:28:43    0:11 bmoc
> R    homer 28341  09:28:55    0:00 nps –o
...
```

The **diff** command produces output in pairs: recent snapshot and previous snapshot, followed by an underscore. The first reading indicates that command **bmoc** used 11 CPU seconds in the recent snapshot and 1 second in the previous snapshot. The next reading shows that **bmoc**'s CPU use increased to 22 seconds. By contrast, the **ypserv** command has used no CPU time between snapshots. Although **ypserv** has executed for 20 days (April 27 – May 16), it is using relatively little CPU time. PID 28333, which controls the **bmoc** command, and owner **homer** identify the runaway process in this example.

You may kill this process using the following command:

```
# kill 28333 ↵
```

For details on methods to kill processes, see the section titled "Solving process problems."

Memory leaks

A memory leak is excessive allocation of memory that is not freed, slowing system response over time. A memory leak usually results from a program that contains an instruction to allocate memory, but lacks an instruction to free it.

If you suspect a memory leak, execute the following **nsar** command to monitor your system load. To obtain meaningful information from the **nsar** reading, you need to monitor your system's performance on a regular basis. You must compare previously collected statistics with the statistics received when you suspect a memory leak.

```
$ nsar -WOkanon-res,upanon-res,usanon-res,pfile-res,dfile-res 30 10 ↵
```

The following statistics describe the distribution of resident pages among various memory classifications, in units of pages. Anonymous memory does not correspond to a file object and, if paged, is paged to the swap area. Private mapped pages exist in the address space of a single process, while shared mapped pages may exist in the address space of more than one process. File pages, when paged, are paged from the file system. Data file pages may be read and written; program file pages are only read from the file system.

where:

-WO	Precedes a comma-separated list of requested statistics
kanon-res	Kernel anonymous pages
upanon-res	User private anonymous pages
usanon-res	User shared anonymous pages
pfile-res	Program file pages
dfile-res	Data file pages
30	30-second interval
10	10 snapshots

Typical output follows:

```
dgux headroom 5.4R3.10 generic AViiON    05/15/95

13:36:12  kanon upanon usanon  pfile  dfile
13:36:42   4218   2480     12   2902   4496
13:37:12   4219   2487     12   2902   4498
13:37:42   4219   2490     12   2902   4498
13:38:12   4219   2494     12   2902   4499
13:38:43   4219   2498     12   2902   4500
13:39:13   4219   2508     12   2902   4501
13:39:43   4219   2516     12   2902   4501
13:40:13   4219   2519     12   2902   4501
13:40:43   4219   2525     12   2902   4501
13:41:13   4219   2529     12   2902   4501

Average    4218   2504     12   2902   4499
```

Over the five minute data collection period, only the **upanon** statistic shows a significant increase — from 2480 pages to 2529 pages: an increase of 45 pages. Although one other statistic increased by five pages, none of the others changed at all.

Knowledge of memory internals suggests that there is a memory leak, whereby allocated memory is not being freed. Even without intimate knowledge of how memory operates, you may still observe a suspicious increase in the **upanon** statistic.

Next, try to identify the process that is causing this problem.

IMPORTANT You may use the the optional, separately purchased tool called UX/RPM (the DG/UX Real-time Performance Monitor) to collect and display an ordered list of processes that are consuming the most memory.

If you do not have UX/RPM, you may use the **nps** command. Isolating the PID may require careful and methodical analysis of **nps** output. Knowing what applications are running on the system will help you isolate the problem. Because your system is likely executing many processes, a stream of output probably will scroll up and off your screen. Be sure to redirect your output appropriately. Executing the following **nps** command at regular intervals may reveal suspicious trends.

```
# nps –eodrss,swap,pid,state,uname,args | sort –r ↵
  DRSS   SWAP   PID S      UID CMD
   389    573   733 W     root nsrd
   339    344  2689 W     root ./leaky
   303    565 26323 W    homer /usr/opt/X11/bin/mwm
   192    913   567 W     root /usr/lib/lpsched
   192    438  2691 W    homer /usr/opt/X11/bin/xterm
   172    438 26671 W    homer /usr/opt/X11/bin/xterm
   155    438 29062 W    homer /usr/opt/X11/bin/xterm
   151    438 26320 W    homer /usr/bin/X11/xterm –e
    40    208   526 W     root rpc.lockd –g 0
    40    118   420 W     root syslogd
```

where:

DRSS Data memory resident set size; **drss** is a memory location from which the **malloc** command allocates resources.

SWAP Memory location from which **malloc** allocates anonymous memory.

The **sort** command with the **–r** option orders the output in reverse, organizing DRSS statistics, from highest to lowest values. Processes appearing toward the top of the list consume the most DRSS memory. A process that moves up the list with each subsequent snapshot, while all other processes remain stable in memory consumption, is a likely suspect.

After you isolate one or more suspicious processes, you may wish to write a script to collect more statistics about the suspicious process. The following Bourne shell script verifies a process's consumption of memory resources from DRSS and SWAP.

```
$ while true
> do
> nps –opid,drss,swap,args –p 26733 | tee –a nps.out
> sleep 15
> done
```

Typical output follows:

```
  PID   DRSS   SWAP CMD
26733     40     49 leaky
  PID   DRSS   SWAP CMD
26733     45     49 leaky
  PID   DRSS   SWAP CMD
26733     57     65 leaky
  PID   DRSS   SWAP CMD
26733     61     65 leaky
...
  PID   DRSS   SWAP CMD
26733    143    147 leaky
  PID   DRSS   SWAP CMD
26733    151    163 leaky
  PID   DRSS   SWAP CMD
26733    159    163 leaky
  PID   DRSS   SWAP CMD
26733    167    180 leaky
```

Ellipses (...) indicate intervening output.

During the data collection period, PID 26733 has consumed significant memory from both DRSS and SWAP memory areas. DRSS consumption increased from 40 to 167 pages— an increase of 127 pages; SWAP consumption increased from 49 to 180 pages — an increase of 131 pages.

The example shows that PID 26733 produces a memory leak and requires termination.

The process controlled by program **leaky** probably contains a bug. For example, the following program example includes a **malloc** statement, which allocates 1024 bytes from the memory pool, but lacks a corresponding statement to free the allocation. For details on **malloc**, refer to the **malloc**(3C) and **malloc**(3X) manual pages.

```
#include <stdlib.h>
main()
{
while (1)
{
malloc (1024);  /* Forgotten malloc without free */
sleep (2);      /* Other application work.       */
}
}
```

You may kill this process using the following command:

```
# kill 26733 ↵
```

For details on methods to kill processes, see the section titled “Solving process problems.”

Solving process problems

Solutions for process problems from least to most drastic, are:

- Rescheduling a process during off-peak hours
- Suspending and resuming a process
- Adjusting process priority
- Killing a runaway process

Use the first three solutions for legitimate CPU hogs; use the final one for runaway processes and memory hogs.

Rescheduling a process during off-peak hours

If you determine that a process that uses excessive CPU resources is legitimate, you have several options. The least drastic solution is to ask the owner of the process to reschedule the process during a time when the system will not be so busy, such as during night or weekend hours. Such a process can be scheduled using either the **crontab** scheduler or the **at** or **batch** commands. Refer to the **cron**(1M), **at**(1), and **batch**(1) manual pages and *Using the DG/UX® System* for information on submitting automated jobs at a later time.

Suspending and resuming a process

> **IMPORTANT** If you use the C or Korn shell, we prefer that you use its built-in job control facility for controlling job execution. For the Bourne shell, you may use shell layering. For information on the C shell and the Bourne shell, refer to *Using the DG/UX® System*. For information on Korn shell job control, refer to the *Kornshell Command and Programming Language*.

If the owner is unavailable, you might suspend the process until you can talk with the owner. Also, you should notify the owner. Suspend a process with the **kill** command format:

kill -23 *PID*

where **23** represents the SIGSTOP signal.

You must have appropriate privilege to use the **kill** command.

> **IMPORTANT** The parent of a child process may not be prepared to handle a suspended child. Resumption of a child process may result in abnormal behavior. Use the suspend and resume functions carefully.

An example follows:

```
# kill -23 28376 ↵
```

This command stops PID 28376 with the SIGSTOP signal. The process is not dead, but inactive. The process can be resumed with this command:

```
# kill -25 28376 ↵
```

This command activates PID 28376 with the SIGCONT signal. If you are near the I/O devices accessed by the **tar** command that is controlled by this PID, you may hear the disk and tape drives clatter to indicate intensive read and write activities.

Adjusting process priority

Lowering the priority of a legitimate CPU hog is another way to control it. A process's priority is assigned a default "nice" value of 20 (on a scale from 0 to 29) when the process begins execution. The term "nice" refers to how considerate you are of other system users when assigning process priority. The process owner can alter the priority with the **nice** command when the process starts. Because of competition for limited system resources, users should exercise good manners when assigning process priority. Refer to the **nice**(1) manual page for details on setting a process's priority.

Checking process priority

Check the priority assigned to your processes with the following command, which limits output to only the fields that you are interested in.

```
# nps -o pid,cmd,prior ↵
PID  CMD       PRI
5173 csh      1544
5177 lp       1544
5211 nps      1530
```

Look in the "PRI" column for the process's calculated priority value. The priority value results from computing several scheduling factors, including the **nice** value. For details on computing a process's priority, refer to *Analyzing DG/UX® System Performance*. Comparing priority values, you can determine the relative priorities of processes. Higher numbers mean a higher priority. In the example, the first two processes have an equal priority; the last one has a lower one.

Contrary to what you may think, the higher the nice value, the lower is the computed process priority. The default nice value of 20 produces a higher computed priority than would the maximum nice value of 39. To increase process priority, remember that you want a lower nice value. To lower process priority, increase the nice value.

```
To increase process priority --> Select lower nice value
To decrease process priority --> Select higher nice value
```

Lowering process priority

A lowered priority may allow other processes a greater share of CPU. To reduce the priority of the runaway process, use the **renice** command format:

renice *nice-value PID*

where *nice–value* is a number between –20 and 19 to be added to the default *nice-value* of 20.

> **IMPORTANT** Both the owner of the process and a user with appropriate privilege can change the priority of a process. However, only the user with appropriate privilege can raise priority.

The *PID* identifies the particular process. Remember that a higher nice value translates to a lower process priority; a lower nice value produces a higher process priority. To lower the priority of a runaway process, you will choose a higher nice value.

An example of lowering a process priority follows:

```
# renice 19 5177 ↵
5177: old priority 0, new priority 19
```

The **renice** command will execute PID 5177 at the nice value 19 greater than the default nice of 20. A higher nice value correlates to a lower CPU priority.

Display the lowered process status by issuing the **nps** command. An example follows:

```
# nps –o pid,cmd,prior ↵
PID  CMD       PRI
173  csh      1544
5177 lp       1344
5211 nps      1530
```

In this example, the process priority of PID 5177 was reduced from 1544 to 1344.

Killing a runaway process

IMPORTANT If you use the C or Korn shell, we prefer that you use the built-in job control facility for controlling job execution. For the Bourne shell, you may use shell layering. For information on the C shell and the Bourne shell, refer to *Using the DG/UX® System*. For information on Korn shell job control, refer to the *Kornshell Command and Programming Language*.

After you have identified the runaway process, terminate it with the **kill** command. Either the owner of the process or a user with appropriate privilege may kill processes. Kill a process with the **kill** command format:

kill *PID*

An example follows:

kill 28333 ↵

This command terminates PID 28333 and performs cleanup.

If this command fails to kill the process, specify the **9** kill signal, causing an explicit kill. The 9 kill signal disables the process's ability to perform cleanup.

kill -9 *PID*

where **9** represents the SIGKILL signal.

This command terminates the specified PID with the SIGKILL signal. The process is dead.

Killing a process is not always tidy. Such processes may load log files with error messages, leave various files in interim states, and leave expectant users waiting. Understanding the application that produced the runaway process will help you figure out how to clean up in the aftermath.

End of Chapter

7 Diagnosing a problem

A system problem can cause a variety of symptoms: a hang, a halt, an error message sent to your screen or to a log file, or some other unexpected behavior. Conversely, a symptom can result from a variety of problems. The need to troubleshoot may produce customer panic and a rush to the phone for help from the Customer Support Center. But please don't place the call just yet. The goal of troubleshooting is to find out what is wrong, not necessarily how to fix what is wrong.

By asking a few questions about your system in a logical step-by-step manner and recording your observations, you are actually troubleshooting. This process allows you to connect the symptoms of the problem to the underlying cause. You may be able to solve the problem by yourself or you may need the help of the Customer Support Center. If you need to call the Center, you can furnish them with the information they need to continue the troubleshooting process.

This chapter provides a worksheet containing troubleshooting questions that help you to organize the information that you have about your system. If you need to call the Support Center, also have available your system administration records.

Checking the hardware

The source of problems is software, hardware, or a combination of the two. Some error messages report recoverable conditions while others require more investigation. The first step of any troubleshooting procedure might be to go through a checklist of hardware connections and switch settings. The solution to a problem may be as simple as securing cable connections.

- Make sure that all cable connections are secure. Tighten screws on connections that attach the cable to the unit.
- Verify that power switches on the affected units are in the ON position. Make sure that all power cords are connected properly.
- Make sure that the power supply for each unit is operational. Signs of an active power supply are engine fans, lit indicators on control panels, and whirring disks inside drives.
- Verify that units containing removable media are properly seated with drive doors properly closed.

- For printers, make sure there is paper in the tray and that the online switch is turned on. Also, verify the commands you used. For detailed printer troubleshooting information, refer to *Installing and Managing Printers on the DG/UX® System*.

Determining the extent of the problem

Part of the troubleshooting process is to determine how extensive the problem is. It may be limited to a single terminal or workstation within a DG/UX configuration or it may pervade the entire system, causing all operations to stop. Consider the items in the following checklist to help determine the extent of the problem.

- If you are using a DG/UX system connected to a network, find out if all remote users are experiencing the same problem. If so, the source of the problem is with the network. For network troubleshooting information, consult the appropriate network documentation.

- If you have a problem while using a particular application, try to approximate the operation with another application. If you have no trouble with the test application, you may suspect your application is faulty.

- If you have a problem while using a particular mass-storage device that uses removable media, try a different medium, if possible. If the problem persists, try a different operation on the device. If the problem persists, try another device. From this exercise, you can determine whether you have a faulty medium, a faulty application, or a faulty device.

  Similar logic applies to printers. If you have a printer problem, try sending the output to another printer. You may be able to isolate the problem to one printer. Otherwise, for printer troubleshooting help, see *Installing and Managing Printers on the DG/UX® System*.

- If you use a PC, workstation, or terminal connected to a server, ask other system users if they are experiencing the same problem. If so, the problem resides with the server or the network and communications components.

  Table 7–1 lists the typical network configurations. Select the one that applies to your situation and continue troubleshooting.

Table 7–1 Network configurations

	Display Device	Connection	To server
1.	PC, workstation	Communications package such as TCP/IP or Novell NetWare	Local Area Network (LAN) controller
2.	PC, workstation	Communications package and **telnet** command	VMEbus terminal controller (VTC)
3.	PC, workstation, terminal	termserver	LAN controller
4.	PC, workstation, terminal	termserver	VTC
5.	PC, workstation, terminal	direct connect	Direct RS-232 connection

1. If you use a PC or workstation connected to the network through a communications package and connected to a server through a LAN controller, see *Managing TCP/IP on the DG/UX® System* for detailed troubleshooting procedures.
2. If you have a PC or workstation connected to the network through a communications package and connected to a server through a VTC:

   To troubleshoot possible workstation or server problems, see *Managing TCP/IP on the DG/UX® System* for detailed troubleshooting procedures.

   To troubleshoot possible VTC problems, check the **/var/adm/messages** log file.

   To troubleshoot possible problems associated with tty monitor lines on the VTC, check the **/var/saf/***ttymon_name***/log** and **/var/saf/_log** log files.

   Make sure that the **ttymon** is running, using the following command:

   ```
   # admportmonitor -o list ↵
   ```

   If a **ttymon** has stopped, then restart it. If it stops again, check **/var/saf/_log**.
3. If you have a PC, workstation, or terminal connected to the network through a termserver and connected to a server through a LAN controller:

Try to determine whether all devices connected to the termserver cannot communicate. If no devices are operable, then the problem probably resides with the termserver. Verify that the termserver is operational by inspecting its fault indicator lights (usually yellow and red). If a fault light is displayed, reboot the unit by turning power off and then on. If the termserver does not reboot, the unit is faulty and needs repair.

If you determine that the termserver is not faulty, you may suspect a network problem. See *Managing TCP/IP on the DG/UX® System* for detailed troubleshooting procedures.

4. If you have a PC, workstation, or terminal connected to the network through a termserver and connected to a server through a VTC:

   To troubleshoot possible workstation or server problems or VTC problems, perform all actions in step 2.

   To troubleshoot possible termserver problems, perform all actions in step 3.

5. If you have a PC, workstation, or terminal connected to a server through an RS-232 direct connection:

   To troubleshoot possible **ttymon** problems, check the **/var/saf/***ttymon_name***/log** log file.

   Make sure that **ttymons** are running, using the following command:

   # **admportmonitor –o list** ↵

   Verify that the port to which the display device is connected is being monitored, using the following command:

   # **admterminal –o list** ↵

   Try to verify the operation of each hardware component in the configuration.

Troubleshooting worksheet

Use this worksheet to record important configuration information that you and the Customer Support Center can use to isolate and fix your problem.

Figure 7–1 Troubleshooting worksheet

Troubleshooting Worksheet

1. Software revision number (**uname –a**) _____

2. Processor type (**uname –a**) _____

3. Packages and applications (**admpackage –o list –v**)

dgux	❑	onc	❑	X11	❑
dgux.aco	❑	sdk	❑	X11.aco	❑
gcc	❑	sdk.X11	❑	X11.sde	❑
nfs	❑	tcpip	❑	manual pages	❑

Other:_____
_____

4. Describe problem.

Error message ❑ _____
Halt code ❑ _____
Is it recurrent? ❑ Document date and time._____
_____
Is it sporadic? ❑ Document date and time_____
_____
Hang❑_____
Other _____
_____
_____

5. Describe what was happening prior to the problem. _____
_____
_____
_____

6. Have you changed the configuration recently?
Added new software ❑ _____
Upgraded system ❑ _____
Applied patch ❑ _____
Installed new hardware ❑ _____
Rebuilt kernel ❑ _____
Installed new rc scripts ❑ _____
Other ❑ _____

7. Check and record **/var/adm/messages** or appropriate log file
(See Chapter 5.) _____
_____
_____
_____
_____

(1 of 2)

Troubleshooting Worksheet

8. What is the state of your system?
   Operational? ❑ Init level _____
   Down to SCM? ❑
   At the prompt to take a dump? ❑
   Hung? ❑

9. What recovery actions have you taken?
   Suggested or recommended by error message? ❑
   Describe actions. _____
   _____
   _____
   Suggested by viewing log files? ❑
   Describe actions. _____
   _____
   _____
   Alter log file? ❑
   Describe actions. _____
   _____
   _____

   Reboot from SCM? ❑
   Press RESET? ❑
   Power off, then on? ❑
   Memory dump? ❑
   Run the file system checker (**fsck**)? ❑
   Fsck error message: _____
   Any core files? ❑
   Name and location: _____

10. Have you developed a work-around? What? _____
    _____
    _____

11. Do you have a printer problem? ❑
    For help, go to *Installing and Managing Printers on the DG/UX® System*.
    Do you a have a tty line problem? ❑
    For help, go to *Managing the DG/UX® System.*
    Do you have a modem problem? ❑
    For help, go to *Managing Modems and UUCP on the DG/UX® System*.

(2 of 2)

End of Chapter

8 Common problems and error messages

This chapter covers observable problems that do not report error messages, and it lists verbatim error messages in alphabetical order with recovery actions.

Common problems

This section covers observable problems that do not report error messages.

A user is not receiving mail, but other users are

The permissions on a user's mailbox may have been altered, preventing that user mailbox from receiving mail. Check the permissions for the user's mailbox. Correct the permissions if the user is not listed as the owner and the group is not listed as **mail**.

A mailbox having incorrect permissions can also cause mail sent to a group of users to be delivered repeatedly to the mailboxes that have correct permissions.

Check the system's **root** mailbox, typically at **/var/mail**, for this user's misdirected mail. The permissions assigned to the mailbox should be as follows:

```
# ls -l /var/mail ↵
-rw-rw----  1 harris   mail    1529 Apr 19 17:18 harris
```

For information on setting file permissions, see *Using the DG/UX® System* and the **chmod**(1) manual page.

If no users are receiving mail, see section "No one on system is receiving mail."

Cannot access an application that has been previously accessed

If you are trying to access an application on your host, verify that the application still exists. Someone may have deleted it.

If you are accessing an application on a remote host, verify that the NFS server, the network, and the remote file system are operational. Change to the directory in which the application resides and try to list the contents of the directory. If you cannot list the directory, try to connect with the remote host using the **ping** *hostname* command format. If you cannot do either, there is a host or network problem. Contact the remote host system administrator and the network administrator for help.

You may recognize this problem when trying to execute a command (such as **ls**) or execute an application that is located in a file system that has become corrupted. If you are trying to access an application that resides in a file system other than **/** or **/usr** file system, unmount that file system and check the file system with the **fsck** command. Command formats follow:

admfilesystem –o unmount *mount–point*
fsck [*options*] *filesystem*

Refer to Chapter 10 for details on running **fsck**.

If the application you are trying to run resides in either the **/** or **/usr** file systems, either or both of them may be corrupted. To restore these file systems, take the system down and boot stand-alone **sysadm**, which can be used to check and repair the **/** and **/usr** file system. Refer to Chapter 10 for details on restoring the **/** and **/usr** file systems.

Another possible explanation for this problem is that an unavailable remote file system included in your search path may prevent access to any other application located in a subsequent file system in the search path.

If you cannot access application **bbox**, for example, check your search path and identify all preceding file systems in the search path list as potential culprits.

```
# echo $PATH ↵
/usr/bin:/pdd/harmony/mmp/bin:/grinterleaf/gleaf5/bin:/usr/op
t/bbox3.1/bin
```

In this example, there may be problems with **/usr/bin**, **/pdd/harmony/mmp/bin**, or **grinterleaf/gleaf5/bin**. Verify that each file system is mounted. If a file system is mounted, try to access it. Finally, check each of these file systems with the **fsck** command.

As a temporary measure, rearrange your search path so that your application appears first in the search path list, bypassing the possible culprits. An example of reordering a search path follows:

Bourne shell:

```
$ PATH=/opt/bbox3.1/usr/bin:/pdd/harmony/mmp/bin:
/grinterleaf/gleaf5/bin:/usr/bin ↵
$ export PATH ↵
```

C shell:

```
$ setenv PATH = /opt/bbox3.1/usr/bin:/pdd/harmony/mmp/bin:
/grinterleaf/gleaf5/bin:/usr/bin ↵
```

Mirror sync failure

At the beginning and end of a synchronization session, the system logs a message using the **syslog** error logging facility. The message is from the **kern** facility and is level **warning**. By default, **kern.notice** messages go to the system console and to the file **/var/adm/messages**. If synchronization fails, the system logs a **kern.err** message, which by default goes to the system console and to **/var/adm/messages**.

An example of such a warning follows:

```
Oct 22 10:31:45 psycho dg/ux: Warning:  Image
'vdm(gth_image2,2CC7EE0E,0C052A9D,0)' on mirror
'vdm(gth_mirror,2CC7EE30,0C052A9D,0)' has failed
(status=77005171).
```

Subsequently, when the mirror attempts to update the time stamp of both images, it gets a failure when writing to the failed one. The following warning is then displayed:

```
Oct 22 10:31:45 psycho dg/ux: Error:  Cannot store the attributes for
   virtual disk 'vdm(gth_mirror,2CC7EE30,0C052A9D,0)' on disk
   'sd(insc(0,7),1,0)' (status = 77005171).
```

Check the description of the `77005171` status code in **/usr/release/dgux_5.4R4.10.status.codes** .

Retry the synchronization. Should the message persist, the hardware (cable, disk drive, or disk controller) is probably faulty. Report the problem to the Customer Support Center, supplying the status code that was delivered in the warning message.

If you determine that a hardware component is broken, you must replace the damaged hardware, and perform the following steps to re-create the mirror:

1. Unlink the damaged mirror image.
2. Build a new mirror image on a new disk.

3. Link the new image to the mirror.
4. Start a new synchronization from the master image to the new image.

For complete information on creating a mirror image, see *Managing Mass Storage Devices and DG/UX® File Systems.*

No one on the system is receiving mail

The file system that contains your system's mailbox may be full. You need to delete some files or compress files in that file system. See Chapter 3 for information on maintaining sufficient file system space.

Ask your system's users to clean up their mailboxes, freeing up space.

Additionally, you need to clean up the system mailbox's file system. For example, if the system mailbox is located in **/var/mail**, the **/** file system will be full. You may choose to expand the file system, increasing its size, using the following command format:

admfilesystem –o expand *method 'mount directory'*

See *Managing Mass Storage Devices and DG/UX® File Systems* for details on expanding a file system.

Perform the recovery actions provided in the section "No space left on device."

Slow response to remote file access

If access to remote files seems unusually slow, type the following command at the server console to be sure it is not being disrupted by a runaway daemon or a bad tty line:

```
# ps –ef ↵
```

If the server seems fine and other users are getting good response, make sure the client's block I/O daemons are running. Verify that **biod** is running, using the following command:

```
# ps –f | grep biod ↵
```

If **biod** is running, check your network connection.

Determine whether or not packets get lost or are being ignored, using the following command:

```
# netstat –i ↵
```

Use the following commands to check whether the client is doing lots of retransmitting or whether there are many bad calls:

```
# nfsstat -c ↵
# nfsstat -s ↵
```

A retransmission rate of five percent is considered high. Excessive retransmission usually indicates a bad network board, a bad network tap, a mismatch between board and tap, or a mismatch between your network board and the server's board.

System is running slowly

A slow system suggests that either your system or the network or both are overloaded. This condition usually subsides as network traffic and system processes diminish. To ease your system's workload, you may schedule some jobs during off-peak hours or kill processes that are not urgent. Refer to Chapter 6 for information on monitoring processes.

You may also need to tune your system's performance. Refer to Chapter 6 in this manual and *Analyzing DG/UX® System Performance* for help on tuning system performance.

Finally, you may need to purchase additional disk devices, memory or CPU processors to bear the workload. For product information, contact your Data General sales representative, or access Data General's electronic DASH (Direct Access Support Help) facility or Data General's WWW home page, which is accessible through URL address **http://WWW.dg.com/**.

Error messages

This section lists verbatim error messages in alphabetical order and suggests recovery actions.

Access denied

You do not have permission to access the file. Check the file's permissions with the **ls –l** command. An example follows.

```
# ls -l games ↵
-rw-------   1 root   doc        676 Jan 30  1995 games
```

Since **root** has ownership, ask **root** to allow group or other read or write access, as desired. If you are the owner, grant permission with the **chmod** command, as follows:

```
# chmod go +rw games ↵
# ls -l games ↵
-rw-rw-rw-   1 root   doc        676 Jan 30  1995 games
```

Cannot fork

A process is a command being executed or waiting to be executed. You have exceeded the maximum number of processes allowed. Check the number of processes on your system and determine how critical they are. Refer to Chapter 6 for details on monitoring processes.

```
# ps -ef | wc -l ↵
```

To enable critical operations to execute, you may kill idle processes (such as your electronic mail system, clock, publishing systems, and various windows). Use the following command to terminate processes, freeing up system resources.

```
# kill PID ↵
```

Alternatively, you may tune two kernel parameters, MAXUP and NPROC, increasing the maximum number of processes allowed. MAXUP specifies the maximum number of processes that a user can have at one time. The default is 50. This value should not exceed the value of NPROC (NPROC should be at least 10% more than MAXUP). This value is per user identification number, not per terminal. For example, if ten people logged in with the same user ID, the default limit would be reached very quickly.

NPROC specifies the maximum number of user processes the system can have at one time. The default value is 64. Larger values are needed for any graphical user interface. For various sized systems, use the following values: small (such as workstations), 96; medium (the default, such as AViiON 4xxx and 5xxx computers), 256; and large (such as AViiON 62xx, 8xxx, and 9xxx computers), 2048. The overall number of processes needed depends on the number of terminal lines available, the number of windows opened, the number of processes spawned by each user, and the number of system processes and network daemons. If the maximum number of processes is used up, the fork or vfork system call will result in a process table overflow and will fail. Refer to the **fork**(2) or **vfork**(2) manual pages for more information on these system calls.

See *Managing the DG/UX® System* for information on tuning and building kernels.

Rebuild and reboot the kernel.

Cannot open /dev/rmt/x: no such device or address

The system doesn't recognize the device. Make sure that the drive door is closed securely. Ensure that the medium is not write-protected. If you're using a QIC cartridge tape, make sure the arrow points away from the SAFE position.

Make sure you are using the correct device name.

Verify that the name of the specified device resolves to an actual device name. An example follows:

```
# ls -al /dev/rmt ↵
total 4
drwxr-xr-x 2 root     root         512 Apr 18 10:28 ./
drwxr-xr-x 11 root    root         5632 Apr 26 10:25 ../
lrwxrwxrwx 1 root root 26 Apr 18 10:28 0 ->  /dev/rmt/st(ncsc(0,7),2,0)
lrwxrwxrwx 1 root root 27 Apr 18 10:28 0n -> /dev/rmt/sd(ncsc(0,7),2,0)n
```

For example, device **0** resolves to a device named **st(ncsc(0,7),2,0)**.

Cannot open /dev/ttyxx: resource temporarily unavailable

Requiring no action on your part, this message indicates that **ttymon** could not open the given port, probably because the exiting process had not completed closing the port. The port monitor probably is successful on a subsequent open of the line.

Cannot open /dev/ttyxx: text file busy

Requiring no action on your part, this message indicates that **ttymon** could not open the given port, probably because the exiting process had not completed closing the port. The port monitor probably is successful on a subsequent open of the line.

Cannot open dev /dev/ptmx

The pseudo tty clone device **/dev/ptmx** is opened when assigning an available pseudo tty. If the open fails, the kernel parameter PTYCOUNT probably needs to be increased. PTYCOUNT is the number of pseudoterminal devices that will be created when the system boots. The default is 64.

Can't find swap

There are two likely causes for this error.

1. There is no **swap** area defined in the system.
   a. Boot the stand–alone **sysadm** and create a swap area using:
      SCM> **b sd(insc(),0) usr –f /stand/sysadm** ↵
   b. Then follow this path:
      `File System –> Swap Area –> Add`
2. The **swap** area is on a device not defined in the kernel.
   a. Boot the installer kernel from the SCM as in the following example:
      SCM> **b sd(insc(),0) root –f /dgux.installer –i 1** ↵
   b. From the stand-alone **sysadm** main menu, follow this path:
      `System –> Kernel –> Build`
   c. Specify the name of the physical disk that contains the **swap** virtual disk and rebuild the kernel and reboot the system. For details on kernel building, refer to *Managing the DG/UX® System*.

Although unlikely, you could receive this message if you booted a kernel that had no knowledge of virtual disks, with the swap area residing on a physical disk in virtual disk format. For example, you may have accidentally booted a DG/UX 5.4R2.xx kernel on a system that runs at a DG/UX 5.4R3.00 or later release. Verify the kernel that you booted by checking the SCM and the **dg_sysctl** boot settings. See Chapter 4 for information on setting the default boot device.

Device at <device–name> has encountered a hard error at block

This error message indicates a hardware problem. If the device is a tape drive, be sure that you have followed the recommended cleaning schedule. If you do not know the cleaning schedule, consult the hardware documentation. If the error persists, try another tape.

If the device is a disk, a bad sector has been encountered. Refer to *Managing Mass Storage Devices and DG/UX® File Systems* for information on remapping bad block areas.

Determine whether the device is manufactured by Data General or a third-party vendor. If it is a Data General drive, contact the Customer Support Center. If it is a third-party product, attempt the operation using a known operational Data General drive. If the problem persists, contact the Support Center. Otherwise, contact the third-party vendor.

Device busy

The device you are trying to access is already in use. A device is considered busy if a user is accessing a file on the device or if a user is "sitting" in a directory (resulting from the **cd** command) that is located on the device. A device is considered busy even if a user has an inactive or iconified xterm window whose current directory is on the device.

If you know the device's node name, you can find out who currently is accessing the device, with this command:

```
# /etc/fuser –u /dev/rpdsk/1 ↵
dev/rpdsk/1:      792t(davis)     814t(spade)
```

You may also get this message if you try to deregister a disk device that contains files systems that are still mounted. Unmount all file systems on the physical disk before deregistering it.

Also, you may get this message if you try to deconfigure a disk device that is still registered. Unregister the device before deconfiguring it.

This message may occur if you attempt to unmount a "parent" file system before unmounting any "child" file systems. First, try unmounting the child file system, then unmount the parent. If unmounting in the proper order does not work, you must reboot the system to clear the error.

This message also can mask other problems. A file system's type in the **/etc/fstab** file on the "busy" device may be incorrect or absent. This error may have resulted from omitting the **–t** option to the **admfilesystem –o add** command issued from the shell. Verify the file system types on the "busy" device using the **admfilesystem –o list** command. Make sure that the listing accurately identifies the file system type. You may also use the **admpdisk –o list** and **admvdisk –o list** commands to associate the file system types with their host virtual and physical disks.

To fix a misidentified file system type, remove the entry from the **fstab** file with the **admfileystem –o delete –x** *'mount–point–directory'* command. Then re-enter the file system, specifying the correct type, into the **fstab** file.

If, for example, you learned that the **/usr/opt/quotes** file system located on a CD-ROM device was incorrectly identified as being on a diskette device, you might do the following:

```
# admfilesystem –o delete –x '/usr/opt/quotes' ↵
# admfilesystem –o add –t cdrom –f 'quotes' /usr/opt/quotes ↵
```

If you try to access a CD-ROM device that is formatted in the High-Sierra format and get this message, check the device's registration status with the **admpdisk –o list** command. Since the device does not contain virtual disks, it needn't be registered. If the device is registered, deregister it with the **admpdisk –o deregister** command and retry the operation.

Finally, this message may also result from a system crash, which may have corrupted a file system on the "busy" device. Using the **admpdisk**, **admvdisk**, and **admfilesystem** commands with the **–o list** argument, you can derive the names of the corrupt file system(s) on the "busy" device. After you identify the corrupt file system, you must unmount it before recovering it with the file system checker (**fsck**). An example follows:

```
# admfilesystem –o unmount /usr/opt/quotes ↵
# fsck /usr/opt/quotes ↵
```

Refer to Chapter 10 for details on the commands to list and unmount file systems and to fix problems with **fsck**.

File not found

The directory that contains the command that you executed may not be in your search path. Most commands are located in **/bin** and **/usr/bin**. Review the contents of the DG/UX file system in *Managing the DG/UX® System*.

Find out where the command is located on your system using the **whereis** command, which checks all standard locations, such as **/bin** and **/usr/bin**:

```
# whereis cat ↵
cat: /bin/cat /usr/bin/cat /usr/ucb/cat /usr/catman/u_man/man1/cat.1.Z
```

The output identifies the location for the source, binary, and manual page for the **cat** command.

If **whereis** doesn't reveal the command's location, ask someone else who can access the **cat** command on her system to use the **which** command. The **which** command scans the user's search path (which contains both standard and nonstandard locations) for the location of the **cat** command. If the **which** command is successful, update your search path with this information.

Check your search path as follows:

```
# echo $PATH ↵
/usr/bin:/pdd/monarch/mmp/bin:/usr/global/bin
```

How you add the location to your path depends on the shell you are using.

Bourne shell:	C shell:
$ **PATH=$PATH:/usr/bin** ↵ $ **export PATH** ↵	% **setenv PATH=$PATH:/usr/bin** ↵

See *Using the DG/UX® System* for details on setting shell variables.

In you receive this message while booting the operating system, verify the bootstrap on the boot disk by listing the contents of the physical disk with the **admpdisk –o list –a** command. Look for the appearance of .**Bootstrap**. If there is no bootstrap, you must reinstall it. Boot stand-alone **sysadm** from the distribution release tape, and follow this **sysadm** path:

```
Install Software -> Prepare physical disks
```

Another possible explanation is that you are booting a file that either doesn't exist or is on a malfunctioning physical disk. If you receive this message when trying to boot **/dgux**, try booting other files such as **/dgux.installer** or **usr:/stand/sysadm**. If an alternate file boots, the file you originally tried booting probably doesn't exist. Someone may have deleted the file without your knowledge. You may try to recover the file from a backup tape. If you used the **dump2** or **cpio** commands to perform a backup, refer to Chapter 11 for help on restoring the file. Otherwise, refer to *Managing the DG/UX® System* for extensive information on backup services.

If you cannot boot another file from the disk, try booting a file from another known operational disk. If it boots successfully, the disk you first tried is probably faulty. Contact the Customer Support Center to report the problem.

File system full

Check the file system's access permissions. If it is read-only, you are not permitted to write to it, so may get this message.

You may get this message if you try to access a file system on a disk in a removable-media drive (diskette drive, CD-ROM drive or magneto-optical drive). Someone may have removed the disk without first unmounting the file system and deregistering the disk. To recover, you must re-insert the disk in the drive and run **fsck**. For information on running **fsck**, see Chapter 10.

Verify file system free space with the **df –k** command; refer to Chapter 3 for details on using the **df** command. If **df** shows that there is space, unmount the file system and check it with the **fsck** program. For information on running **fsck**, see Chapter 10.

If **df** shows limited free space, then the file system that you are working in is full, and you need to delete some files in root to free up disk space now. Refer to Chapter 3 for information to free up disk space.

Alternatively, if free space is available elsewhere, you may choose to expand the file system. For information about expanding a file system, see *Managing Mass Storage Devices and DG/UX® File Systems*.

File system on device <device–name> sealed

You lose access to a file system on a disk in a removable-media drive (diskette drive, CD-ROM drive or magneto-optical drive) if you take out the disk without first unmounting the file system and deregistering the disk.

To recover, you must re-insert the disk in the drive and run **fsck**.

Fork failed –– too many processes

A process is a command being executed or waiting to be executed. You have exceeded the maximum number of processes allowed. Check the number of processes on your system and determine how critical they are. Refer to Chapter 6 for details on monitoring processes.

```
# ps -ef | wc -l ↵
```

To enable critical operations to execute, you may kill idle processes (such as your electronic mail system, clock, publishing systems, and various windows). Use the following command to terminate processes, freeing up system resources.

```
# kill PID ↵
```

Alternatively, you may tune two kernel parameters, MAXUP and NPROC, increasing the maximum number of processes allowed. MAXUP specifies the maximum number of processes that a user can have at one time. The default is 50. This value should not exceed the value of NPROC (NPROC should be at least 10% more than MAXUP). This value is per user identification number, not per terminal. For example, if ten people logged in with the same user ID, the default limit would be reached very quickly.

NPROC specifies the maximum number of user processes the system can have at one time. The default value is 64. Larger values are needed for any graphical user interface. For various sized systems, use the following values: small (such as workstations), 96; medium (the default, such as AViiON 4xxx and 5xxx computers), 256; and large (such as AViiON 62xx, 8xxx, and 9xxx computers), 2048. The overall number of processes needed depends on the number of terminal lines available, the number of windows opened, the number of processes spawned by each user, and the number of system processes and network daemons. If the maximum number of processes is used up, the fork or vfork system call will result in a process table overflow and will fail. Refer to the **fork**(2) or **vfork**(2) manual pages for more information on these system calls.

See *Managing the DG/UX® System* for information on tuning and building kernels.

Hard error

Your hardware is faulty. Determine whether the device is manufactured by Data General or a third party. If it is a Data General drive, contact the Customer Support Center. If it is a third-party product, attempt the operation using a Data General drive that you know works. If the problem persists, contact the Center. Otherwise, contact the third-party vendor.

Hard error at SCSI device <device–name>

See “Hard Error.”

Kernel build failed

The most common cause of a build failure is insufficient space in either **/** or **/var/Build**. Check the amount of free space remaining in these locations with the **admfsinfo**, **df**, and **du** commands, and reclaim free space. For information on determining and reclaiming free space, refer to Chapter 3.

A critical kernel build script or library could have been deleted accidentally. Verify the existence of these files: **/usr/etc/master.d**, **/etc/sysdef**, and **/usr/sbin/config**. If any is missing, restore from a backup tape to the appropriate locations on your system. If you used the **dump2** or **cpio** commands to perform a backup, refer to Chapter 11 for help on restoring the file. Otherwise, refer to *Managing the DG/UX® System* for extensive information on backup services.

In addition, check **/var/Build/system.** *hostname***.build** for possible kernel build error messages, where *hostname* is the name of your computer. Look for references to missing files or insufficient space.

Another explanation is that you may have installed an MU (maintenance update) that is incompatible with the current operating system. Verify the DG/UX release and MU release numbers. Check your MU notice for installation requirements. At the shell, use these commands to determine when the last kernel successfully was built and to find out its DG/UX release number.

```
# ls –ail /dgux ↵
# uname –a ↵
```

Rebuild and boot a new kernel only after installing the correct MU release.

Finally, check for the absence of a comment symbol (#) used to flag notes that are to be ignored. Be sure you comment out all text you want ignored.

If you hand-edited the configuration file, verify the accuracy of all your entries, including DG/UX device names, drivers, and kernel parameters. You could have made a typographical error or entered a valid specification for a package that has not been loaded. Check **/usr/etc/master.d/*** for a list of valid device names. After you fix any mistakes, rebuild and reboot the kernel.

Mount: No such file or directory

The full text of the error message follows:

```
mount: /dev/dsk/usr_opt_fred on /usr/opt/fred: No such file or directory
mount: giving up on:   /usr/opt/fred
```

Possible causes for this message are that the virtual disk **usr_opt_fred** does not exist or the mount point for the virtual disk does not exist. List the virtual disks and mount points with these commands:

Try to list the file system.

```
# ls –d /usr/opt/fred ↵
```

Also, try these commands to check the names of mounted local file systems:

```
# admvdisk –o list ↵
# admfilesystem –o list –lm ↵
```

If the virtual disk was deleted deliberately, then you must also remove its entry from the **/etc/fstab** file, using the following command:

```
# admfilesystem –o delete –x /usr/opt/fred ↵
```

For more information on unmounting file systems, refer to *Managing Mass Storage Devices and DG/UX® File Systems*.

If the virtual disk exists, but its file system has not been added to the **/etc/fstab** file, do so with the following command format:

```
# admfilesystem –o add –t dg/ux –f 'usr_opt_fred' options  /usr/opt/fred
↵
```

Then mount the file system with the following command:

```
# admfilesystem –o mount '/usr/opt/fred' ↵
```

For details on creating a mount point and adding the file system to the **/etc/fstab** file, refer to *Managing Mass Storage Devices and DG/UX® File Systems*.

Mount: server not responding

The full text of the error message follows:

```
mount: custdoc:/usr/opt/cddoc server not responding: RPC: Port mapper
failure – RPC: Unable to send
```

Either the remote file server is down or its port mapper is not responding. Use the following command to list registered programs:

```
# rpcinfo –p server ↵
```

If no list is displayed, the port mapper on the remote server should be killed and restarted. Restarting the port mapper requires that other RPC daemons on the server be killed and restarted. Notify the system administrator of the remote server of this problem and the need for assistance. For more information on the port mapper, see the **portmap**(8C) manual page.

Mount: host not in hosts database

The full text of the error message follows:

```
mount: csc101 not in hosts database
mount: giving up on: /usr/fep
```

An OS client's file system that you are attempting to mount is not listed in **/etc/hosts** database, which contains host and Internet address pairs. Add the OS client to the OS server's hosts database using a command such as

```
# admhost –o add –a 128.224.24.9 csc101 ↵
```

or using the **sysadm** path:

```
Networking –> TCP/IP –> Databases –> Hosts –> Add
```

Network is unreachable

Your local host-to-network connection may have been misrouted. A route is the path that network traffic takes from its source to its destination. A router forwards packets of a particular protocol type from one network to another. To check the routes, use the following command:

```
# netstat -r | more ↵
```

Consult with your network administrator, if possible. Set the routes with the **route** command by changing the routing parameters in **/etc/tcpip.params**, or by using the **gated** command. Refer to *Managing TCP/IP on the DG/UX® System* for details on setting routing parameters.

If you are using dynamic routing, make sure that the **gated** route management daemon is running. Verify that the following definition is in the **/etc/tcpip.params** file.

```
DAEMON=gated
```

To enable **routed**, through **sysadm**, make this selection:

```
Networking -> TCP/IP -> Routes -> Dynamic Routing -> Start
```

If this entry is omitted, your attempts to connect to other computers on the local network will fail.

NFS server <server–name> not responding

If this message is displayed intermittently with the message, `NFS server <server–name> OK`, the server system probably is running slowly because of heavy network traffic. To reduce network load, you may run some jobs during off-peak hours. Refer to Chapter 6 for information on running processes during off-peak hours.

If the message is persistent, verify that the NFS server is responding with the **ping** command.

ping *hostname*

Verify the domain name. An example follows:

```
# domainname ↵
ux_lan
```

Check your network cable connections.

For information on networking, see *Managing ONC/NFS™/NFS® and Its Facilities on the DG/UX® System*, and consult your network administrator, if possible.

NO_SAC

Determine whether or not a SAC process is running with the **ps** command:

```
# ps –f | grep sac ↵
```

Refer to Chapter 6 for information on monitoring processes.

If SAC is running, check its status as follows:

```
# admportmonitor –o list ↵
```

If **ps** reports that SAC is running, but **admportmonitor** reports NO_SAC, then SAC is in an inconsistent state. The most likely explanation is that log file **/var/saf/_log** was not cleaned using the procedures described in Chapter 5.

If SAC is running, kill it with this command format:

kill –15 *PID–number*

After the **init** program detects that SAC died as the result of a kill operation, it will then automatically restart SAC.

No space left on device

There are numerous causes of this error, the most likely of which is that the file system is corrupt. Check the file system with the file system checker, **fsck**, as explained in Chapter 10. Other causes are listed below. Procedures for creating a new file system are given at the end of the section, since this solution potentially involves the loss of data.

- The file system you are attempting to access may be full. For information on how to verify free space and to reclaim space, see Chapter 3.
- You may have exceeded the size of a control point directory (CPD). You set a directory's size limit when you create the directory with the **mkdir** command. You can identify CPDs by using the **ls –1** command; a percent sign (%) in the first column of output signifies a CPD. Use the **cpd** command to check and increase the size of a CPD. Refer to the **cpd**(1) manual page for information on increasing the size of a control point directory.
- You may be trying to access a file system that is not mounted. Check the mounted file systems with the **mount** command. If necessary, mount the file system with the following command format:

admfilesystem –o mount *mount–directory*

For more information on mounting file systems, see *Managing Mass Storage Devices and DG/UX® File Systems.*

- The file system may not be mounting because its parent file system is full; for example, **/usr/opt/X11** cannot be mounted because the **/usr** file system is full. To recover, you must free up space in or expand the size of the parent file system. Refer to Chapter 3 for information on freeing up disk space.

  IMPORTANT To expand the root (**/**) or **/usr** file systems, remember that you cannot boot from a file system built on a virtual disk that spans multiple physical disks. Expansion partitions must reside on the same physical disk.

  Refer to *Managing Mass Storage Devices and DG/UX® File Systems* for details on expanding virtual disks.

- The file system may have insufficient inodes to allow you access to the file system. Inodes represent the total number of files that can exist on the system. Check a file system's inodes with this command format:

  # **df –i** *file_system_mount_point*

  You may make adjustments by tuning the file system or creating a new file system of the same size, with an increased number of inodes.

  To expand the file system, use the **sysadm** utility's `File System –> Local Filesys –> Expand`.

  Detailed information on how to create a new file system with increased number of inodes follows.

To create a new file system with increased number of inodes

If you have sufficient free disk space on your system, and expanding the current file system is not an option, then you can leave the old data in place while you create a new virtual disk and file system with an increased inode count. Steps 1 – 6 below show you how to do this. If you do not have sufficient free space, then skip this procedure.

1. If sufficient disk space is available, leave the existing file system in place, and create a virtual disk of the same size as the existing file system.

   ```
   admvdisk –o create –P
   ″device_spec:starting_block:number_of_blocks″
   new_vdisk_name
   ```

2. Use the **mkfs** command to create a file system on this new virtual disk with the required number of inodes. See the **mkfs** man page's discussion of the **–f** option, as well as other options you may want to use.

   ```
   mkfs –f number_inodes_required /dev/dsk/new_vdisk_name
   ```

3. Mount the new filing system in a convenient location.

   ```
   mkdir /mnt ↵
   mount /dev/dsk/new_vdisk_name /mnt ↵
   ```

4. Use the **cp** command with the **–r** option to move the data from the old filing system to the new one.

   ```
   cp –r /filesystem_mount_directory /mnt
   ```

5. Unmount the old filing system, and then mount the new one in its place, making sure that the **/etc/fstab** file is updated with the changed virtual disk name.

   ```
   umount /old_filesystem_mount_point ↵
   admfilesystem –o modify –f /dev/dsk/new_vdisk_name
   filesystem_mount_directory ↵
   mount filesystem_mount_directory ↵
   ```

6. After ensuring that all of your data has been moved to the new location, remove the original virtual disk.

   ```
   admvdisk –o remove original_vdisk
   ```

If you do not have sufficient disk space to leave the original data in place or to expand the file system, then you must make a backup copy of your data, recreate the file system with an increased inode count, and restore your data from backup.

1. Using any backup command other than dd, make a backup of your existing data. See *Managing the DG/UX System* for information on backing up a file system.

2. Unmount the file system.

   ```
   umount /filesystem_mount_directory
   ```

3. Recreate a new file system over the same virtual disk with an increased number of inodes.

   ```
   mkfs –f number_of_inodes /dev/dsk/vdisk_name
   ```

4. Mount the recreated file system.

   ```
   mount /filesystem_mount_directory
   ```

5. Restore your data.

For more information on creating a virtual disk and file system, see *Managing Mass Storage Devices and DG/UX® File Systems*. For more information on increasing inodes, see the **mkfs**(1M) manual page.

Not mounted

If you are attempting to unmount a file system, it may have been mounted before its parent, causing it to be hidden. When file systems are mounted in reverse order (child before parent), attempts to unmount a child file system before the parent will fail with the message **not mounted** instead of **device busy**, and erroneously remove the record for the child file system from **/etc/mnttab**. To avoid these problems:

1. Order file system records correctly in **/etc/fstab**. Parent file systems must be mounted before child file systems (for example, mount **/var** before **/var/spool)**.
2. When mounting a file system manually, ensure that any parent file systems have already been mounted by issuing **mount** with no arguments.
3. Unmount file systems in the opposite order from which they were mounted. If file systems were mounted incorrectly, it is still necessary to unmount them in the order opposite of mounting order.

Out of contiguous disk space

The file system in which you are working is full, and you need to delete some files to free up disk space now. To decide which files are candidates for deletion, see Chapter 3.

Out of STREAMS resources

The cause of this problem is either an excessive use of **FIFO** (first in first out) type files or an excessive network traffic.

Each **FIFO**, which is implemented through STREAMS, writes "invisible" control information that occupies streams memory. You may need to increase the STREAMS kernel parameter, PERCENTSTR, to use more memory. PERCENTSTR specifies the maximum percentage that can be used for STREAMS buffers from the lesser of the following: physical memory or available kernel logical memory (about 750 MB).

To improve network service, tune the STREAMS kernel parameter, NPIPE, and configuration parameter, MAXUP.

NPIPE specifies the maximum number of STREAMS pipe devices available. The default is 64. Note that this does not affects IPCs created by using **pipe**(). Large NPIPE values may adversely affect **ttyname** lookup times and increase CPU usage.

MAXUP specifies the maximum number of processes that a user (other than root) can have at one time. The default is 50. This value should not exceed the value of NPROC (NPROC should be at least 10% more than MAXUP). This value is per user identification number, not per terminal. For example, if ten people logged in with the same user ID, the default limit would be reached very quickly.

You alter the values of these parameters by building a custom kernel, editing directly the system file to tune these values, as desired. Boot the new kernel to initialize the revised values. Refer to *Managing the DG/UX® System* for information on kernel building.

Out of swap space

The number of users and applications is overloading the system. You may terminate processes that are contending for swap, freeing up space for the desired application. Refer to Chapter 6 for information on terminating processes.

Alternatively, create a second virtual disk for additional swap to distribute the swapping load. For detailed instructions on creating additional swap, see the chapter on swap areas in *Managing the DG/UX® System*. After determining the size of the additional swap space, create a virtual disk named **swap2**, for example, with no file system, using the following command format:

admvdisk –o create –n *number–of–blocks* **swap2**

Add **swap2** to the **/etc/fstab** file using the following command:

admswap –o add swap2 ↵

Notify the system of the additional swap space listed in **fstab** for subsequent reboots using the following command:

admdefault –o set –f /usr/dgux.params swapon_ARG = "–a" ↵

Finally, notify the system of the additional swap for immediate use with the following command:

swapon ↵

Permission denied

You do not have permission to access the file. Check the file's permissions with the **ls –l** command. An example follows.

```
# ls –l games ↵
–rw–––––––   1 root   doc        676 Jan 30  1995 games
```

Since **root** has ownership, ask **root** to allow group or other read or write access, as desired. If you are the owner, grant permission with the **chmod** command, as follows:

```
# chmod go +rw games ↵
# ls –l games ↵
–rw–rw–rw–   1 root   doc        676 Jan 30  1995 games
```

Printer hard fault –– waiting for auto–retry

This message, which results when checking the printer's status with the **lpstat –t** command, indicates a recoverable condition. Try enabling the printer with this command:

enable *printer–name* ↵

If the problem persists, try turning printer power off, then on.

Process table overflow

A process is a command being executed or waiting to be executed. You have exceeded the maximum number of processes allowed. Check the number of processes on your system and determine how critical they are. Refer to Chapter 6 for details on monitoring processes.

ps –ef | wc –l ↵

To enable critical operations to execute, you may kill idle processes (such as your electronic mail system, clock, publishing systems, and various windows). Use the following command to terminate processes, freeing up system resources.

kill *PID* ↵

Alternatively, you may tune two kernel parameters, MAXUP and NPROC, increasing the maximum number of processes allowed. MAXUP specifies the maximum number of processes that a user can have at one time. The default is 50. This value should not exceed the value of NPROC (NPROC should be at least 10% more than MAXUP). This value is per user identification number, not per terminal. For example, if ten people logged in with the same user ID, the default limit would be reached very quickly.

NPROC specifies the maximum number of user processes the system can have at one time. The default value is 64. Larger values are needed for any graphical user interface. For various sized systems, use the following values: small (such as workstations), 96; medium (the default, such as AViiON 4xxx and 5xxx computers), 256; and large (such as AViiON 62xx, 8xxx, and 9xxx computers), 2048. The overall number of processes needed depends on the number of terminal lines available, the number of windows opened, the number of processes spawned by each user, and the number of system processes and network daemons. If the maximum number of processes is used up, the fork or vfork system call will result in a process table overflow and will fail. Refer to the **fork**(2) or **vfork**(2) manual pages for more information on these system calls.

See *Managing the DG/UX® System* for information on tuning and building kernels.

SAD autopush configuration failed

The **/etc/inittab** file has been corrupted. This file contains instructions that define the processes to be created or terminated for each run level while the system is booting. A damaged **inittab** file prevents your system from booting.

To recover, you must re-create your **inittab** file from a prototype file. Refer to Chapter 10 for complete instructions.

SCSI timeout

Loose cable connections and cable terminators may have caused this error. Tighten all cable connections.

Another possible explanation is that a command was issued to a SCSI device, but that device did not complete the command within the allotted time. A timeout usually indicates a hardware problem. The system instructs the controller to request a response from the device through a SCSI reset. If the device still does not respond, the system may shut down the controller and device to prevent data loss or corruption.

If possible, retry the operation using a device attached to a controller that you know is operational. For example, if you are having the problem with device **sd(ncsc(0),1,2)**, which is attached to controller 0, try the operation on a device attached to another controller such as **sd(ncsc(1),3,2)**, which is attached to controller 1. If the problem persists on another controller, contact the Customer Support Center. If the operation succeeds on a good controller, you will know that the first tried controller is faulty. Contact the Support Center.

Set termio: set ttymode() failed: Interrupted system call

This message indicates that ttymon encountered an error when it set the initial terminal characteristics, and will retry later. Typically, no intervention is required. However, if the error persists, make sure your cables do not exceed the required length. Consult your hardware documentation for information on legal cable lengths. This message could indicate a bad port in a cluster box. If the message persists, consult with the Customer Support Center.

Soft error

See “Soft memory error”

Soft memory error

A soft memory error indicates that a single bit in a memory word is not correct, but that the system has been able to compensate by reconstructing the correct memory data. There are many environmental factors that contribute to an occasional soft, single bit error. Most soft memory error reports should cause no concern. However, if soft memory errors exceed the limits shown in the following table, contact the Customer Support Center. You may check the frequency of these errors by checking the error log **/var/adm/messages**.

Board Size (in Mbytes)	Maximum number of soft errors tolerated per Month
192	40
128	27
64	13
48	40
32	27
16	13

Tape drive at <device–name> encountered an unacceptably high number of correctable (soft errors)

Soft tape errors are reported on the system console for SCSI-2 tape drives. The system does not report these errors as they occur. Instead, the system compares the number of bytes transferred and the number of errors that occurred, and reports according to the ratio of errors to bytes. If there are a large number of errors ("a high number"), the system reports an acceptance level of marginal. Suggested maintenance includes cleaning the drive heads.

If there are a very large number of errors, the system reports an acceptance level of bad ("unacceptably high number"). If you receive this message, you are urged to clean the drive heads. If the message persists, contact the Customer Support Center for help.

This instance of mail is read only

You receive this message when you attempt to use **mailx**. You may read mail, but you cannot send mail. Each time you invoke **mailx**, a file named **.Maillock** is created in your home directory to indicate that **mailx** is executing. After you exit **mailx**, **.Maillock** is deleted. If **mailx** terminates abnormally or your system crashes, **.Maillock** remains in your home directory. Upon system restart, with **.Maillock** still in your home directory, the system believes you already are using **mailx** when you attempt to execute it again.

To recover, check for the **.Maillock** file in your home directory with the **ls –a** command. Delete **.Maillock** and restart **mailx**.

Unable to access imagex. Mirror breaking.

A mirror image is no longer operational, so it is broken from the mirror. If the number of remaining functional images is less than the minimum images required, the entire mirror may be unavailable.

To verify the status of the mirror, list the mirror's images using the following command format:

admvdisk –o list –a *mirror–name*

Refer to *Managing Mass Storage Devices and DG/UX® File Systems* for information on listing mirror images.

Create another image on a known good disk and synchronize the data. Your mirror is now operational.

The device containing the broken mirror image may be faulty. Contact the Customer Support Center for help.

Unable to configure device

If you custom build the kernel, make sure that the physical device being configured is physically attached to your configuration.

If you autoconfigured your kernel, the **probedev** utility sends a signal to all attached devices. Devices that return a reply are configured automatically. Those not returning a reply are not configured. When you boot, all configured devices will be operational.

Determine which device is not configured by using the **sysdef** command to check the configured devices. Make sure that the device is powered up and on line, and check the device's cable and terminator connections. Verify that the correct cable, terminator, and device interface type (single-ended or differential) is used. Refer to your hardware documentation for information on hardware requirements.

Verify the device's SCSI ID. You inadvertently may have assigned the same SCSI ID on the same controller to two devices, one of which does not get configured. Refer to *Managing Mass Storage Devices and DG/UX® File Systems* for information on device naming. Try to configure the device again. For example:

```
# admdevice –o configure 'sd(ncsc(0),2,5)' ↵
```

The device will be configured until you reboot the system. You must rebuild the kernel to incorporate this device before you reboot.

If the device still cannot be configured, you probably have a hardware problem. Call the Customer Support Center for help.

VDIT for disk <device–name> has degraded and is using only one VDIT copy

Although rare, a virtual disk information table (VDIT) can get damaged. Usually, this results from a software bug; for example, when you attempt to boot a pre-DG/UX 5.4R3.00 kernel on a DG/UX 5.4R3.00 system. Such an illegal operation can destroy the data at the beginning of a physical disk, which contains the VDIT.

1. To recover, copy the duplicate VDIT, which was written at the end of the physical disk during soft formatting, to the damaged location. Select the **sysadm** operation:

   ```
   Device –> Disk –> Physical –> Repair VDIT
   ```

Alternatively, you may use the shell command:

admpdisk –o repair_vdit *device–name*

2. In **sysadm**, you may list the physical disk for a display of both VDIT locations using this **sysadm** operation.

   `Device –> Disk –> Physical –> List`

   The physical disk must be deregistered and re-registered before the VDIT repair can take effect. The repair operation will attempt to deregister and re-register the physical device but will fail if it can't deregister the device. If deregistration fails, you may continue to use the disk in degraded mode until you are able to successfully deregister and register the physical disk. A degraded mode of operation allows read and write operations to continue, but prohibits operations that alter the disk's metadata such as enlarging, shrinking, moving, and mirroring. Although allowed, do not operate in a degraded mode for an extended period. (For information on deregistering a device and explanations of why deregistration fails, refer to *Managing Mass Storage Devices and DG/UX® File Systems*.)

3. Supply the name of the physical disk whose VDIT requires repair: For example:

   `Physical disk(s): sd(insc(0),1,0) ↵`

   This operation takes only a few seconds. The system copies the VDIT located toward the end of the physical disk to the beginning as well.

You are already reading mail

You receive this message when you attempt to use **mailx**. You may read mail, but you cannot send mail. Each time you invoke **mailx**, a file named **.Maillock** is created in your home directory to indicate that **mailx** is executing. After you exit **mailx**, **.Maillock** is deleted. If **mailx** terminates abnormally or your system crashes, **.Maillock** remains in your home directory. Upon system restart, with **.Maillock** still in your home directory, the system believes you already are using **mailx** when you attempt to execute it again.

To recover, check for the **.Maillock** file in your home directory with the **ls –a** command. Delete **.Maillock** and restart **mailx**.

End of Chapter

9 Recovering from power failures, hangs, and halts

This chapter describes the procedures for recovering from three kinds of system failures: power failures, hangs and halts.

Recovering from a power failure

Following a complete power failure, the system will reboot without operator intervention as soon as power returns to the system if a boot path is set in the PROM or through the DG/UX system.

> **IMPORTANT** Make sure that your bootable diskette drive is empty. If your system contains a diskette drive with a bootable diskette engaged, the system automatically attempts to reboot from the diskette drive if you experience a power failure. This produces a failure.

After such a reboot, check to see if the power failure damaged file systems as described in the section "Restoring file systems after a system failure" in this chapter. If your system has the Uninterruptible Power Supply (UPS) subsystem, refer to *Managing the DG/UX® System*.

If there is a partial power failure, such as a brownout, the kernel generates a halt.

Recovering from a hang

A hang occurs when an undetected condition causes system activity to stop, freezing every process on the system.

Verifying a hang

The following would indicate that your system is hung:

- You cannot log on.
- You get no response.
- The master console doesn't respond.

From a terminal or remote system, try logging in to the system that you suspect is hung. Use the following command format:

telnet *hung-system*

If you can log in, the system is not hung. But if you have problems, you can look for the cause. Refer to chapter 6 for isolating hung processes.

Generating a halt

You may be able to interrupt a hang by generating a halt, which then allows you to take a dump and reboot the system. Record the state of the LED indicators on the IOCs in the rear of the computer and attempt generate a halt:

- Press the ABORT button on the back of the first IOC board if your system has one. If your system does not have an ABORT button, try one of the following:
- Enter the hot-key sequence at the system console: hold down the Ctrl key while typing the following series of six bracket characters:

  ] [ ] [ ] [

- If the hot-key sequence fails to generate a halt, press the RESET button on your system. If the system is restored to the SCM (as indicated by the prompt SCM> on the system console) enter the following commands to generate your halt dialog:

  ```
  SCM> reset ↵
  SCM> start 1000 ↵
  ```

When the halt dialog is displayed, follow the steps in the section "Recovering from a halt during normal operation."

Recovering from a halt during normal operation

A halt is a condition that is detected by the kernel and indicates an internal software malfunction or inconsistency.

It is assumed that a halt policy is in place and a destination dump device has been defined. (For information on developing a policy for handling halts, see Chapter 4.) The lack of a system dump policy and a dump device can be a serious obstacle to finding and fixing system problems.

The following steps are the recommended procedure for recovering from a halt during normal operation:

1. Take a system dump.
2. Identify the halt code (if you have a hard copy or access to the file).
3. Reboot if possible (it usually is).

4. Create a diagnostic tape.
5. Prepare an STR.

These procedures are described in the sections that follow.

Taking a system dump

If the halt occurs during normal operation, you need to take a dump. How you perform the dump depends on your system configuration. You can dump directly to tape or a virtual disk set aside specifically for this purpose. Refer to Chapter 4 for information on setting up default halt behavior.

In general, you should limit the dump to kernel memory only, unless otherwise instructed by a Data General support representative.

Dumping directly to tape

The following dialog is displayed if your system has been configured to dump directly to tape:

```
The operating system has detected a serious error and halted.
Please record the following halt code:
"DG/UX R4.11MU03 halt code 30000027"

Do you want to take a system dump? [Y]
Dump destination device? [st(apsc(pci(0),B),2)]
```

If you want the dump to go to a different device than the default, type it in.

```
Limit the main memory dump to kernel memory only (no user
memory)? [Y]
```

> **IMPORTANT** Always go with the default, **y**, which instructs your system to dump only the kernel memory. If you enter **n** for no, the system will dump all the user and kernel memory, which is extremely time consuming and requires a large amount of physical space on your drive or disk.
>
> You should only enter no and opt for the full dump if instructed to do so by DG support personnel.

```
Mount tape.  Type newline when tape is ready.
dumping device: main_memory  size: 7708672 bytes
dumping device: MIPACKET  size: 2140 bytes
Tape volume 1 completed.
System dump completed successfully.

PLEASE REMEMBER TO APPEND THE CURRENTLY EXECUTING
SYSTEM IMAGE TO THE END OF THIS DUMP TAPE.
```

Dumping to a virtual disk

The following is an example of a dialog that would be displayed if the system has been configured to dump to a virtual disk named “halt”:

```
The operating system has detected a serious error and halted.
Please record the following halt code:
”DG/UX R4.11MU03 halt code 30000027”

Do you want to take a system dump? [Y]
Dump destination device?
vdm_dump(sd(npsc(pci(1),A,0,6),0,9),halt)]
```

If you want the dump to go to a different device than the default, type it in.

```
Limit the main memory dump to kernel memory only (no user
memory)? [Y]
```

> **IMPORTANT** Always go with the default, **y**, which instructs your system to dump only the kernel memory. If you enter **n** for no, the system will dump all the user and kernel memory, which is extremely time consuming and requires a large amount of physical space on your drive or disk.
>
> You should only enter no and opt for the full dump if instructed to do so by DG support personnel.

```
dumping device: main_memory  size: 36818944 bytes
System dump completed successfully.


PLEASE REMEMBER TO APPEND THE CURRENTLY EXECUTING
SYSTEM IMAGE TO THE END OF THIS DUMP TAPE.
```

Submit a dump and an STR to the Customer Support Center. Follow the instructions in the section titled “Creating a diagnostic tape for STR submission.”

Identifying halt codes

If you have access to the halt code file, look up the code to determine the condition that caused the halt. If you don’t have access to that file at the time of the halt, look up the halt code once your system is available again.

> **IMPORTANT** It is a good idea to keep a hard copy of this file available to refer to when your system halts.

Halt codes are documented in files in **/usr/release**. For example, to find the file containing information on halt code 2000075, execute the following commands:

```
# cd /usr/release ↵
# grep 2000075 *halt.codes ↵
```

This file, derived from comments in the kernel source code, contains a description of each halt code and the condition that caused the halt. You can read the indicated file using a command such as **view** or **more**, both of which offer commands for locating the desired text. For how to use these commands, see the **view**(1) and **more**(1) manual pages.

A search for the 02000075 halt code in the file **/usr/release/dgux_R4.10.halt.codes** shows this explanation:

```
VM_HALT_INVALID_KERNEL_ADDRESS          0x08003d 02000075
    An attempt was made to reference (load or store to) an
    invalid kernel address. **
```

The condition that caused the halt determines how you recover from the problem.

Rebooting the system

Reboot the system by your usual procedure.

Creating a diagnostic tape for STR submission

Depending on how you defined your halt policy for handling halts and configured your system, the dump will have been written to either a tape or a virtual disk set aside for this purpose. If the dump was not written directly to a tape, then you must transfer the dump from its other destination to the tape drive. You submit this tape to the Support Center along with a Software Trouble Report (STR) explaining the problem.

The diagnostic tape needs to be in the following format:

file 0	The system memory contents, called the system dump
file 1	The kernel executable, typically **/dgux**

If a tape device was the destination, then the system dump already occupies file 0 on tape. The tape may also include other files if you suspect that they may have contributed to the failure.

IMPORTANT The diagnostic tape must include both the system dump and the kernel. The tape is useless without both of these images.

Be careful not to overwrite files on the tape. You can prevent this by making sure that you do not use the rewind option in the commands to copy files to tape.

Transferring a dump to tape from a virtual disk (if necessary)

If your system has been configured to dump to a virtual disk, you must transfer the dump to a tape using the **lsd** command. For example, if your virtual disk is named **sys_dump**, use the following command line to dump it to the tape at **/dev/rmt/0**:

```
# lsd -t /dev/rdsk/sys_dump /dev/rmt/0n ↵
```

The **n** following the **/dev/rmt/0** device short name prevents the tape from being rewound. Refer to the **lsd**(1M) manual page for information on using the **lsd** command.

Copying the kernel executable

With the system dump written to tape, you are ready to copy the kernel executable to the tape.

IMPORTANT Be careful not to overwrite the system dump when you dump the kernel; if you overwrite the system dump, the Customer Support Center cannot diagnose your problem.

To make sure the tape is positioned at the end of the system dump (file 0), use the **mt** command to position the tape. See the **mt**(1) manual page for details on using the **mt** command. For example, to position the tape in tape drive **/dev/rmt/0** at the end of file 0, issue the following commands:

```
# mt -f /dev/rmt/0 rewind ↵
# mt -f /dev/rmt/0n fsf 1 ↵
```

If there is not enough room on the tape for the kernel executable, you may write it to a second tape instead.

By default, the kernel executable is **/dgux**. If the failure occurred with a different kernel, however, you should dump it instead of **/dgux**.

To copy the kernel, use the **cpio** command with the **–oBcv** options. For example, the following command line dumps /**dgux** to the tape at **/dev/rmt/0**, rewinding the tape when done:

```
# cd / ↵
# echo dgux | cpio –oBcv > /dev/rmt/0 ↵
```

IMPORTANT Do not try to use a shortcut by sending an absolute pathname (**/dgux**) to the **cpio** command. Instead, change to the directory containing the kernel executable file and then use the **echo** command on the kernel executable file (**dgux**, in this example).

If you have one of the information security products (B2 or C2) installed, use the **–Z** option with the **cpio** command to keep security attributes from being written to the tape.

Copying other files to tape

If you suspect that other programs or files may have contributed to your system's failure, you may include them on the tape as well. You may copy these files as tape files 2, 3, and so on. As with the kernel executable, copy the other files using the **cpio** command with the **–oBcv** options (or **cpio –oBZcv** if you have an information security product) and use relative path names.

Labeling the tape

After you have finished making the tape, label it with the name of your company, the cause of the failure, the date, and the contents of the tape. Your label might look like this:

```
Originator ID: 54012
BLUE DAEMON SYSTEMS, INC., Durham, NC
Halt code:  3400002
Date:  August 6, 1995
File 0:  system memory dump
File 1:  kernel executable
File 2:  miscellaneous files
Density:  QIC–525 tape at high density
cpio format:  cpio –oBcv
```

Preparing a Software Trouble Report (STR)

See Appendix E for an STR form. You also will find an STR form that is suitable for printing on a line printer in the file **/usr/release/STR_form** on your DG/UX system. STR forms are also available from the nearest Data General office, from your sales representative, or from the Customer Support Center. Depending on your service agreement, you can submit STR reports to the Center or directly to your sales representative. Mail the STR with related materials to this address:

Data General Corporation
Attn: STR Administrator
1626 Juergens Court
Norcross, Georgia 30093

When completing the STR, specify the product name as **DG/UX**, and the model number as **Q501A**. To complete the STR, check the information gathered with the useful commands identified in Chapter 2.

Recovering from other halts

The following sections describe procedures for halts that occur under special circumstances or during booting or shutdown.

Handling a halt during system booting or shutdown

Halts that occur early in system initialization or during system shutdown usually result from serious system failures, such as hard memory failures or an inability to load the kernel image properly during the boot process. When such a halt occurs, the standard kernel dump mechanism is not yet operational. Instead, debug screens display the general register contents and some kernel stack trace information generated around the time of the halt.

1. Press the Enter key several times to invoke the SCM (System Control Monitor) prompt.
2. Turn off power to the computer for five minutes and then turn the computer's power back on.
3. If the system is restored to the SCM (you will see the prompt `SCM>` on the system console), boot the system as you ordinarily do.

   If the halt does not recur and the system boots normally, the system may have experienced a memory error that was corrected by recycling power.

   If the problem recurs, record the general register contents and the kernel stack trace information that appears on your screen, and contact the Customer Support Center for help.

Handling a halt generated by the watchdog timer

Supporting selected AV models, the watchdog timer is designed to intercept hangs and convert them to halts or aborts depending on the configuration, allowing continued system operation. The watchdog facility is represented as a pseudo device that is configured automatically in the kernel. For details on the operation of the watchdog timer, refer to the section titled "Using the watchdog timer to detect and recover from system halts" in Chapter 4.

The system displays one of these halt codes to indicate that the watchdog timer has been triggered:

323000001
323000002
150000044

Following the display of any of these halt codes, perform these steps:

1. For halt code 150000044, if your system is configured to take a dump automatically or if the system prompts:

   ```
   Do you want to take a dump?
   ```

   Answer yes to take a dump and follow the instructions for preparing the dump in the section titled "Taking a system dump."

   If there is no prompt or your system did not automatically generate a dump, then your system hardware is incapable of taking a dump. For continued error recovery actions, proceed to the next step.

2. If the system ran power-up diagnostics and displayed the SCM prompt, reboot the system, according to the procedures you normally follow.

3. For these halt codes:

   323000001
   323000002

   after the system boots, deconfigure the watchdog timer either dynamically or rebuild the kernel, removing its entry from the system configuration file.

   If you deconfigure the watchdog timer dynamically, it will be deactivated for the remainder of your login session. If the watchdog timer remains in the system configuration file, it will be reactivated at boot time.

   ```
   Device –> Deconfigure
   ```

Specify the **wdt()** device name.

If you edit the system file, you must rebuild and reboot the kernel. For information on kernel building and booting, see *Managing the DG/UX® System*.

Await the recurrence of a hang, which triggered the watchdog timer-generated halt condition, and see "Recovering from hangs" in this chapter.

Handling a halt if AV/Alert is configured

The condition does not apply to releases prior to DG/UX R4.11.

1. If DGUX_MI_CALLOUT is enabled, and the system is connected to a modem, the Customer Support Center is notified automatically of the problem before the system runs power-up diagnostics.

2. If DGUX_MI_CALLOUT is disabled and the system is connected to a modem, an MI_CALL will be made automatically to the Customer Support Center after the reboot.

3. If the system is not connected to a modem, interrupt the MI_CALLOUT by typing Ctrl–C, or wait for the MI_CALLOUT to fail. Report the halt condition to the Customer Support Center.

   If DGUX_MI_CALLOUT does not successfully connect with the Support Center on the first try, it will retry the number of times selected as the default through the AV/Alert Service Manager Menu.

   If the SCM is restored, reboot the system.

4. Submit a Software Trouble Report, and proceed with a dump for the Customer Support Center to review. Follow the instructions in the section titled "Recovering from a halt during normal operation."

Restoring file systems after a system failure

A failure on a DG/UX system may not damage files on your system. Damage does occasionally occur, however, resulting in inconsistent file system metadata or lost data. The first time you boot your system after a failure, the DG/UX system performs operations to seek out and, where possible, repair damage to files and file systems. When the system cannot repair a file system problem, you must repair or restore the file system.

For the symptoms of a corrupt file system, boot messages that indicate file system problems, and how to repair or restore corrupt file systems, refer to Chapter 10.

End of Chapter

10 Fixing corrupt file systems with fsck

File systems contain complex sets of internal structures that enable the operating system to access and use the files and data stored in them. Rarely, but on occasion, during normal system operation, a file system can be corrupted or damaged. If a file system becomes corrupt, you can no longer use it safely. You must evaluate the problem and perform recovery actions to make the file system usable again.

To fully understand how a file system can be corrupted, you need to know the details of a file system's internal structure. If you are interested in learning how the DG/UX system manages files, ask your sales representative for the DG/UX Technical Brief on the DG/UX 5.4 File System. You can recover from this problem, however, without intimate knowledge of how file systems work. This chapter helps you to learn how to recognize symptoms of a corrupt file system and how to perform recovery actions.

These primary topics are covered:

- Symptoms of a corrupt file system
- Running the file system checker utility (**fsck**), which is pronounced f–s–c–k
- Checking for orphaned files in the **lost+found** directory
- Strategies for fixing corrupt **/** or **/usr** file systems

IMPORTANT **Fsck** can verify the structure of file systems that are formatted for use on the DG/UX system only. It cannot check file systems created by third-party database programs, or those on CD-ROMs, or other read-only file systems.

Symptoms of a corrupt file system

File system corruptions tend to occur when something irregular happens such as a power outage, a file system copy operation while the file system was mounted and being accessed, a disk head crash, or an abnormal system shutdown.

The primary sign of a corrupt file system is an error message during system booting.

System booting messages

During the boot process, the kernel determines whether or not each local file system can be mounted. If there is a problem with a file system, it invokes **fsck**. Three types of file system status messages are sent to the screen and to the **/etc/log/fsck.log** file (or **/etc/log/fsck_fast.log** if are you using the fast recovery version of **fsck**). Even if your system boots successfully, it is a good policy to check the log file for potential problems that you may not have caught during system booting.

File system can be mounted and requires no check

The following message indicates that the file system can be mounted.

```
Checking local file systems ....
```

The following message is written to the log file:

```
/dev/dsk/usr_opt_X11: No check necessary for
/dev/dsk/usr_opt_X11.
```

File system has recoverable problem

The following messages indicate that the file system needed to be checked by **fsck**, but eventually can be mounted.

```
Checking local file systems ....
```

The following messages are written to the log file:

```
/dev/dsk/usr_opt_fredware:  Superblock copies differ; using newer copy.
/dev/dsk/usr_opt_fredware:  File System is now mountable.

/dev/dsk/usr_opt_fredware:  1946 of 2000 blocks used (54 free); 11 of
254 inodes used (243 free).

/dev/dsk/usr_opt_fredware:  Time to fix /dev/dsk/usr_opt_fredware was 5
seconds.
```

File system is corrupt and cannot be mounted

When booting, the message “No space left on device” is an indication that you need to run **fsck**. This message will also appear if you try to mount the file system.

The recovery action you perform next depends on the type of file system that is corrupt:

- **/** (root) file system
- **/usr** file system
- file system that is neither **/** nor **/usr**

An example of a message with recovery actions is shown for each of the file system types.

Corrupt / file system — If either the **/** or **/usr** file system is unmountable, your system is inoperable. The following message indicates a corrupt **/** (root) file system:

```
'fsck –xlq' failed to fix /dev/dsk/root
   Root disk name (or q to quit)? [root]
```

The second line prompts for a new root disk name. You have two options:

1. If you have an alternate **root** virtual disk (named **root4.10**, for example), try booting it so that you may repair the corrupted **/** file system. Go to the section on running **fsck** for further procedures.

2. If you do not have an alternate **root**, type **q** to abort the booting process. For further recovery actions, go to the section on strategies for fixing a corrupt **/** or **/usr** file system.

Or, you may see this message, which is an indication of a corrupt **/etc/inittab** file.

```
... SAD autopush configuration failed ...
```

If you have a damaged **inittab** file, you must try to repair it. For further recovery actions, go to the section on strategies for fixing a corrupt / or /usr file system.

Corrupt /usr file system — If the **/** file system mounts successfully, but the **/usr** file system does not, the **init** program displays the following message, followed by a shell prompt:

```
init was unable to mount /usr.  You must manually fsck and
mount /usr.
#
```

From the shell, you may execute **fsck**. Go to the section on running **fsck** for details.

Corrupt file that is not / or /usr — If both the **/** and **/usr** file systems mount successfully, but a file system that is neither **/** nor **/usr** does not mount, your system is operable, but you cannot access and use these unmounted file system. The following message indicates such a problem:

```
 Checking local file systems ....
  Fsck failed to fix /usr/opt/fredware. It is not mountable.

  WARNING:  File system check (fsck –xlp) may have failed.
              See /etc/log/fsck.log for more information.
.
.
.
later in the boot process
.
.
.
    Mounting local file systems ....

    mount: /dev/dsk/usr_opt_fredware on /usr/opt/fredware: No space left
on device
        mount: giving up on: /usr/opt/fredware
```

Inspection of **/etc/log/fsck.log** shows messages like this:

```
/dev/dsk/usr_opt_fredware:  Superblock copy 1 is invalid.

/dev/dsk/usr_opt_fredware:  Superblock copy 2 is invalid.

/dev/dsk/usr_opt_fredware:  Fatal Error:  Cannot find a valid copy of
the superblock.
```

Run **fsck** to fix this problem. See the next section on running **fsck** for details.

Running fsck

IMPORTANT You may perform these operations using either **sysadm** or the shell. To escape to the shell from a **sysadm** prompt, type **!** and a shell prompt will appear.

It is assumed that you know the names of the file systems to be checked.

1. To find out the names of files that are listed in **/etc/fstab**, select the **sysadm** operation:

   `File System –> Local Filesys –> List`

   Alternatively, you may use the command:

   `admfilesystem –o list`

2. A file system must be unmounted before you can run **fsck** on it. Typically, the file system already will be unmounted. You can check a file system's mount status quickly with the **mount** command.

```
$ su ↵
# mount ↵
```

If you do not need to unmount the file system, go to step 4. Otherwise, go to the next step.

3. Unmount a file system with the **sysadm** operation:

   ```
   File System -> Local Filesys -> Unmount
   ```

   Alternatively, you may use one of these commands:

   admfilesystem -o unmount *mount-point*

   or

   umount *mount-point*

   For details on unmounting a local file system, see *Managing Mass Storage Devices and DG/UX® File Systems*.

   If you cannot unmount the file system, you may need to take the system down to single user mode, which attempts to unmount all file systems except the **/** and **/usr** file systems. Shut down your system using the following command:

   # **shutdown -g0 -y** ↵

   Refer to *Managing the DG/UX® System* for details on taking the system down.

   In single user mode, verify that all file systems are unmounted using the **mount** shell command. If there are any mounted file systems, unmount them with the **umount** command.

   mount

   umount *mount-point*

4. To invoke **fsck**, follow this path through **sysadm**, choosing the stand-among or stand-alone version, whichever form is appropriate:

   ```
   File System -> Local Filesys -> Check

   File System -> Check a File System
   ```

   Alternatively, you can use the command:

   fsck [*options*] [*file-system*]

   options are single character flags that modify the behavior of the command. Refer to the section on **fsck** options for details.

file-system refers to the file systems that you want to check. If you do not specify which file systems to check, **fsck** will check all file systems listed in the **/etc/fstab** file having a nonzero pass number and an "rw" or "ro" mounting status. For **fstab** file field descriptions, see *Managing Mass Storage Devices and DG/UX® File Systems.* If you specify multiple file names, **fsck** checks them in sequential order. You may specify a file system as a short name as shown in the **/dev/dsk** or **/dev/rdsk** files.

5. From **sysadm**, supply the name of one or more file systems to check. An example follows:

   ```
   File System(s) to Check: /usr/opt/fredware ↵
   ```

6. When you added to **/etc/fstab** the file system to be checked, you had a chance to tune the behavior of **fsck**. For information on tuning **fsck** behavior, see *Managing Mass Storage Devices and DG/UX® File Systems.*

   Go to the next section for details on selecting **fsck** options. Otherwise, press Enter to select the default **p** option, which detects all possible inconsistencies, but corrects only those that may result from an abnormal system halt.

   ```
   Fsck Options: ↵
   ```

Fsck options

You enable the option's function by selecting its single letter value. You must precede one or more options with a hyphen or you can combine the options **–p** and **–x** as follows: **fsck –px**. Table 10–1 lists the **fsck** options.

Table 10–1 Options to fsck command

Option	Description
–l	Checks file systems by performing a single pass through the file systems. This method does not check file systems in a particular order; it disregards the pass number that you assigned the file system when you added it to the **/etc/fstab** file. It will perform fast recovery **fsck** if you enabled the **fsck** logging features when you added the file system to the **/etc/fstab file**, and this is the first time that **fsck** has checked the file system since it was mounted. If fast **fsck** fails to repair the file system, multi-pass **fsck** is then run. Multi-pass **fsck** checks one or more file systems according to the pass numbers you assigned them when you added them to **/etc/fstab**. **Fsck** passes through each file system five times, each time checking on a different aspect.
–p	Detect all possible inconsistencies, but correct only those inconsistencies that may be expected to occur from an abnormal system halt. For each corrected inconsistency, one or more lines will be printed identifying the file system and the nature of the correction. Any other inconsistencies will cause the check of that file system to fail. **Fsck** corrects 15 types of inconsistencies for the specified file systems. See the next section for a list.
–q	Repair the inconsistencies listed under the **–p** option automatically, without asking for user approval. Unlike **–p** however, more serious inconsistencies will not cause **fsck** to fail; the user must still answer the resulting queries.
–y	Audit and repair all file system inconsistencies assuming a "yes" response to all questions asked by **fsck**.

Continued

Table 10–1 Options to fsck command

Option	Description
–n	Audit all file system inconsistencies, assuming a "no" response to all questions asked by **fsck**. This option also means that all file systems will be opened in read-only mode.
–x	Look at a file system's superblock to see if it is marked mountable. If so, do not check the file system for inconsistencies. If the file system is marked unmountable, check it.

The following options are mutually exclusive, you can use only one option from this set per **fsck** invocation: **–l**, **–p**, **–q**, **–y**, and **–n**.

Inconsistencies checked and corrected by fsck

1. An inode has an incorrect count of the blocks it uses. The count is corrected.
2. An inode is partially truncated. Partial truncation can occur if the system is abnormally halted while a file is being truncated, leaving the file claiming more data blocks than its size in bytes would require. The extra blocks are freed.
3. A directory has an incorrect child count. The count is corrected.
4. A directory entry exists for an unallocated inode. The directory entry is removed.
5. A directory entry's file name length is incorrect. The length is corrected.
6. An inode is unreferenced (has no directory entries anywhere in the file system). The inode is reconnected in the **lost+found** directory. Refer to the section about the **lost+found** directory for more information.
7. No **lost+found** directory exists, but an inode needs to be reconnected there. The **lost+found** directory is created.
8. The root directory needs to be expanded to make room for a **lost+found** directory entry. The directory is expanded.
9. The **lost+found** directory needs to be expanded to make room for a directory entry for **lost+found**. The directory is expanded.

10. An inode's link count is incorrect. The count is corrected.
11. The root control point directory's resource accounting (blocks, inodes) is incorrect. The counts are corrected.
12. A disk allocation region (DAR) has an incorrect free-block bitmap. The bitmap is corrected.
13. A DAR has an incorrect free-inode list. The list is corrected.
14. A DAR has incorrect summary counts of used blocks, inodes or directories. The counts are corrected.
15. The summary counts in the superblock are incorrect. The counts are corrected.

Fsck status messages

If **fsck** finds no errors, file system checking proceeds without any input from the user. If it finds a fatal inconsistency, it does no further checking on that file system; the **fsck** program either exits or proceeds to the next specified file system. If the **–p** option discovers an inconsistency and that error is one of those listed in the section, "Inconsistencies checked and corrected by fsck," or if the **–y** option discovers the inconsistency, **fsck** fixes the inconsistency without your intervention. Any other discoveries of inconsistencies require that you make a decision. The **fsck** program prompts with its recommended action. If you answer **yes**, **fsck** takes the recommended action. In the case of **–p**, **fsck** takes no damaging action without approval.

Appendix B lists all possible **fsck** error and advisory messages.

> **IMPORTANT** You may invoke **fsck** with the **–y** or **–n** option to grant advance approval or disapproval, respectively, of **fsck**'s recommended recovery action.

Typical output follows for a problem that **fsck** can fix without your intervention:

```
/dev/dsk/usr_opt_fredware: File system is now mountable.

/dev/dsk/usr_opt_fredware:  20 of 330 blocks used (310 free);
1 of 62 inodes used (61 free).

/dev/dsk/usr_opt_fredware:  Time to fix
/dev/dsk/usr_opt_fredware was 0 seconds.
```

Typical output follows for a problem that does require your intervention:

```
** Phase 1 – Check Blocks and File Sizes
** Phase 2 – Check Directory Contents
** Phase 3 – Check Connectivity
** Phase 4 – Check Link Counts and Resource Accounting

Inode 67 (owner: 2 [bin]; group: 2 [bin]; size: 52736 bytes;
type: Ordinary; mode: 755; mtime: Fri Nov 20 17:54:36 1987) has
incorrect link count (2 should be 1) –– fix?
```

If you respond **yes**, the **fsck** program corrects the incorrect link count that it found for inode 67 and moves on to Phase 5. After it finishes, it reports that the file system is mountable.

```
** Phase 5 – Check Disk Allocation Region Information
File system is now mountable.

13936 of 50000 blocks used (36064 free); 288 of 5822 inodes used (5534
free).

#
```

If you respond **no**, **fsck** does not fix the inconsistency and the file system remains unmountable.

The **fsck** program will refuse to check any file system for which any of the following conditions are true:

- The file system is mounted (except when you specified the **–n** option, which opens the file system read-only).
- The specified pathname (or its device node associate in **/etc/fstab**) is not a block-special, character-special, or regular file whose size can be determined.
- The special file associated with the file system cannot be opened.

For complete descriptions of **fsck** error messages and recovery actions, see Appendix B.

After **fsck** fixes a local file system, check the file system's **lost+found** directory for possible fragments of data that need to be recovered. If **fsck** does not fix the file system, **lost+found** is inaccessible.

Checking for orphaned files in lost+found directory

One of the functions of **fsck** is to locate files that have become disconnected from their parent directories. If the **fsck** utility cannot reconnect the file, it puts the file in a directory called **lost+found** in the file system's top-level directory.

For example, if you have a file system mounted at **/sales/accounts**, **fsck** puts any orphaned files in **/sales/accounts/lost+found**. The files in this directory have names that indicate the original file node number of the file that is now linked to the **lost+found** directory.

It is good practice to check the **lost+found** directory of a file system after **fsck** has completed successfully.

You may also use the **file** command to determine the nature of a file by classifying it as English text, data, binary executable code, and so on.

```
# cd /sales/accounts/lost+found ↵
# ls -l ↵
total 8
-rw-rw-rw-  1 willy  sales    3273 Sep 23 1991 #177431
# file #177431 ↵
#177431:  English text
```

You see that **fsck** found a file belonging to user **willy**. The **file** command classifies the contents as English text. Now you can tell user **willy** that one of his directories was damaged, and, using the file from the **lost+found** directory as a clue, he can begin to research what directory was damaged so that he can repair it or have it restored from backup.

Strategies for fixing corrupt / or /usr file systems

The files that compose the DG/UX system are stored in the required partitions **root** and **usr** mounted at **/** and **/usr**. Loss of either of these file systems means that the DG/UX system will not operate properly. A corrupt or missing **/** file system prevents the DG/UX system from booting. A corrupt or missing **/usr** will not prevent booting, but will prevent access to important utilities and system libraries after the system is booted.

If either the **/usr** or some other file system is corrupt but the **/** file system is not corrupt, your system will still run. In this case, you can work on the problem using **fsck** through **sysadm** or from the command line. Refer to the section on running **fsck** for instructions.

If the **/** file system is corrupt, you cannot run **fsck** because your system is inoperable. Your challenge is to reboot your system, specifying a boot device from which you can access **fsck**. If you cannot access **fsck** or if you can access **fsck**, but you still cannot recover, you may need to reinstall the entire DG/UX file system.

Select one or more recovery actions from the following list, as appropriate to your situation. These options are organized from least to most drastic.

- Boot stand-alone **sysadm** from the system disk or tape and run **fsck**
- Boot stand-alone **sysadm** from CD-ROM and run **fsck**
- Restore the **inittab** or **fstab** file from a backup file
- Reinstall the **/** or **/usr** file systems from system tape made with **systemtape** utility, if applicable
- Reinstall the entire DG/UX system

IMPORTANT Each option requires that you reboot your DG/UX system. The method for booting the system will depend on the type of DG/UX system you have. An example of each type of boot command follows:

```
SCM> b sd(ncsc(0),0,0)usr –f /stand/sysadm ↵

Boot comand: sd(npsc(pci(0),6)) usr –f /stand/sysadm ↵
```

Throughout the procedures given in the remaining sections of this chapter, a generic boot path will be provided. Use the specific boot path that is suitable to your system.

Boot stand-alone sysadm from the system disk or tape and run fsck

Stand-alone **sysadm** gives you access to the **fsck** utility and to a memory resident file system that contains many commands you can use for recovery. Refer to Appendix C for information on booting stand-alone **sysadm**.

1. You may boot stand-alone **sysadm** from the **/usr** file system (which contains stand-alone **sysadm**) on the system disk:

   boot-path **usr –f /stand/sysadm**

   or from tape:

   boot-path

   IMPORTANT If the **/usr** file system is corrupt or if the **/usr** file system is built on a virtual disk whose partitions span multiple physical disks, an attempt to boot stand-alone **sysadm** will fail. Your only option then is to boot it from tape.

2. Run **fsck** to fix this problem.

   Go to the section on running **fsck** for details. When prompted for **fsck** options, specify **–xlp**.

3. Take your system down and reboot your system.

   If **fsck** doesn't fix the problem, reboot stand-alone **sysadm** and execute **fsck** again, this time specifying the **–y** option. The **–y** option repairs all non-fatal flaws in the file system, even if the repair results in lost files or data. For further troubleshooting procedures, go to the section on restoring the **inittab** or **fstab** file from a backup file.

   If **fsck –y** fails and you have a release CD, try booting from it, as described in the next section. Otherwise, you must either partially or fully reinstall your system. Go to the appropriate section for other options.

Booting stand-alone sysadm from the CD-ROM and running fsck

1. To boot the CD, specify the **–R** option in the boot command line:

   bootpath **–R** ↵

   The **–R** option invokes a read-only installer kernel which allows you to mount from the release CD–ROM the **/** and **/usr** file systems which contain the shell commands that you can use for file recovery.

   IMPORTANT These commands have restricted capabilities which are different from those of a completely installed operating system. Do not replace commands in **/** or **/usr** with commands from this subset. Refer to Appendix C for a complete list of restricted shell commands.

2. Invoke **fsck** following this sysadm path:

   `File System –> Check –> File System`

   When prompted for **fsck** options, specify **–xlp**.

3. Take your system down and reboot your system.

   If the problem persists, reboot stand-alone **sysadm** and execute **fsck** again, this time specifying the **–y** option. The **–y** option repairs all non-fatal flaws in the file system, even if the repair results in lost files or data. For further troubleshooting procedures, go to the section on restoring the **inittab** or **fstab** file from a backup file.

If **fsck –y** succeeds and the system successfully boots the kernel, but booting appears to fail during the initialization phase, then either the **/etc/fstab** or **/etc/inittab** file may be corrupt:

- If you receive a message indicating that the **/** or **/usr** file system failed to mount, then you need to restore the **/etc/fstab** file from backup.
- If the **/** and **/usr** file systems mount, but the system fails to boot to run level i or run level 1, then you need to restore the **/etc/inittab** file from backup.

Restoring the inittab or fstab file from a backup file

The **inittab** file contains instructions that define the processes to be created or terminated for each run level during a system boot. A damaged **inittab** file prevents your system from booting. The **fstab** file lists both local and remote file systems to be mounted and available at system boot.

In this procedure, you will try to re-create your **inittab** or **fstab** file from a prototype file. If you booted from CD-ROM, skip the first three steps and go directly to step 4. Otherwise follow these steps:

1. Boot stand-alone **sysadm**, and select this operation from the main menu:

   ```
   Install Software –> Prepare Virtual Disks
   ```

2. The installer utility presents you with the default **root** and **usr** virtual disks to be mounted at **/** and **/usr**. Accept the defaults by pressing Enter (or by selecting **no** to modify the default values).

   IMPORTANT If you have alternate **root** and **usr** virtual disks (named **root410** and **usr410**, for example) that you wish to mount, respond **yes** and specify the alternate names when prompted.

```
2.  Prepare required virtual disks

Run this step now?  [yes] ↵
Required File System Mount Points:

File System    Virtual  Current   Action     Blocks  Physical
Mount Point    Disk     Blocks    Required   To Add  Disk
------------  --------  -------   --------   ------   --------
–none–         swap      50000      None        –    sd(insc(0),0,0)
/              root      40000      None        –    sd(insc(0),0,0)
/usr           usr       240000     None        –    sd(insc(0),0,0)

Modify this information? [no]
```

3. At the **sysadm** prompt, escape to the shell by entering **!**.

   ```
   Enter a number, a name, ? or <number>? for help, <NL> to
   redisplay menu or q to quit: [All Steps]: !
   ```

   The **/** file system is mounted as **/mnt/root**.

   Skip step 5, and go to step 6.

4. To boot the CD, specify the **–R** option in the boot command line:

   bootpath **–R** ↵

   The **–R** option invokes a read-only installer kernel and mounts automatically the **/** and **/usr** file systems from the CD-ROM. The booting process registers all physical disks in the configuration, including the disk containing the corrupted **/** file system that contains the damaged **inittab** or **fstab** files.

5. Create directories and mount the system's usual **/** and **/usr** file systems at temporary mount points.

   IMPORTANT If you booted from tape, skip this step and go to step 6.

   ```
   # mkdir /mnt/root ↵
   # mkdir /mnt/usr ↵
   # mount /dev/dsk/root /mnt/root ↵
   # mount /dev/dsk/usr /mnt/usr ↵
   ```

   With access to operational **/** and **/usr** file systems, you may restore the corrupted **inittab** or **fstab** files with prototype files.

6. Change your directory to **/mnt/root/etc**.

   ```
   # cd /mnt/root/etc ↵
   ```

7. Copy the corrupted file to another file so that you can refer to it or fix it later.

   ```
   # cp inittab inittab.corrupt ↵
   ```

   or

   ```
   # cp fstab fstab.corrupt ↵
   ```

8. Replace the corrupt file with a backup or prototype:

   - To fix **/etc/inittab**, look in **/mnt/root/etc** for **inittab.backup** or **inittab.proto**. If there is a **.backup** version, copy that file into **inittab** (the **inittab.backup** file was created when you changed the **inittab** file with **sysadm** and is more current than the default **inittab.proto** file). If there is no **.backup** version of **inittab**, copy the **.proto** version.

```
# cd /mnt/root/etc ↵
# ls inittab.backup inittab.proto ↵
inittab.backup    inittab.proto
# cp inittab.backup inittab ↵
```

- To fix **/etc/fstab**, look in **/mnt/root/etc** for **fstab.backup** or **fstab.proto**. If there is a **.backup** version, copy that file into **fstab** to **/mnt/root/etc/fstab** (the file **fstab.backup** was created when you changed the **fstab** file with **sysadm** and is more current than the default **fstab.proto** file). If there is no **.backup** version of **fstab**, copy the **.proto** file.

  ```
  # ls fstab.backup fstab.proto ↵
  fstab.backup    fstab.proto
  # cp fstab.backup fstab ↵
  ```

  IMPORTANT Although the ***.backup** or ***.proto** file will be adequate for booting your system, it may not be sufficient for continued use. After you boot, you may need to customize **inittab** or **fstab**. Be sure to protect yourself from corrupting this file again. Keep a backup.

9. If you booted from CD-ROM, go to step 10. Otherwise, since you booted from tape, exit the shell, quit **sysadm**, and reboot the repaired DG/UX system.

   ```
   # exit ↵

   Enter a number, a name, ? or <number>? for help, <NL> to
   redisplay menu or q to quit: [All Steps]: q ↵
   ```

   bootpath

   If you still cannot boot your system, you have two more options, which are covered in the next two sections.

10. If you booted from CD-ROM, halt the system, and reboot the repaired DG/UX system.

    ```
    # halt -q ↵
    ```

    bootpath

    If you still cannot boot your system, you have two more options, which are covered in the next two sections.

Reinstalling the dgux package from a system tape made with the systemtape utility

A system tape contains a snapshot of the **root** and **usr** virtual disks taken at the time the tape was created. DG/UX 5.4R3.00 introduced the **systemtape** utility to allow you to create a bootable tape of your operating system. To use this procedure, it is assumed that you already have created a system tape. Instructions for creating a system tape are in *Managing the DG/UX® System* and the **systemtape**(1M) manual page.

To restore the contents of the system tape, you boot the tape and perform only the first three installation steps through stand-alone **sysadm**.

1. Insert the system tape in the appropriate drive.
2. Boot the system tape from the appropriate drive.
3. From stand-alone **sysadm**, select option 4, `Install Software.` Perform only the first three operations:

   ```
   1 Prepare physical disks
   2 Prepare virtual disks
   3 Load software
   ```

4. Instead of loading multiple packages as you ordinarily do when performing a complete installation, load only one package — **dgux**.
5. Since the **dgux** package does not require setup, and you already have a bootable kernel, skip **sysadm** menu choices `4 Set up Software` and `5 Build Kernel`. But, since you must reboot the system, be sure to pick `6 Reboot kernel.`

If you still cannot boot your system, you have one more option: a complete reinstallation, as discussed in the next section.

Reinstalling the DG/UX system

If all of these options fail, you must reinstall the DG/UX system completely.

1. Shut down the system.

   ```
   # cd / ↵
   # shutdown -g0 -y ↵
   # halt -q ↵
   ```

2. Insert the release medium into your computer drive.

3. Boot the medium, using the appropriate boot command. Typical examples for a tape and CD follow:

   From tape or CD-ROM:

   bootpath

4. From the Standalone Sysadm Main Menu, select `Install Software`.

   Refer to *Installing the DG/UX® System* for details.

   After you have reinstalled the DG/UX system, you may replace any files you wish by restoring them from backup. For information on restoring selected files, refer to Chapter 11.

End of Chapter

11 Frequently asked questions

How do I restore files created with the dump and dump2 commands?

Before you restore backup file systems to your disk, determine whether you have sufficient disk space to accommodate the original file system plus the backup. You may have problems with disk space if you are restoring large file systems or large single files. Since you will be restoring a file or file system to a file system, check available file system space with the **admfsinfo**, **df**, and **du** commands, which are described in Chapter 3.

Restoring a single file from tape

IMPORTANT Use this procedure only if you used the **dump** or **dump2** command to back up the file system that you want to restore.

1. Establish the destination for the dump files by going to the **/tmp** directory where you will put them temporarily .

   # **cd /tmp** ↵

2. Issue the **restore** command; for example:

   # **restore if /dev/rmt/0** ↵

   where:

   i is interactive mode. You can search through the dump tape and look for files to restore.

   f specifies a nondefault archive name. In this example, **/dev/rmt/0** is the archive name.

3. At the `restore>` prompt, you can use the following commands:

ls	List directory contents, or just filenames. (Do not use the **ls** options given in the **ls**(1) manual page.) Files that have been added to the extract list are marked with an asterisk (*).
cd	Change directory. The **/** directory on the tape is the file system's mount point directory (the directory from which you made the backup). Make sure you specify a pathname relative to the top-level root directory when specifying the desired file system. For example, assume that your backup tape contains the **/usr/opt/finance** file system. Since **/usr/opt/finance** is the top-level root directory, you would use the command **cd** *directory* instead of **/usr/opt/finance/***directory* to change to the desired directory.
pwd	Print working directory.
add	Add filename to the list of files to be extracted.
delete	Delete filename from the list of files to be extracted.
extract	Extract requested files.
quit	Exit program.
help	Print list of files to be extracted.

For example, to restore the file **/usr/opt/finance/reports/january**, change to the appropriate directory, if necessary:

```
restore> cd /reports ↵
```

Verify that the file exists, and add it to the list to be extracted:

```
restore> ls january ↵
january

restore> add january ↵

restore> ls january ↵
*january
```

Files to be extracted are preceded by an asterisk (*). Use the **extract** command to copy the file **january** from tape to the **/tmp** directory on disk.

```
restore> extract ↵

You have not read any tapes yet. Unless you know which
volume your file(s) are on, you should start with the last
volume and work towards the first.
Specify next volume #: 1 ↵
Set owner/mode for '.'? [yn] no ↵
```

Answer **no** to the preceding prompt asking whether or not you want to copy the file system's mode (access permissions) from the root file system on tape to the file system in /**tmp**.

4. Type quit at the restore prompt when finished.

```
restore> quit ↵
```

5. List the **/tmp** directory.

```
# ls -p /tmp ↵
accounts      claims      reports/
```

6. Change to the **reports** directory and list its content:

```
# cd reports ↵
# ls ↵
january
```

The file on tape has been copied successfully to **/tmp**.

Restoring a file system from tape

1. Create a virtual disk for the file system that is being restored. To control the exact placement of the virtual disk on the physical disk, specify the desired size using the following format example:

```
# admvdisk -o create -P 'sd(ncsc(0),0,0):*:40000'
usr_opt_finance ↵
```

A virtual disk named **usr_opt_finance** is sized at 40,000 blocks.

2. Create the file system, and mount and add the mount point to the **/etc/fstab** file as follows:

```
# admfilesystem -o create -q 'usr_opt_finance' ↵
# admfilesystem -o add -t dg/ux -f 'usr_opt_finance' -x
/usr/opt/finance↵
```

3. Establish the destination for the file system by going to its directory. For example,

```
# cd /usr/opt/finance ↵
```

4. Issue the **restore** command; an example is shown as follows:

```
# restore  rf  /dev/rmt/1 ↵
```

I just backed up a file, and restored it, but now cant find it. Where is it?

If you use the **restore** command, the file or file system is copied from tape to disk at the current directory. Before you begin the restore operation, you explicitly must change your directory to the desired location for the restore. We recommend that you change directory to **/tmp**. If you fail to do so, the file or file system will be restored in the current directory, which is wherever you currently are positioned.

The behavior of a restore operation depends on the particular command — **tar**, **cpio**, **dd**, and **dump2** — used to back up the file or file system to tape. You need to understand fully the behavior of commands that you use. Although you believe something has gone wrong, the command probably performed in a predictable manner. Make sure you know the destination for the restore operation before you perform the restoration.

An example of a common misunderstanding about file system backup is the structure of the files being backed up. Typically, the hierarchy of the file system being backed up is retained on the tape. Consequently, the file structure is restored to your disk.

What does kernel building really mean?

Kernel building comprises four separate operations:

Configure — Editing, by hand with a text editor, or generating automatically two system configuration files: **/var/Build/system.params.** *hostname* and **/var/Build/system.device.** *hostname*. The former file contains kernel and software package tunable parameters; the latter contains device configurations. The two files are then combined to form **/var/Build/system.** *hostname*.

Build — Invoking various utilities and libraries to compile the system configuration file to produce an executable file named **/var/Build/dgux.** *hostname*.

You may build multiple different kernels in **/var/Build** that are not linked to **/dgux**. Such kernels are not installed or linked. An example of how to boot such a kernel follows:

```
SCM> b sd(ncsc(0),0) root -f /var/Build/dgux.test ↵

Boot command: sd(npsc(pci(0),0)) root -f
/var/Build/dgux.test ↵
```

Install — Moving the executable from **/var/Build/dgux.** *hostname* to **/dgux.***hostname*.

Link — Linking the executable to **/dgux** for booting. An example of how you to boot such a kernel follows:

```
SCM> b sd(ncsc(0),0) root -f /dgux ↵

Boot command: sd(npsc(pci(0),6)) root -f /dgux ↵
```

After the kernel is built, it must be booted to activate it. A kernel can be booted as part of the installation process, through stand-among **sysadm**, and the SCM.

For more information on building and booting kernels, refer to *Managing the DG/UX® System*.

How do I file an STR?

For information on filing an STR, refer to the section titled “Preparing a Software Trouble Report (STR)” in Chapter 9. See Appendix E for the STR form itself.

What is the difference between expanding a virtual disk and expanding a DG/UX file system?

You expand a virtual disk when it does not contain a DG/UX file system. You expand a file system when a virtual disk does contain a DG/UX file system. The expand file system operation will expand both the virtual disk and the file system.

Expanding a virtual disk increases disk space in physical blocks. If the virtual disk contains a DG/UX file system, it cannot use the additional blocks until the new space has been remapped through the **admfilesystem** operation.

If you are unsure whether a virtual disk contains a DG/UX file system, use the **admfilesystem** command to list the file systems on your system. An example follows:

```
# admfilesystem –o list ↵
/dev/dsk/root         /             dg/ux    rw    x     0
/dev/dsk/usr          /usr          dg/ux    rw    x     0
/dev/dsk/usr_opt_X11  /usr/opt/X11  dg/ux    rw    x     1
/dev/dsk/usr_opt_acct /usr/opt/acct cdrom    r     x     1
```

From this listing, you see the virtual disks containing DG/UX file systems, which can be enlarged with the following expand file system command format:

admfilesystem –o expand *method 'mount directory'*

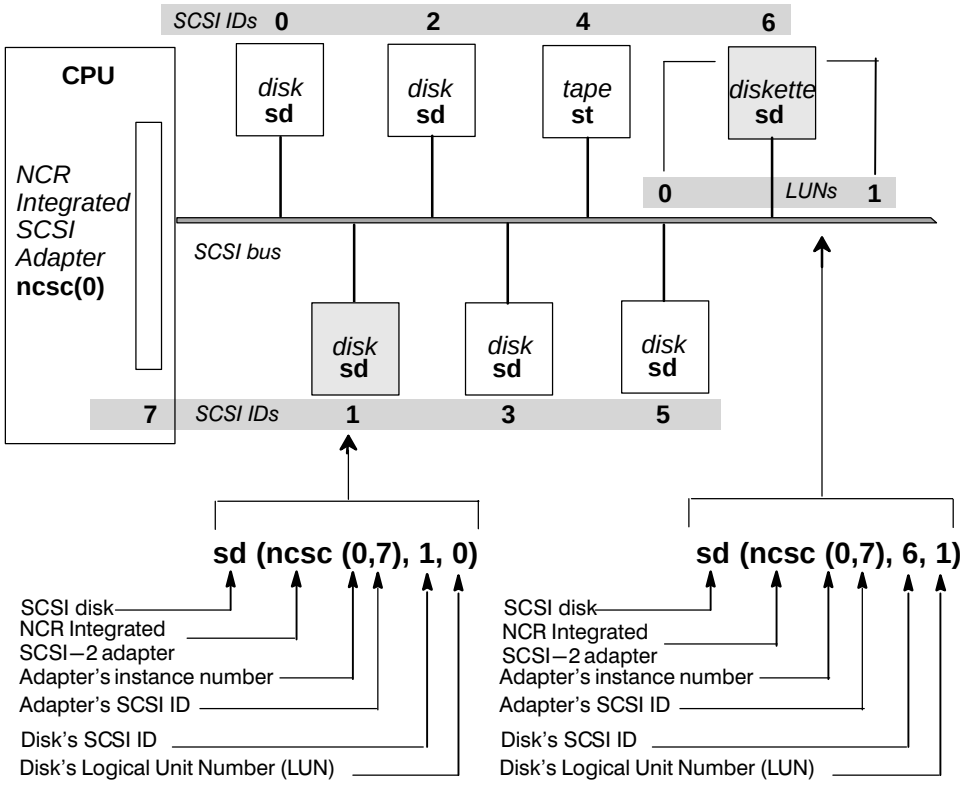
See *Managing Mass Storage Devices and DG/UX® File Systems* for information on expanding DG/UX file systems.

How can I find out what a device's name means?

A device's DG/UX name reflects the simplicity or complexity of your DG/UX configuration. You can construct a simple DG/UX device name by examining its context in the hardware configuration and answering the following questions:

Question	Sample Answer
What driver does the device use?	**sd** (SCSI device driver)
What type of controller or adapter is the device connected to?	**ncsc** (NCR integrated SCSI-2 adapter.
What is the instance number of the controller or adapter?	**0** This is the first SCSI (**ncsc**) adapter in the configuration.
What is the SCSI ID of the controller or adapter?	**7** (This is preset at the factory. Check your inventory list or look at the rear of the CPU.) To find out if the SCSI ID has changed, check the system's SCM .
What is the SCSI ID of the device?	**0** (This is preset at the factory. Check your inventory list or look at the rear of the device.)
What is the logical unit number (LUN) of the device?	A LUN is used to differentiate devices sharing a single device SCSI ID; for example, diskette drives, optical disks, and disk arrays. For a non–removable disk, the LUN is always **0**.

The following example illustrates how you might derive a simple DG/UX device name from the hardware configuration.



For more information on device naming, refer to *Managing Mass Storage Devices and DG/UX® File Systems*.

How can I find out if a device has been configured?

Use the **sysdef** command to list the names of devices that are listed in the kernel, **/dgux**.

```
# sysdef ↵
# Configured devices
#
kbd()
grfx()
lp()
...
sd(ncsc(),0)
st(ncsc(),4)
st(ncsc(1),*)
...
```

Regardless of a device's inclusion in the **sysdef** command output, a device still may fail to configure at system boot time. A boot message will note devices that fail to configure. If you do not see this message, Use the **admdevice** command to list the names of configured and registered devices.

If you suspect that a device is not configured, try to configure it explicitly. Examples follow:

```
# admdevice –o configure ”kbd()” ↵
Warning: Device ”kbd()” is already configured
```

```
# admdevice –o configure ”sd(ncsc(0,7),1,0)” ↵
Error: Cannot configure device ”sd(ncsc(0,7),1,0)”
No such device or address.
Configuration of supporting SCSI adapter device failed.
```

This message implies that there is a problem with the SCSI adapter **(ncsc())**.

To find out what disk devices are registered, use the following command:

```
# admpdisk –o list ↵
Disk name          State   Reg? Format Total blocks  Free blocks
sd(insc(0),0,0)    avail      y vdisks      1295922      1230279
```

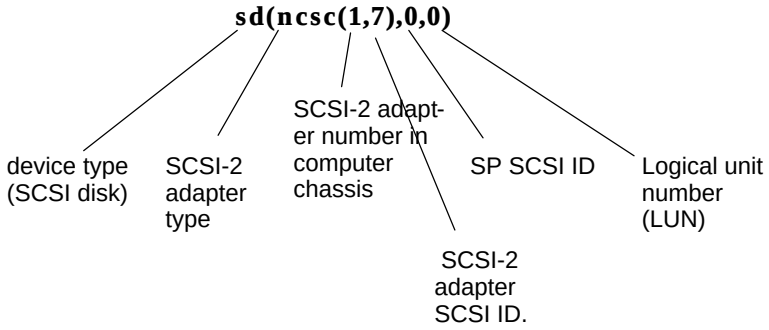
For more information on device naming, refer to *Managing Mass Storage Devices and DG/UX® File Systems*.

How do I find out the DG/UX device names for the physical disks in my CLARiiON disk array?

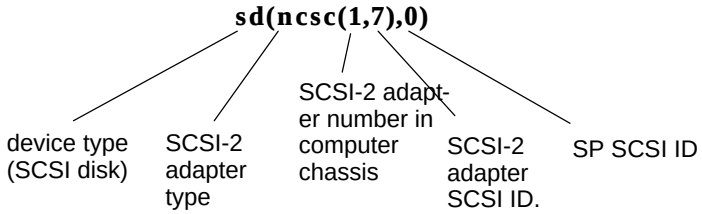
You need to know the following facts:

1. The type of SCSI-2 adapter card (for example, **ncsc**) connected to the CLARiiON.
2. The number of the SCSI-2 adapter card in the computer chassis (the first card of its type is number 0, the second of that type number 1, and so on).
3. If this CLARiiON will be used in a dual-initiator (dual host, shared bus) environment, the SCSI ID of the SCSI-2 adapter in each host; the SCSI IDs must be different.
4. The SCSI ID of the SP (system-control processor) in the CLARiiON that you will use to bind the physical disk.
5. The physical disk number (LUN number) the disk will receive when you bind it (by default the first disk you bind is 0, the second 1, and so on, in hex, up to F in a CLARiiON chassis).

Along with the letters sd (for SCSI disk), these facts will help you form the CLARiiON physical disk device names, as follows.



DG/UX **sysadm** software can help you with the name. If you have already bound the CLARiiON disks, you can create a kernel and select Auto Configure, to have **probedev** generate the root part of all device names on every SP, and you can use this as a model for the other unit names. For example, **probedev** creates the following entry for the system kernel file:



Then on each host that would use this CLARiiON, you would build a new kernel and manually create as many device names as needed for all the physical disks, complete with the disk unit numbers. For example, the device name entries based on the preceding name may look like this:

```
sd(ncsc(1,7),0,0)
sd(ncsc(1,7),0,1)
sd(ncsc(1,7),0,2)
sd(ncsc(1,7),0,3)
```

> **IMPORTANT** Omitting the physical unit number will cause your host to take over all disks in the CLARiiON bound by the indicated SP at startup — not a desirable result if two hosts share a storage-system, since it will prevent the other host from using any CLARiiON disks.

Learning disk unit names when DG/UX is running the CLARiiON disks

The CLARiiON disks are already supported by the kernel, registered, and mounted.

If you completed the disk planning worksheet, you can simply look at your worksheet. If there is no completed worksheet handy, you can list the **pdsk** devices in **/dev** and try to learn the disk/device correspondence from this, as follows.

For all disks except CLARiiON disks and diskettes, the physical disk unit (LUN) field of a disk's DG/UX device name is always 0. Therefore if a **pdsk** device listing ends with a number other than 0, you can assume this is a CLARiiON disk. Then you can use the CLARiiON disk device names as a model to determine which device name belongs to CLARiiON disk unit 0. For example,

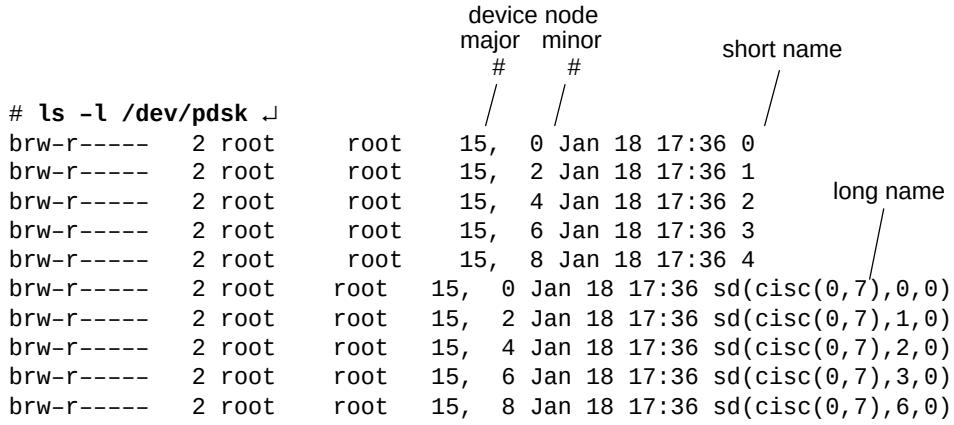
```
#  ls  /dev/pdsk/sd* ↵
0        /dev/pdsk/sd(ncsc(0,0),0,0)
1        /dev/pdsk/sd(ncsc(0,7),0,0)
2        /dev/pdsk/sd(ncsc(0,7),0,1)
3        /dev/pdsk/sd(ncsc(0,7),1,2)
3        /dev/pdsk/sd(ncsc(0,7),1,3)
```

CLARiiON physical disks

The physical disk unit number is the one assigned when disk was bound (displayable by the CLARiiON utility program that bound the disk).

If I know a DG/UX device long name, how do I find out its short name?

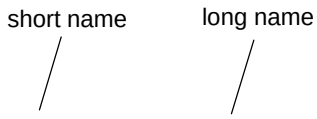
If you know the DG/UX disk device's long name, list the **/dev/pdsk** directory with the **ls –l** command to find out its short name:




```
# ls -l /dev/pdsk ↵
brw-r-----   2 root     root    15,  0 Jan 18 17:36 0
brw-r-----   2 root     root    15,  2 Jan 18 17:36 1
brw-r-----   2 root     root    15,  4 Jan 18 17:36 2
brw-r-----   2 root     root    15,  6 Jan 18 17:36 3
brw-r-----   2 root     root    15,  8 Jan 18 17:36 4
brw-r-----   2 root    root   15,  0 Jan 18 17:36 sd(cisc(0,7),0,0)
brw-r-----   2 root    root   15,  2 Jan 18 17:36 sd(cisc(0,7),1,0)
brw-r-----   2 root    root   15,  4 Jan 18 17:36 sd(cisc(0,7),2,0)
brw-r-----   2 root    root   15,  6 Jan 18 17:36 sd(cisc(0,7),3,0)
brw-r-----   2 root    root   15,  8 Jan 18 17:36 sd(cisc(0,7),6,0)
```

Locate the device's long name in the column designated as "long name" — for example, **sd(cisc(0,7),0,0)**. Then locate the device's corresponding device node to the left in the same row; in this case, **15,0**. Next, locate another row with the same device node. In this example, the device node in the top row matches that of the sixth row. You can infer that device name **sd(cisc(0,7),0,0)** corresponds to short name **/dev/pdsk/0**.

For tape devices, the short names in **/dev/rmt** are symbolic links to the long names. List the **/dev/rmt** directory to determine the correspondence between long and short names.




```
# ls -l /dev/rmt ↵
total 4
lrwxrwxrwx 1 root root 27 Jan 18 17:36 0 -> st(cisc(0,7),4,0)
lrwxrwxrwx 1 root root 28 Jan 18 17:36 0n-> st(cisc(0,7),4,0)n
lrwxrwxrwx 1 root root 27 Jan 18 17:36 1 -> st(cisc(0,7),5,0)
lrwxrwxrwx 1 root root 28 Jan 18 17:36 1n-> st(cisc(0,7),5,0)n
```

As shown in the listing, tape device **st(cisc(0,7),4,0)** is associated with the short name **0**. You can infer that the device name **st(cisc(0,7),4,0)** corresponds to short name **/dev/rmt/0**.

Alternatively, if you know only the device's short name, you can find out its long name with the **ls** command and the **–L** option. An example follows:

```
# ls -L /dev/rmt/1 ↵
/dev/rmt/st(cisc(0,7),5,0)
```

The output indicates that short name **1** is linked to long name **st(cisc(0,7),5,0)**.

When do I use a DG/UX device long name or a short name?

Use a device's long name under these circumstances:

- To specify the boot device for the DG/UX system and other bootable software at the SCM prompt.
- To operate on a physical disk — list, register, deregister, copy, configure, deconfigure, convert, and soft format. Refer to *Managing Mass Storage Devices and DG/UX® File Systems* for information on these operations.
- To specify physical disk devices on which you create virtual disks, mirrors, and caches.
- To recognize devices that are configured into the kernel and listed in the system file.
- To identify a device that may be experiencing hardware failures.
- To identify devices listed in log files.
- To perform operations using physical disks and tapes in single user mode.
- To specify the destination device for a system dump.

Use a device's short name under these circumstances:

- To specify mount points for file systems on these devices: CD-ROM, DOS diskettes, and memory file systems.
- To make DOS or DG/UX file systems on diskettes.
- To recognize the names of configured devices that are created in the **/dev** directory after the kernel is booted.

How can I identify a disk, CD-ROM, or diskette device on a physical disk listing of device names?

If you do not know the device's name, you may be able to derive it from the physical disk listing. An example follows:

```
Disk name          State   Reg? Format Total blocks  Free blocks
sd(ncsc(0,7),0,0)  avail      y vdisks       663476        81143
sd(ncsc(0,7),1,0)  avail      n n/a            2812          100
sd(ncsc(0,7),3,0)  avail      n n/a           63554          n/a
```

Diskette and CD-ROM devices are assigned default SCSI IDs at the factory, shown as follows:

Default SCSI ID for Diskette	Default SCSI ID for CD-ROM
1	4

A SCSI ID is specified in the third numeric field in a device name. Assuming that these device SCSI IDs have not been altered, you reasonably can assume that the second and third device names are for a diskette and CD-ROM device, respectively.

Another clue to identify a device's type is its registration status and format type. Only devices containing virtual disks, typically disks, require registration. From the preceding listing, the "y" registration status and the "vdisks" format strongly indicate that the first device name is for a disk.

From this listing, you can reasonably guess that the second device is a DOS-formatted diskette, and that the third is a CD-ROM device containing a read-only file system in High Sierra (or **cdrom**) format.

If the diskette had been formatted for DG/UX file systems, it would be listed as follows:

```
Disk name          State   Reg? Format Total blocks  Free blocks
sd(ncsc(0,7),1,0)  avail      y vdisks         2812          100
```

CD-ROM media can contain not only ISO 9660 and High Sierra-type file systems, but also the DG/UX or DOS-type file systems.

How do I mount a High Sierra, DG/UX or DOS–type file system for a CD-ROM device?

Consider these issues:

1. Find out the CD-ROM device's node name. Go to the **/dev/pdsk** directory to locate the long device name, which is associated with the short name. Short name name **/dev/pdsk/2** is an example.
2. Find out the CD's file system type. File system types are: **cdrom** (for High Sierra, ISO 9660, and Rock Ridge), **dgux**, and **dos**. To find out the CD's file system type, read the CD's label, or check its release notice or accompanying literature.

3. Decide whether you want to mount the file system temporarily (for copying files, for example) or permanently, making the file system available for use each time the system boots. You make a file system permanently available by putting an entry in the **/etc/fstab** file (for a database, for example).

   If you intend to swap media in and out of the drive frequently, you probably do not want to add the file system to the **fstab** file. If, however, you expect the medium to remain in the drive for continuous access, you probably want to add the file system to the **fstab** file.

For more information on issues 1 and 2 in the preceding list, refer to the previous three sections in this chapter .

Procedures follow for creating a temporary mount point and a permanent mount point.

Temporarily mounting a file system

Follow these steps to mount for temporary use a file system of type **dgux**, **dos**, or **cdrom**:

1. Create a mount point for the file system. Mount point **/cdrom** is used in this example.

   ```
   # mkdir /cdrom ↵
   ```

   IMPORTANT To mount multiple file systems of type **dgux** from the DG/UX operating system — **root**, **usr**, **usr_opt_X11**, or **usr_opt_networker** — go to steps 2 and 3. To mount all other file system types, go directly to step 4.

2. If you intend to mount, consecutively, multiple DG/UX file systems from the CD, register the CD-ROM device. Registration enables the operating system to recognize multiple DG/UX file systems on a medium.

   ```
   # admpdisk –o register 'sd(ncsc(0,7),3,0)' ↵
   ```

3. List the file systems on the CD.

   ```
   # ls /dev/dsk+* ↵
   +root
   +usr
   +usr_opt_X11
   +usr_opt_networker
   ```

   All file systems composing the DG/UX operating system begin with a plus sign (+).

4. Mount the desired file system. The syntax of the **mount** command varies according to the type of file system being mounted.

Table 11–1 Temporary mount points for CD–ROM devices

File system type	Mount command example
DG/UX (multiple file systems from the DG/UX operating system)	mount –o ro /dev/dsk/+root /cdrom
dgux	mount –o ro /dev/pdsk/2 /cdrom
dos	mount –t dos –o ro /dev/pdsk/2 /cdrom
High Sierra, ISO 9660, Rock Ridge (cdrom)	mount –t cdrom /dev/pdsk/2 /cdrom

If the device is unregistered, you may supply the device's node name in the **mount** command. For file systems of **dgux** type composing the DG/UX operating system, supply the single file system's name in the form **/dev/dsk/***file–system–name***+**.

5. Verify a successful file system mount, and list the mount point directory's contents.

```
# admfilesystem –o list –m ↵
/cdrom
# ls –l /cdrom ↵
Directory structure at the mount point
```

6. The CD must remain in the drive while you use the mounted file systems. When you no longer need access to the file system, unmount it and deregister the device before removing the CD from the drive.

```
# admfilesystem –o unmount '/cdrom' ↵
# admpdisk –o deregister 'sd(ncsc(0,7),3,0)' ↵
```

Creating a permanent file system mount point in /etc/fstab

For continuous use of a file system on a CD, you may want to create a permanent file system mount point in the **/etc/fstab** file. Its location there ensures that the file system will be mounted each time the system boots. A read-only database stored on a CD is a good candidate for inclusion in **/etc/fstab**. Keep in mind that the CD containing the file system being mounted must remain in the CD-ROM drive. Its accidental removal or replacement with another CD of a different file system type will result in booting errors.

Follow these steps to create a permanent file system mount point in the **/etc/fstab** file for these file system types: **dgux**, **dos**, or **cdrom**:

Mount the desired file system. The syntax varies according to the file system type being mounted.

Table 11–2 Permanent mount points for CD–ROM devices

File system type	Mount command example
dgux	**admfilesystem –o add –f /dev/pdsk/2 –t dgux –p ro –x /cdrom**
dos	**admfilesystem –o add –f /dev/pdsk/2 –t dos –p ro –x /cdrom**
cdrom	**admfilesystem –o add –f /dev/pdsk/2 –t cdrom –p ro –x /cdrom**

How should I mount a file system on a removable medium for temporary use?

To temporarily mount a file system, do not add (with the **admfilesystem –o add** operation) the mount point to the **/etc/fstab** file. File systems listed in the **fstab** file are mounted each time the system boots. Instead, simply create a mount point and mount the file system for immediate access. The file system will not be available on subsequent system reboots. Be sure that you insert the medium in the drive and close the drive door before you issue any commands.

An example of creating a mount point and mounting a single cdrom-type file system on a CD-ROM device follows:

```
# mkdir /cdrom ↵
# mount  -t cdrom  /dev/pdsk/2  /cdrom ↵
```

An example of creating a mount point and mounting a DOS-type diskette follows:

```
# mkdir /diskette ↵
# mount  -t dos  -o ro  /dev/pdsk/2  /diskette ↵
```

To remove the medium from the drive, make sure you unmount the file system first. An example follows:

```
# umount /cdrom ↵
```

Removing a medium without unmounting the file system results in an error message.

How do I mirror the operating system?

You mirror the DG/UX operating system — **root** and **usr**, and optionally **usr/opt/X11**, and **/usr/opt/networker** — just as you would any other virtual disk. Before you create the mirror, make sure that you have two or more virtual disks to form the mirror with. In the following example, virtual disks **root** and **root2** already exist.

```
# admvdisk –o insert –t mirror –n 'root1''root' ↵
# admvdisk –o link –c 'root2' 'root' ↵
# admvdisk –o sync 'root' ↵
```

Repeat these steps for other virtual disks comprising the operating system.

For details on mirroring a virtual disk, refer to *Managing Mass Storage Devices and DG/UX® File Systems*.

To boot a virtual disk containing the operating system, be sure that you specify the name of the mirror image instead of the mirror itself. Providing the mirror name results in a failure on m88k systems. An example of how to specify the bootable image follows:

```
SCM> b sd(cisc(0),0,0) root2 –f /dgux –i 3 ↵

Boot command: sd(npsc(pci(0),0)) root2 –f /dgux –i 3↵
```

The mirror images are named **root1** and **root2**; the mirror is named **root**.

IMPORTANT On Intel systems, you *can* boot directly from a mirror. To boot directly from a mirror, you first name the mirror virtual disk on the command line, for example, **root**. The bootstrap locates the best available image for the mirror on the specified physical disk. The system uses this image to load the DG/UX kernel. When the kernel registers the disks, all the images of the mirror are put into use.

See *Managing the DG/UX® System* for details on methods to boot the DG/UX system.

When upgrading my system, how can I preserve my tuned parameters?

For DG/UX System Release 4.10, the system configuration file, named **/var/Build/system.***identifier*, is formed by two separate files: **system.device.***identifier* and **system.params.***identifier*, where *identifier* is the name you assign to the system configuration file, which is typically a host's name. A backup of your previous system configuration file is saved in **Pre4.10.system.***identifier*.

You do not edit the **system.***identifier* file directly. Instead, you edit its components: **system.device.***identifier* and **system.params.***identifier*.

Figure 11–1 shows an example of how the system files are arranged for DG/UX System R4.10.

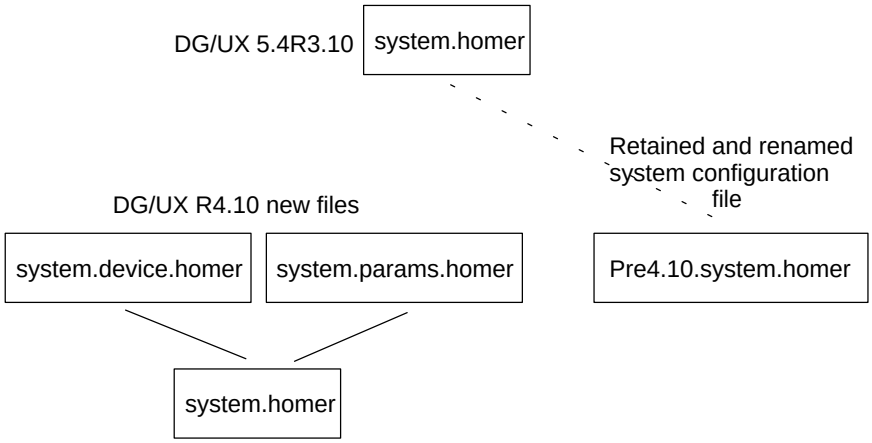


Figure 11–1 New system configuration file arrangement for DG/UX R4.10

This new arrangement allows you to confine changes to only devices or only the kernel tunable parameters and software package attributes.

To restore your customizations, follow these instructions:

1. Use the procedures in *Installing the DG/UX® System* to upgrade your DG/UX system. You will be instructed to build and boot a new kernel. The basis for the new kernel will reside in two separate configuration files: **system.device.***identifier* and **system.params.***identifier*. Your previous kernel configuration, including customizations, are preserved in **Pre4.10.system.***identifier*.

2. Immediately following the upgrade, using a text editor, enter your previous device customizations to **system.device.***identifier* and previous kernel tunable parameters and non-standard software package attributes to **system.params.***identifier* from **Pre4.10.system**.*identifier*.

   Entering your kernel customizations to **system.device.***identifier* and **system.params.***identifier* is a one-time event. Subsequent kernel building sessions will update the kernel using the customized files as a base.

3. After editing your configuration files, through stand-among **sysadm**, you must rebuild your kernel and reboot the system to implement your customizations.

Why don't my rc scripts run at boot time?

The **init** program runs **rc.init**, and **rc.init** runs the appropriate scripts. If a script does not run, it probably is unavailable or is not executable in the appropriate **/etc/rc?.d** directory.

Review the principles of rc scripts and run levels in *Managing the DG/UX® System*.

During installation, I supplied the wrong configuration information to a package setup prompt. Can I fix it before I build the kernel?

If you supply the wrong information during **tcpip** package setup, your kernel will build and boot successfully, but your network will not operate. We recommend that you build and boot the kernel to run level 1 before you correct the problem. Follow these steps to recover:

1. Boot the kernel to run level 1 using a boot command that is appropriate to your system.

2. Shut down the system.

   # **shutdown –g0 –y** ↵

3. Rerun package setup for the desired package, such as **tcpip**, using the following command:

   # **admpackage –o setup –F –rPRIMARY tcpip** ↵

   Answer the queries, this time supplying the correct data. Refer to *Installing the DG/UX® System* for details.

   IMPORTANT You may supply the name of any package in place of **tcpip** in the example.

4. Change the run level to init 3.

   # **init 3** ↵

5. Log in to your system.

End of Chapter

A Tuning the system and network logging facilities

The system error logging facility, **syslogd**, directs messages to **/var/adm/messages** that describe errors, abnormal conditions, routine checkpoints, and status from various system programs.

The logging facility directs messages according to these properties: the message originators, the severity levels of the message, and the destinations for the message. Severity and destinations are described in the **/etc/syslog.conf** file, which you can view through **sysadm**, choosing this path:

```
Logging -> System -> List
```

Through **sysadm**, you may choose which type of messages to log: such as those generated by user processes, the kernel, the mail system, and the print spooler. Also, you may produce a report of system errors that have been logged.

Starting and stopping the system error log (syslogd)

Syslogd is enabled by default.

To stop **syslogd** from being initialized at boot time, choose this **sysadm** path:

```
System -> Parameters -> Set
```

Answer **no** at the prompt:

```
Start System Error Log Daemon at Boot? [yes] no ↵
```

To disable **syslogd** for the current login session, use this command:

```
# dg_kill -15 syslogd ↵
```

Tuning syslogd

You can tune these **syslogd** properties: the message originators, the severity levels of the message, and the destinations for the message.

Choose **sysadm** `Logging -> System -> Add`, which presents you with these prompts:

`Facility Producing Error`

Specify the originator of the message being logged. Valid values are:

user	User processes. This is the default.
kern	The kernel.
mail	The mail system.

daemon	Daemon system servers such as **ftpd**.
auth	The authorization system: **login**, **su**, and **ttymon**.
lpr	The printer spooling system.
cron	The **cron** or **at** facility.
local0–7	Local use.
mark	Timestamp messages produced internally by **syslogd**.
news	The USENET network news system.
uucp	The UUCP system.
all	All facilities except the mark facility.

`Error Severity Level`

Enter the severity level. Valid severity levels, in descending order of severity, are:

emergency	Halt conditions that normally would be broadcast to all users.
alert	Conditions that should be corrected immediately, such as a corrupted system database.
critical	Messages about critical conditions, such as hard device errors.
error	Other errors.
warning	Warning messages.
notice	Conditions that are not caused by an error, but may require attention.
information	Informational messages.
debug	Messages that normally are used when debugging a program.
none	No messages.

`Logging Destination:`

Specify where you want to forward the message to:

save to file	Enter the pathname of the file to which you want the message logged.
write message to user(s)	Specify the recipients of the message: Enter one user name, a list of user names separated by commas, or an * for all logged-in users. The message is written to the screens of all specified recipients who are currently logged in.
send to remote host	Enter the name of the remote system to which you want the message sent.

Deleting log selections

Through Delete, you can delete a logging selection by the facility producing the error, severity level, and logging destination for a particular host. Choose `Logging -> System -> Delete`. The Delete operation presents you with this prompt:

```
Entry to Delete:
```

There is no default. Enter the selection by facility, level, and action to be deleted. Type **?** for a list, from which you can select entries for deletion.

Modifying log selections

You can change any of the settings previously made when you enabled logging. Choose `Logging -> System -> Modify` to alter error logging. The Modify operation presents you with this prompt:

```
Entry to Modify:
```

Enter the system logging entry to be changed in the database. Type **?** to get a current list, an example of which follows:

```
Choices are

     1  *.err           /dev/console
     2  kern.debug      /dev/console
     3  auth.notice     /dev/console
     4  *.err           /usr/adm/messages
     5  kern.debug      /usr/adm/messages
     6  daemon.notice   /usr/adm/messages
     7  auth.notice     /usr/adm/messages
     8  mail.crit       /usr/adm/messages
     9  kern.crit       /var/adm/dgsvcmgr/log.com
    10  user.info       /dev/console
    11  *.alert         root
    12  daemon.info     /var/adm/daemon.info
    13  *.emerg         *
    14  daemon.notice   /var/adm/log/admd.log
    15  daemon.notice   /var/adm/messages

Enter a number, a name, the initial part of a name,
<NL> to take the default,? for help, ^ to return to
the previous query, < to restart the operation,or q
to quit.
```

Supply the number of the entry to modify. Remaining prompts are identical to those presented for the Add operation.

Listing system log selections

You can display the list of facilities and severity levels that are being logged. Choose `Logging -> System -> List`. Sample output from this selection follows:

```
Facility    Level       Destination
------      -----       -----------------------
*           error       file /dev/syscon

kernel      debug       file /dev/syscon

auth        notice      file /dev/syscon

*           error       file /usr/adm/messages

kernel      debug       file /usr/adm/messages

daemon      notice      file /usr/adm/messages
auth        notice      file /usr/adm/messages

mail        critical    file /usr/adm/messages

kernel      critical    file /var/adm/dgsvcmgr/log.com

user        info        remote host jester
*           alert       write to root

daemon      info        file /var/adm/daemon.info

*           emergency   all logged-in users
```

An asterisk (*) selects all facilities.

Generating a log report

You can generate reports of system errors that have been logged to files on the target host. Choose `Logging -> System -> Report`. The Report operation presents you with these prompts:

`Originating Host(s):`

Enter the names of the hosts whose messages you want to see. Only local log files are consulted, but those files may include messages sent from other hosts by way of remote host entries. To specify a list of hosts, separate host names with a comma. The default is all hosts.

`Identifier(s):`

Enter the names of the programs that report the system error. To specify a list, separate identifier names with a comma. The default is all identifiers.

`Message(s):`

Allows you to provide a regular expression to match the type of error messages you are interested in. You may need to view all messages so that you can derive a meaningful pattern. For example, to see only NetWorker-related errors, you can enter this regular expression:

```
Net.*
```

to match the pattern:

```
NetWorker savegroup: info patriot_full (with one client)
```

Sample output follows:

```
Date            Host       Identifier   Message
--------------  ---------  -----------  ------------------
Feb 19 23:49:35  patriot    dg/ux       Tape device at
                                    st(cisc(vme(0),0,7),5,0)
Feb 19 23:49:35  patriot    dg/ux       encountered a hard
                                        error at block
                                        0,status = 4005007
Feb 19 23:48:15  patriot    syslog      NetWorker media:
                                        (notice) 8mm tape
                          patriot_week3_a used 274 MB of 2000
Feb 19 23:48:10  patriot    dg/ux       Tape device at
                                    st(cisc(vme(0),0,7),5,0)
Feb 19 23:48:10  patriot    dg/ux       encountered a hard
                                        error at block
                                        0, status = 4005007
Feb 19 09:52:37  patriot    syslog      NetWorker index:
                                        (notice) cross-
                                        check has completed.
Feb 19 09:52:23  patriot    syslog      NetWorker index:
                                        (notice) nsrck has
                                        completed.
Feb 19 09:52:12  patriot    syslog      NetWorker Server:
                                        (notice) started
Feb 19 09:51:03  patriot    dg/ux       Firmware in SCSI
                                        controller
                         cisc(vme(0),0) is out of date
Feb 19 09:51:03  patriot    dg/ux       see release notice.
Feb 19 09:51:03  patriot    dgsvcd      AV/Alert System:
                                        Disabled
Feb 19 09:47:06  patriot    syslogd     going down on
                                        signal 15
Feb 19 09:47:05  patriot    dgsvcd      AV/Alert System:
                                        Going down on
                                        signal 15
```

Using the network error log

You establish the conditions under which network error messages are listed and deleted. The errors reported are dependent on the software that is installed. LAN device drivers, TCP/IP, X.25, or OSI can report such errors.

Viewing the network error log

Choose `Logging -> Network -> List` to view the network error logs. The List operation presents you with these prompts:

`Age (in days):`

Enter the age (in days) of the oldest logged message to view. The default is three days. The default would present messages up to and including three days old, including one and two days old.

`Facility Producing Error:`

Enter the origin of the message being logged. The list produced depends on the software that is installed. By default, messages from all sources are listed. Type **?** for a list of sources.

`Error Severity Level:`

Enter the severity level. Recognized values, in descending order of severity, are:

emergency	Halt conditions that normally would be broadcast to all users.
alert	Conditions that should be corrected immediately such as a corrupted system database.
critical	Messages about critical conditions, such as hard device errors.
error	Other errors.
warning	Warning messages.
notice	Conditions that are not error conditions, but may require attention.
information	Informational messages.
debug	Messages that normally are used when debugging a program.

Deleting log messages

You can delete log messages by age. Choose `Logging –> Network –> Delete`. The Delete operation presents you with a single prompt:

`Age (in days):`

Enter the age (in days) of the logged messages you want to keep. The default is three days. All messages older than the specified age will be deleted.

End of Appendix

B Fsck messages

When **fsck** detects an inconsistency, it reports the error condition on the console screen. If a response is required, **fsck** prints a prompt message and waits for a response. This section explains the meaning of each error condition, possible responses, and related error conditions.

In “Error messages for phased checking,” the error conditions are organized by the phase of the **fsck** program in which they can occur. The error conditions that may occur in more than one phase are discussed under “General error messages.”

Error messages that occur only during fast recovery file system checking (**fsck** logging is enabled) appear in “Error Messages Exclusive to Fast Recovery Checking.”

The following error messages are presented in their basic form. Fatal errors cause the error message to be prefaced by the string `Fatal Error`. Running with **–p** also causes messages to be preceded by the name of the file system to which the message applies. The following abbreviations appear in the description of error messages:

B	A (decimal) disk block number.
N	A decimal number.
O	An octal number.
C	A character.
D, *F*	A directory name, file name or pathname string.
I	A file node description string. At the very least, this will consist of the file node number. If possible, the file node's size, file type, file mode, UID, GID, time of last modification, owner name, group name and pathname will also be present.

General error messages

The messages described in this section may appear at any time during an **fsck** session.

`Cannot allocate memory for internal tables (N bytes requested)`

The **fsck** program cannot allocate enough memory; this can occur only when invoked through stand-alone **sysadm**. The **fsck** program will abort. Bring up your system and use the **fsck** command instead.

`Cannot read block B`

A disk read of block number *B* has failed. The **fsck** program treats the block it could not read as if it were filled with all zeroes, and continues execution, but the file system is not marked as mountable upon conclusion of checking. Use **sysadm** (`Device -> Disk -> Physical -> Bad Blocks -> Map`) to remap the bad block *B* and run **fsck** again.

`Cannot write block B`

A disk write of block number *B* has failed. The **fsck** program continues execution, but the file system being checked is not marked as mountable upon conclusion of checking. Use **sysadm** (`Device -> Disk -> Physical -> Bad Blocks -> Map`) to remap the bad block *B* and run **fsck** again.

`Fork failed`

The **fsck** program has failed in an attempt to spawn a child process. This will occur only when running **fsck** with the **–p** option. The only file system affected will be the one for which the child **fsck** process was being created; no check will occur.

`Internal Software Error: Cannot seek to block B -- aborting`

A disk seek to block number *B* has failed; this should never happen. Contact your Data General support representative if this message is displayed. The **fsck** program terminates.

`Invalid response; please answer yes or no`

An invalid answer has been entered in response to one of **fsck**'s questions. The **fsck** program will not continue until a valid response has been entered. The following strings are valid responses: **y, Y, yes, YES, n, N, no** and **NO**.

Errors during fsck invocation

Before starting to check a file system, **fsck** must parse its command line and determine which files to check. The following messages result from command line errors or information in the file **/etc/fstab**.

`The directory D is the mount point for F`

The **fsck** program has been given a directory *D* to check and has determined that *D* is the mount point for the file system *F*. This message is purely advisory.

`The flags –y, –n, –p, –q and –S are all mutually exclusive`

More than one of the above flags has been specified on the **fsck** command line. Only one is allowed. The **fsck** program will abort.

`Unknown option: –C`

An unknown option flag, *C*, has been specified on the **fsck** command line. Valid flags are as follows: **–l**, **–y**, **–n**, **–p**, **–q**, **–t**, **–D**, **–f**, **–s**, **–S**, and **–x**. When you give it an invalid option, **fsck** will abort.

Errors during fsck initialization

Before a file system check can be performed, **fsck** must set up certain tables and open certain files. The following messages can result from errors during this phase.

`Block B is invalid file node Table Block –– rewrite as empty block?`

The file node table block *B* does not contain the proper self-identification information. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to rewrite the block.

Possible responses to the `rewrite as empty block?` prompt are:

YES	Fix this error condition by rewriting this block as an empty file node table block. Any file nodes that formerly occupied slots in this block will be cleared and any associated data will be lost.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Cannot determine disk size of F`

The **fsck** program has been given a file system *F* to check, but the size of *F* cannot be determined. The **fsck** program will abort checking this file system. This should never happen. Contact your Data General support representative if this message is displayed.

`Cannot find a readable copy of the superblock`

Neither of the two copies of the superblock can be read. The **fsck** program will abort checking this file system.

`Cannot find a valid copy of the superblock`

Neither of the two copies of the superblock contain the required self-identification information. The **fsck** program will abort checking this file system.

`Cannot open F for reading`

The **fsck** program has been given a file system *F* to check, but *F* cannot be opened for reading. Check the mode of *F*. The **fsck** program will abort checking this file system.

`Cannot open F for writing`

The **fsck** program has been given a file system *F* to check, but *F* cannot be opened for writing. Check the mode of *F* and make sure that no disks containing the file system are physically write-disabled. The **fsck** program will abort checking this file system.

`Cannot read superblock copy N`

One of the two superblock copies cannot be read. The **fsck** program will attempt to use the other copy and continue.

`F is not a regular file, block–special file, character–special file or valid mount point`

The **fsck** program has been given a file system *F* to check, but *F* is not of the correct type. *F* must be an ordinary file type, block-special or character-special, or else it must be listed in the file **/etc/fstab** as a valid mount point directory. The **fsck** program will abort checking this file system.

`File system is too large to check`

Stand-alone **fsck** cannot allocate enough memory for its internal tables to begin checking the file system. The **fsck** program will abort checking this file system. Bring up your system and use the **fsck** command instead.

`File system size stored in superblock is incorrect (N1 blocks should be N2) -- fix?`

The superblocks contain an incorrect file system size figure. If run with the **–p** or **–q** options, **fsck** will automatically correct this. Otherwise, **fsck** will ask to correct the size.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the file system size to N2, the actual size of the disk containing the file system.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Invalid default Data Element Size exponent: N -- fix?`

The default data element size for files (stored in the superblocks as a base-2 logarithm) is invalid. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to set the value to the default of 4 (meaning a file data element size of 16 blocks).

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the default data element value to 4.
NO	Ignore this error condition. The **fsck** program will abort checking this file system.

`Invalid default Directory Data Element Size exponent: N -- fix?`

The default data element size for directories (stored in the superblocks as a base-2 logarithm) is invalid. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to set the value to the default of 4 (meaning directory data element size of 16 blocks).

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the default directory value to 4.
NO	Ignore this error condition. The **fsck** program will abort checking this file system.

`Invalid default Directory Index Element Size exponent: N -- fix?`

The default index element size for directories (stored in the superblocks as a base-2 logarithm) is invalid. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to set the value to the default of 0 (meaning directory index element size of 1 block).

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the default directory index value to 0.
NO	Ignore this error condition. The **fsck** program will abort checking this file system.

`Invalid default Index Element Size exponent: N -- fix?`

The default index element size for files (stored in the superblocks as a base-2 logarithm) is invalid. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to set the value to the default of 0 (meaning a file index element size of 1 block).

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the default index element value to 0.
NO	Ignore this error condition. The **fsck** program will abort checking this file system.

`Invalid Disk Allocation Region size: N blocks`

The DAR size stored in the superblocks is invalid. The **fsck** program will abort checking this file system.

`Invalid first allocation threshold file size: N -- fix?`

The superblocks contain an invalid first allocation threshold file size (the number of blocks a file can allocate in its initial DAR before all subsequent allocations are made from a different DAR). If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to correct the size.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the first allocation threshold file size to the default limit for DARs of the size specified in the superblock.
NO	Ignore this error condition. The **fsck** program will abort checking this file system.

`Invalid number of file nodes per Disk Allocation Region`

The number of file nodes per DAR stored in the superblocks is invalid. The **fsck** program will abort checking this file system.

`Invalid second allocation threshold file size: N -- fix?`

The superblocks contain an invalid second allocation threshold file size (the number of blocks a file can allocate in a noninitial DAR before all subsequent allocations are made from a different DAR). If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to correct the size.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the second allocation threshold file size to the default limit for DARs of the size specified in the superblock.
NO	Ignore this error condition. The **fsck** program will abort checking this file system.

`No check necessary for F`

The file system *F* is already marked mountable and **fsck** was invoked with the **–x** flag. The **fsck** program will not check this file system.

`Superblock copies differ; using newer copy`

Both copies of the superblock are readable and both contain the required self-identification information, but they differ. The **fsck** program will use the first copy (which is guaranteed to be more recent) and continue.

`Superblock copy N is invalid`

One of the two superblock copies does not contain the required self-identification information. The **fsck** program will attempt to use the other copy and continue.

`Superblock has invalid contents in reserved area -- fix?`

A copy of the superblock has nonzero contents in a reserved area. If running with the **–p** flag, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to fix the reserved area.

Possible responses to the `fix?` prompt are:

YES	The superblock's reserved area is initialized so that it contains all 0s.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

Error messages for phased checking

The **fsck** program checks file systems in phases only if **fsck** logging was not enabled. This section lists errors you may see during the typical, phased **fsck** check.

A later section "Error messages exclusive to fast recovery checking" lists errors that can appear during file system checking for which **fsck** logging was enabled.

Errors during phase 1 – check blocks and file sizes

This phase of **fsck** operation is concerned with file nodes. The following messages result from errors in file node types, file node format, file sizes and the data element pointers and index element pointers that make up a file's structure.

`Incorrect block count in file node I (N1 should be N2) -- fix?`

The file node *I*'s count of the blocks it uses is incorrect. If run with the **–p** option, **fsck** will automatically correct the count to *N2*. Otherwise, **fsck** will ask to correct the count.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting file node *I*'s block count to *N2*.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`file node I claims an invalid block (B) -- clear bad pointer?`

The file node *I* claims block *B*, which does not exist. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to clear the element pointer claiming the invalid block.

Possible responses to the `clear bad pointer?` prompt are:

YES	Fix this error condition by clearing the pointer in file node *I* that claims the nonexistent block. This may result in a "hole" in the file if the cleared pointer preceded the last block of the file.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`file node I claims a system block (B) -- clear bad pointer?`

The file node *I* claims block *B*, which is a system block (a bitmap block, file node table block, DAR entry table block or superblock). If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to clear the element pointer claiming the system block.

Possible responses to the `clear bad pointer?` prompt are:

YES — Fix this error condition by clearing the pointer in file node *I* that claims the system block. This may result in a "hole" in the file if the cleared pointer preceded the last block of the file.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`file node I has an Index Block (B) with invalid format -- clear bad pointer?`

The file node *I* claims block *B* as an index block, but block *B* does not contain the proper self-identification information. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to clear the element pointer claiming the invalid block.

Possible responses to the `clear bad pointer?` prompt are:

YES — Fix this error condition by clearing the pointer in file node *I* that claims the index block. This may result in a "hole" in the file if the cleared pointer preceded the last block of the file.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`file node I has invalid contents in its reserved area -- fix?`

The file node *I* does not contain the proper self-identification information. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to fix the reserved area.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by initializing file node *I*'s reserved area.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`file node I has invalid fragment size exponent (N) -- clear?`

The file node *I* has an illegal exponent representing the size of the file's fragment. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to clear the file.

Possible responses to the `clear?` prompt are:

YES	Fix this error condition by clearing file node *I*.
NO	Ignore this error condition. The **fsck** program will abort checking this file system.

`file node I is of unknown file type (O) -- clear?`

The file node *I* is of type *O*, which is an unrecognized octal number. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to clear the file.

Possible responses to the `clear?` prompt are:

YES	Fix this error condition by clearing file node *I*.
NO	Ignore this error condition. The **fsck** program will abort checking this file system.

`file node I is partially truncated -- fix?`

The file node *I*'s size is shorter than the number of blocks allocated to it. If run with the **–p** option, **fsck** will complete automatically the truncation. Otherwise, **fsck** will ask to complete truncating file node *I*.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by completing the truncation of file node *I* down to the size stored in the file node.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

Errors during phase 1b – resolve duplicate claims

When **fsck** finds a block claimed by two or more files, it rescans the file system to find the original claimant of that block. This section lists the error messages that result from settling the claim to the disputed block.

`file node I claims another file's blocks -- clear?`

The file node *I* claims some blocks that belong to another file. **fsck** will ask to clear the file.

Possible responses to the `clear?` prompt are:

YES — Fix this error condition by clearing file node *I*.

NO — Ignore this error condition. This will result in the same question being asked about the next claimant of the disputed block. As long as enough files are eventually cleared to resolve the duplicate claims on the block, **fsck** will continue normally. However, if at the end of Phase 1b any duplicate claims still exist, **fsck** will not mark this file system as mountable upon completing the check.

Errors during phase 2 – check directory contents

This phase is concerned with the contents of directories. The messages in this section result from improperly formatted directory blocks, an improperly formatted root (*/*) directory, and bad directory entries. During this phase, all bad entries and file nodes are removed from the file system tree.

`Directory file node I has a hole -- fix?`

The directory file node *I* has at least one "hole" in its file structure (gaps before the end of file). If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to rearrange the directory blocks to fill in the hole.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by rearranging the blocks in the directory to eliminate the hole.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Directory file node I has incorrect child count (N1 should be N2) -- fix?`

The directory file node *I*'s count of children, *N1*, is incorrect. If run with the **–p** or **–q** options, **fsck** will automatically correct the count to *N2*. Otherwise, **fsck** will ask to correct the child count.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting file node *I*'s child count to *N2*.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Directory file node I has an invalid block (B) -- rewrite as empty block?`

The directory file node *I* has a block (address *B*) which does not contain the proper self-identification information. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to rewrite the block.

Possible responses to the `rewrite as empty block?` prompt are:

YES	Fix this error condition by rewriting block *B* as an empty directory block. Any directory entries that formerly occupied this block will be destroyed.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Directory file node I1 has entry for file node I2 of invalid size -- remove bad directory entry?`

The directory file node *I1* has a directory entry for file node *I2*, but the entry is too long, too short, or is not a multiple of 4 bytes in size. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*. If file node *I2* is an allocated file node with no remaining links, there will be an opportunity to reattach it in the **/lost+found** directory during Phase 3.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Directory file node I1 has entry for file node I2 which is out of order -- remove bad directory entry?`

The directory file node *I1* has a directory entry for file node *I2* which has a bad sequence number, meaning that the entry is invalid. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*. If file node *I2* is an allocated file node with no remaining links, there will be an opportunity to reattach it in the **/lost+found** directory during Phase 3.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Directory file node I1 has entry for file node I2 with filename of invalid size -- remove bad directory entry?`

The directory file node *I1* has a directory entry for file node *I2*, but the entry's file name is too long or too short. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*. If file node *I2* is an allocated file node with no remaining links, there will be an opportunity to reattach it in the **lost+found** directory during Phase 3.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Directory file node I1 has entry for file node I2 with an illegal
filename: F -- remove bad directory entry?
```

The directory file node *I1* has a directory entry for file node *I2*, but the entry's name *F* is **.** (dot) or **..** (dot-dot). These two names are reserved for the directory's links to itself and to its parent, respectively. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*. If file node *I2* is an allocated file node with no remaining links, there will be an opportunity to reattach it in the **lost+found** directory during Phase 3.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Directory file node I1 has entry for file node I2, which has a filename
with an illegal character, octal value O -- remove bad directory entry?
```

The directory file node *I1* has a directory entry for file node *I2*, but the entry's name includes the illegal character *O*. Neither a non-ASCII nor the slash character is allowed. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*. If file node *I2* is an allocated file node with no remaining links, there will be an opportunity to reattach it in the **lost+found** directory during Phase 3.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Directory file node I1 has entry for file node I2, which has an
illegally long pathname -- remove bad directory entry?
```

The directory file node *I1* has a directory entry for file node *I2*, but the pathname for that entry relative to the root of the file system would exceed **MAXPATHLEN** (1024) bytes. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*. If file node *I2* is an allocated file node with no remaining links, there will be an opportunity to reattach it in the **/lost+found** directory during Phase 3.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Directory file node I1 has entry for file node I2, which has invalid
contents in its reserved area -- fix?
```

The directory file node *I1* has a directory entry for file node *I2*, which has nonzero information in its reserved area. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to fix the contents of the reserved area of file node *I2*.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by initializing the reserved area of file node *I2*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Directory file node I1 has entry for file node number I2, which is
invalid -- remove bad directory entry?
```

The directory file node *I1* has a directory entry for file node number *I2*, but *I2* is not a valid file node number. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Directory file node I1 has entry for file node number I2, which is
unallocated -- remove bad directory entry?
```

The directory file node *I1* has a directory entry for file node number *I2*, but *I2* is not an allocated file node. If run with the **–p** option, **fsck** automatically will remove the directory entry for file node *I2*. Otherwise, **fsck** will ask to remove the directory entry.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Directory file node I1 has entry which is an extraneous link to
directory file node I2 -- remove bad directory entry?
```

The directory file node *I1* has a directory entry for file node number *I2*, but *I2* is a directory that does not list *I1* as its parent directory. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*. If file node *I2* is an allocated

file node with no remaining links, there will be an opportunity to reattach it in the **/lost+found** directory during Phase 3.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Directory file node I1 has entry which is an extraneous link to symbolic
link file node I2 -- remove bad directory entry?
```

The directory file node *I1* has a directory entry for file node number *I2*, but *I2* is a symbolic link which already has another hard link. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Directory file node I1 has an entry (for file node I2) which crosses a
control point directory boundary -- remove bad directory entry?
```

The directory file node *I1* has a directory entry for file node number *I2*, but *I1* and *I2* have different space parent control point directories. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the directory entry for file node *I2*.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the directory entry for file node *I2*. If file node *I2* is an allocated file node with no remaining links, there will be an opportunity to reattach it in the **/lost+found** directory during Phase 3.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Incorrect filename length in directory file node I1 for directory file
node I2 (N1 should be N2) -- fix?
```

The directory file node *I1* has a directory entry for file node *I2*, but the entry's name length, *N1*, is incorrect. If run with the **–p** option, **fsck** will automatically correct the directory entry's name length to *N2*. Otherwise, **fsck** will ask to correct the name length.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the directory entry's length to *N2* bytes.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Root file node is of wrong file type -- fix?`

The root file node (file node 2) is not a control point directory. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to fix the incorrect file type.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the file type of file node 2 to type control point directory.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Root file node is not allocated -- fix?`

The root file node (file node 2) is not allocated. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to allocate file node 2.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by allocating file node 2 as the root.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Root file node's parent directory is not the root -- fix?`

The root file node's parent directory is not the root (itself). If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to list the root file node as its own parent.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the root file node's parent directory to itself.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Root file node's space parent control point directory is not the root -- fix?`

The root file node's space parent control point directory is not the root (itself). If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to list the root file node as its own space parent.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the root file node's space parent control point directory to itself.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Root file node's space usage limit is too large (N1 should be N2) -- fix?`

The root file node's space usage limit, *N1*, is bigger than the size of the file system, *N2*. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to reset the limit to *N2* blocks.

Possible responses to the `fix?` prompt are:

YES	Fix this error condition by setting the root file node's space usage limit to *N2* blocks.
NO	Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

Errors during phase 3 – check connectivity

Phase 3 of **fsck** deals with the reconnection of unreferenced files and directories onto the file system tree. The messages in this section result from attempts to connect unreferenced files into the **lost+found** directory. Note also that any of the Phase 2 messages may be seen in this phase, as the contents of any reconnected directories must be checked.

`Cannot find enough contiguous free blocks to expand directory file node I`

The **fsck** program could not find enough contiguous free blocks to expand the directory file node *I*. Some unreferenced files may not be reconnected as a result of this failure; they can be reconnected during a later **fsck** session after enough space has been freed in the file system.

`Control point directory file node I has an entry named 'lost+found' which is not a directory -- remove bad directory entry?`

The control point directory file node *I* already has an entry named **lost+found**, but which is not of type directory. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the entry.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the bad entry from file node *I*. The bad entry's file node will itself be reattached in the new **/lost+found** directory which will be created in directory *I1*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Could not reconnect file node I`

The **fsck** program was unable to reconnect the unreferenced file node *I* because it could not allocate enough blocks to expand the **lost+found** directory, or because it could not allocate a free file node to use as the **lost+found** directory.

`Directory file node I is already as large as it can become`

The **fsck** program has discovered that a directory it was attempting to expand is already the maximum size a directory can become.

`Directory file node I needs to be expanded -- fix?`

The directory file node *I* needs to be expanded so that another directory entry can be added to it; *I* is either the root directory or the **lost+found** directory. If run with the **–p** or **–q** options, **fsck** will automatically attempt to expand the directory. Otherwise, **fsck** will ask to expand it.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by attempting to expand file node *I*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`file node I1 lists as its space parent file node number I2, which is not a valid control point directory -- reset space parent to root?`

The file node *I1* has the non-control point directory file node *I2* listed as its space parent. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to reset *I1*'s space parent to file node *2*, the root of the file system.

Possible responses to the `reset?` prompt are:

YES — Fix this error condition by resetting *I1*'s space parent to file node 2, the root of the file system.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`file node I is unreferenced -- clear?`

The file node *I* has no links in the file system and an earlier reconnection failed or was refused.

Possible responses to the `clear?` prompt are:

YES — Fix this error condition by clearing file node *I*. The contents of the file will be destroyed.

NO — Ignore this error condition. file node *I* will remain unattached and can be reattached during a later **fsck** session provided that enough blocks and/or file nodes are free.

`file node I is unreferenced -- reconnect?`

The file node *I* has no links in the file system. If run with the **–p** option, **fsck** will automatically attempt to reconnect the file. Otherwise, **fsck** will ask to reconnect it.

Possible responses to the `reconnect?` prompt are:

YES — Fix this error condition by reconnecting file node *I* in the **lost+found** directory, with the name "#*N*", where *N* is the file node number of *I*.

NO — Ignore this error condition.

```
The lost+found directory file node I already has an entry named 'F'  --
remove bad directory entry?
```

The **lost+found** directory file node *I* has discovered that it already has an entry of the name *F* when it was trying to reconnect an unreferenced file which would have had the same name. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to remove the spurious entry.

Possible responses to the `remove bad directory entry?` prompt are:

YES — Fix this error condition by removing the entry for *F*; the file node referenced by that entry will be reattached with a name constructed from its file node number.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

Errors during phase 4 – check link counts and resource accounting

This phase checks the link counts of individual file nodes and the resource counts (blocks and file nodes used) of control point directories. The messages result from errors in these counts.

```
Control point directory file node I has incorrect file node allocation
count (N1 should be N2) -- fix?
```

The control point directory file node *I* has a bad count of the file nodes used by it and all its space descendants. If run with the **–p** or **–q** options, **fsck** will adjust automatically the count to *N2*. Otherwise, **fsck** will ask to fix the count.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by adjusting the file node count for file node *I* to *N2*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

```
Control point directory file node I has incorrect space allocation count
(N1 should be N2) -- fix?
```

The control point directory file node I has a bad count of the blocks used by it and all its space descendants. If run with the **–p** or **–q** options, **fsck** will adjust automatically the count to *N2*. Otherwise, **fsck** will ask to fix the count.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by adjusting the space count for file node *I* to *N2*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`file node I has incorrect link count (N1 should be N2) -- fix?`

The file node *I* has a bad link count. If run with the **–p** or **–q** options, **fsck** will adjust automatically the count to *N2*. Otherwise, **fsck** will ask to fix the count.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by adjusting the link count for file node *I* to *N2*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

Errors during phase 5 – check disk allocation region information

This phase deals with the disk allocation regions. Messages in this section result from errors in the components of the DARs: the bitmap, the free file node list, and various resource counts.

`Block B of the Disk Allocation Region Information Area is invalid -- fix?`

The disk allocation region information area block *B* does not contain the proper self-identification information. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to rewrite the block.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by rewriting this block as an empty disk allocation region information area block.

The DAR information in the block will be corrected later in this Phase.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Disk Allocation Region N has incorrect Bitmap -- fix?`

The bitmap for DAR *N* is incorrect. If run with the **–p** option, **fsck** will correct automatically the bitmap. Otherwise, **fsck** will ask to correct it.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by rewriting the bitmap correctly.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Disk Allocation Region N has incorrect count of blocks used (N1 should be N2) -- fix?`

The block count for DAR *N* is incorrect. If run with the **–p** or **–q** options, **fsck** will correct automatically the count to *N2*. Otherwise, **fsck** will ask to correct it.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by changing DAR *N*'s block count from *N1* to *N2*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Disk Allocation Region N has incorrect counts of directories and file nodes used -- fix?`

The counts of used files and directories for DAR *N* are incorrect. If run with the **–p** or **–q** options, **fsck** will correct automatically the counts. Otherwise, **fsck** will ask to correct them.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by rewriting the corrected counts of used file nodes and directories.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Disk Allocation Region N has incorrect free file node list -- fix?`

The linked list of free file nodes in DAR *N* is incorrect: it contains allocated file nodes, duplicates, or it does not contain some file nodes that are actually unallocated. If run with the **–p** or **–q** options, **fsck** will correct automatically the free list. Otherwise, **fsck** will ask to correct it.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by rewriting the free list for DAR *N*.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Disk Allocation Region N has invalid contents in its reserved area -- fix?`

Disk allocation region number *N* has nonzero contents in its reserved area. If run with the **–p** option, **fsck** will abort checking this file system. Otherwise, **fsck** will ask to zero out the reserved area.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by initializing the contents of the reserved area.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

`Incorrect summary counts in superblocks -- fix?`

The counts of used blocks and files in the two copies of the superblock are incorrect. If run with the **–p** or **–q** options, **fsck** will correct automatically the counts. Otherwise, **fsck** will ask to correct them.

Possible responses to the `fix?` prompt are:

YES — Fix this error condition by correctly rewriting the counts of used blocks and files.

NO — Ignore this error condition. The **fsck** program will not mark this file system as mountable upon completing the check.

Advisory messages during file system cleanup

After a file system has been checked, a few cleanup functions are performed. This section lists advisory messages about the file system.

`File System is now mountable`

The **fsck** program has successfully checked the file system and marked it as mountable.

`File System is still inconsistent and not mountable`

The **fsck** program has checked the file system, but inconsistencies remain and the file system is still marked as unmountable. Re-run **fsck** to fix the remaining inconsistencies.

`N1 of N2 blocks used (N3 free); N4 of N5 file nodes used (N6 free)`

The indicated number of blocks and file nodes (*N1*, *N2*, and *N3*) have been used, leaving the indicated number unallocated (*N4*, *N5*, and *N6).*

`Unconnected files still remain. Mount the file system and remove files to free data blocks and file nodes`

The **fsck** program has successfully checked the file system and marked it as mountable. However, there are still unreferenced files in the file system. You can recover from these unreferenced files by running **fsck** again after enough blocks and file nodes have been freed to allow them room to be reconnected.

Error messages exclusive to fast recovery checking

This section lists errors that can appear only during checking of a fast recovery file system, with **fsck** logging enabled. These messages do not appear with phased **fsck** messages in the previous section because fast recovery checking does not occur in phases.

Normal phased checking will return only the messages in the previous section. Fast recovery file system checking may return the messages in this section, as well as many (but not all) of those in the previous section.

`Bitmap block N in Disk Allocation Region N2 has an invalid format`

Fsck discovered a bitmap block with a bad ID. The arguments are the *lda* of the bitmap block *N1* and the number of the DAR it belongs to *N2*.

`Block N is invalid file node Table block`

A file node Table block has an invalid ID. The numeric argument is the invalid block.

`Block N of the Disk Allocation Region Information Area is invalid`

Fsck discovered a DAR table (Disk Allocation Region information area) which had an invalid ID. The argument is the ordinal number (within the DAR table) of the faulty block.

`Cannot allocate memory for internal tables (N bytes requested)`

An attempt to allocate memory failed. The numeric argument is the number of bytes requested.

`Cannot find enough contiguous free blocks to allocate a missing index element for file node I`

Fsck failed to allocate an index element. The argument describes the file node in question.

`Cannot open fast recovery dump file F`

The open of the file to hold the human-readable version of the fast recovery log failed. The argument is the pathname of the dump file.

`Cannot read DAR table`

The DAR table cannot be read.

`Cannot read fast recovery log at lda B`

The read of one of the fast recovery disk logs failed. The argument is the logical disk address of the log.

`Cannot recover from log. Run full fsck`

A fatal error occurred while trying to recover with the log. Normal full **fsck** will not be required to complete the recovery of this file system.

`Cannot write DAR table`

The DAR table cannot be written.

`Directory entry D in Directory file node I contains the wrong file node number (`*`B1`*` should be `*`B2`*`)`

A directory entry was discovered to contain an incorrect file node number. The arguments are the directory entry file name *D*, the file node number of the directory containing the entry *I*, the entry's incorrect file node number *B1*, and the correct file node number *B2*.

`Directory file node I has a spurious entry for file name F with file node number B`

An unneeded directory entry was found. The arguments are the directory's file node number *I*, the name of the missing entry *F*and the file node number that should be in that entry *B*.

`Directory file node I has an incorrect parent file node number (B1 should be B2)`

A directory file node containing the wrong parent file node number was found. The arguments are the directory's file node number *I* and the incorrect and correct parent file node numbers *B2*.

`Directory file node I has incorrect child count (N1 should be N2)`

Fsck discovered an erroneous child count in a directory. The string argument describes the file node in question. The first numeric argument is the old, incorrect count *N1*; the second numeric argument is the correct count *N2*.

`Directory file node I is missing an entry for file name F with file node number B`

A directory entry was found to be missing. The arguments are the directory's file node number *I*, the name of the missing entry *F* and the file node number that should be in that entry *B*.

`Disk Allocation Region N has an incorrect Free file node List`

Fsck discovered a DAR with incorrect Free File Node List information. The first argument is the DAR number *N*.

`Disk Allocation Region N has incorrect Bitmap`

Fsck discovered an incorrect bitmap. The argument is the number of the DAR with the bad bitmap *N*.

`Disk Allocation Region N1 has incorrect count of blocks used (N2 should be N3)`

Fsck discovered a DAR with incorrect space count. The first argument is the DAR number *N1*. The second argument is the incorrect, old count *N2*. The third argument is the correct count *N3*.

`Disk Allocation Region N has incorrect counts of directories and file nodes used`

Fsck discovered a DAR with incorrect directory or file node counts. The first argument is the DAR number.

`File node and space accounting partially recovered. Run full fsck to recover accounting information`

Fsck completed but accounting recovery from the log was not complete. This error does not prevent the file system from being mounted, but at some point, full **fsck** should be run to fully correct accounting information.

`file node I has a bad fragment size for data element #0 (B1 should be B2)`

A file node's *fragment_size_exponent* field was found to contain an incorrect value. The arguments are the file node number *I*, the bad fragment size exponent *B1* and the correct fragment size exponent *B2*.

`file node I has an Index Block (N) with invalid format`

Fsck discovered an element pointing to an index block which does not self-ID. The string argument describes the file node in question. The numeric argument is the bad block's number.

`file node I has an incorrect file size (N1 bytes should be N2 bytes)`

Fsck discovered a node having an incorrect file size. The first argument describes the file node in question *I*. The first numeric argument is the old, incorrect size *N1*. The second numeric argument is the correct size *N2*.

`file node I has an incorrect space parent file node number (B1 should be B2)`

A file node containing the wrong space parent file node number was found. The arguments are the file's file node number *I* and the incorrect *B1* and correct *B2* space parent file node numbers.

`file node I has bad element pointers`

An element pointer in the file node was found to contain the incorrect allocated element address. The argument is the file node number *I*.

`file node I has incorrect file node allocation count (N1 should be N2)`

Fsck discovered a node having an incorrect node count. The first argument describes the file node in question *I*. The first numeric argument is the old, incorrect count *N1*. The second numeric argument is the correct count *N2*.

`file node I has incorrect link count (N1 should be N2)`

Fsck discovered a node having an incorrect link count. The first argument describes the file node in question *I*. The first numeric argument is the old, incorrect count N*1*. The second numeric argument is the correct count *N2*.

`file node I has incorrect space allocation count (N1 should be N2)`

Fsck discovered a node having an incorrect space count. The first argument describes the file node in question *I*. The first numeric argument is the old, incorrect count *N1*. The second numeric argument is the correct count *N2*.

`file node I is incorrectly marked as allocated`

A file node allocation has to be backed out or the results of a deallocation did not make it to disk. The argument is the file node number *I*.

`file node I is partially truncated`

Fsck discovered a partially truncated file. The argument is the file node in question *I*.

`Internal software bug O`

A sanity check failed indicating a software bug. The argument is an octal status code.

`Missing log entry for file system fnode accounting`

There are file node allocations or frees on the file system, but that no log entry giving the initial value of the file system's used file node count was made. This error indicates that there is a bug in the kernel logging code.

`System buffers containing data from the following files may not have been written to disk:`

Fsck failed to allocate an index element. The argument describes the file node in question.

`Unexpected child count N in file node I`

Indicates a bug in the log recovery code.

`Unknown kind of log entry`

An unrecognized type of log entry was found in the log.

End of Appendix

C Using stand-alone sysadm

When your system is down and cannot be brought up in a normal way, use stand-alone sysadm (located at /usr/stand/sysadm) to repair the problem. Stand–alone sysadm allows you to manage physical and virtual disks, file systems, and system software while the DG/UX system is not running. You boot this version to perform operations that you cannot perform while running the installed version of the DG/UX software.

If, for example, the / (root) or /**usr** file systems become damaged and you cannot boot the system, use stand-alone **sysadm** to repair them and then boot the system again. Refer to Chapter 10 for information on repairing corrupt DG/UX file systems.

Booting stand-alone sysadm

If your Intel system has an SCM interface, boot stand-alone **sysadm** from the SCM prompt.

```
SCM> b sd(apsc(pci(),B)) usr –f /stand/sysadm ↵
```

On Intel systems without an SCM interface, boot stand–alone **sysadm** from the Boot command prompt.

```
Boot command: sd(npsc(pci(0),B)) usr –f /stand/sysadm ↵
```

You can boot stand-alone **sysadm** from disk only if the virtual disks containing the **/usr** file system are located on a single physical disk. If **/usr** spans multiple physical disks, you cannot boot from **/usr**. You would have to boot standalone **sysadm** from the release media.

Figure C–1 shows the stand-alone **sysadm** main menu.

```
                    Standalone Sysadm Main Menu

  1  Physical Disk –>          Manage physical disks
  2  Virtual Disk –>           Manage virtual disks
  2  File System –>            Manage file systems
  4  Install Software –>       Install system software


Enter a number, a name, ? or <number>? for help, <NL> to
redisplay menu,or q to quit: [Install Software]:
```

Figure C–1 Stand-alone sysadm main menu

Shell commands supported by stand-alone sysadm

Figure C–2 lists the shell commands in **/sbin** that stand-alone **sysadm** supports:

fsck	init	reboot	umount
halt	mount	sh	su

Figure C–2 Shell commands in /sbin

Figure C–3 lists the shell commands in **/usr/sbin** that stand-alone **sysadm** supports:

devnm	gridman	swapon
dg_sysctl	mkfs	syslogd
exportfs	probedev	xdrtoc

Figure C–3 Shell commands in /usr/sbin

Figure C–4 lists the shell commands in **/usr/bin** that stand-alone **sysadm** supports:

admdefault	cut	idi_confirm	rmdir
admdevice	date	idi_doop	sde_target
admfilesystem	dc	idi_echo	sed
admkernel	dd	idi_error	sort
admpackage	df	idi_log	stty
admpdisk	diff	idi_warning	sync
admrelease	dirname	ifconfig	tail
admservice	du	kill	tar
admtape	ed	ln	tee
admvdisk	egrep	logger	touch
awk	expr	ls	tput
basename	false	mkdir	tr
cat	find	mt	true
chgrp	grep	mv	tty
chmod	gunzip	netinit	uncompress
chown	gzip	newaliases	uniq
comm	head	ping	who
compress	pmtd	hostname	xargs
cp	id	printf	csh
cpio	idc	pwd	rm
idi	csh		

Figure C–4 Shell commands in /usr/bin

Some of the commands listed in the tables above are available immediately upon escaping to a shell after entering standalone **sysadm**. Others, like **cpio**, are available only if you resolve the link that **cpio** in the memory resident **/usr/bin** points to.

Mounting the / and /usr file systems on disk before escaping to a shell resolves the link and allows you to access the **cpio** command. Follow the procedures below to do this:

1. From the standalone **sysadm** main menu, select `Install Software` and then select `2 Prepare Virtual Disks` from the Install System Software menu.

```
                Install System Software Menu

1  Prepare physical disks ...
2  Prepare virtual disks ...
3  Load software ...
4  Set up software ...
5  Configure and Build kernel ...
6  Reboot kernel ...
7  All steps

Enter a number, a name, ? or <number>? for help, <NL> to take the
default, ^ to return to previous menu, or q to quit: [All steps] 2
Register all standard physical disks? [yes]
```

2. In the Prepare Virtual disks dialogue, make sure to use the virtual disks that correspond to your **root**, **swap**, and **usr** virtual disks. If the names are correct, press return <NL>.

```
Required File System Mount Points:

File System    Virtual    Current  Action    Blocks    Physical
Mount Point    Disk       Blocks   Required  To Add    Disk
————————————   ———————————————   ————————  ————————  ————————  –
–none–         swap       100000   None      –         sd(ncsc(0,7),0,0)
/              root       100000   None      –         sd(ncsc(0,7),0,0)
/usr            usr       340000   None      –         sd(ncsc(0,7),0,0)

Modify this information? [no]

Beginning swapping on /dev/dsk/swap.
Mounting /dev/dsk/root.
Mounting /dev/dsk/usr.
```

3. Escape to the shell.

Install System Software Menu

1 Prepare physical disks ...
2 Prepare virtual disks ...
3 Load software ...
4 Set up software ...
5 Configure and Build kernel ...
6 Reboot kernel ...
7 All steps

Enter a number, a name, ? or <number>? for help, <NL> to take the default, ^ to return to previous menu, or q to quit: [All steps] **!sh**
#

After escaping to a shell as shown above, **cpio** and the other commands in the tables should be accessible.

End of Appendix

D System configuration data worksheets

Record the information collected from Chapter 2 in these worksheets for your system administration records. You may need to photocopy these worksheets, making duplicate copies to record this data at regular intervals.

System Administrator: _____ **Today's date:** _____

To find out:	Write down:
Observe power-up output	
Processor speed	
Number of processors	
Firmware revision	
Ethernet address	
Type: uname –a	
Operating system name	
Host name	
Operating system version	
Machine hardware name	
Host processor type	
Physical memory size	
Type: who –r	
Run level	
Type: who –b	
When system last booted	
Type: ls –ail [*kernel-that-was-booted*]	
When kernel linked to **/dgux** was last built	
Type: systemid	
Vendor stamp	
System identifier	

Type: dg_sysctl	
To find out:	**Write down:**
AUTOBOOT	
BOOTPATH	
AUTODUMP	
DUMPDEV	
DUMPLEVEL	

Type: admpdisk –o list					
To find out each of the following, and then write down:					
Disk name	**State**	**Registered?**	**Format**	**Total blocks**	**Free blocks**

Type: sysdef [*kernel-that-was-booted*]		
To find out names of configured devices, and then write down:		

Type: admvdisk –o list			
To find out each of the following, and then write down:			
Virtual disk name	**Volume?**	**Size**	**Type**

Type: admfilesystem –o list –m		
To find out each of the following, and then write down:		
File system source	**Mount directory**	**File system type**

Type: df –k *file-system*		
To find out each of the following, and then write down:		
File system	**Percentage used**	**Mount point**

Type: df –t *file-system*		
To find out each of the following, and then write down:		
File system	**Blocks used**	**Total blocks**

Type: admfsinfo –o diskuse –l			
To find out each of the following, and then write down:			
File system	**Free blocks**	**Total blocks**	**Percentage used**

Type: du –s *file-or-directory*	
To find out:	**Write down:**
File or directory name	**Number of disk blocks**

Type: ifconfig *LAN-controller*	
To find out:	**Write down:**
LAN controller name	**Configured?** **If there is output, the controller is configured.**

Type: netstat –i	
To find out network device names and write down:	

Type: ping *host*	
To find out:	**Write down:**
Host name	**Operational?** **Does report indicate that host is alive?**

Type: arp *host*		
To find out each of the following, and then write down:		
Hostname	**Internet address**	**Ethernet address (Also known as hardware address)**

Type: admpackage –o list –c *host*		
To find out each of the following, and then write down:		
Package name	**Release loaded**	**Date loaded**

Type: admpackage –o list –c *OS-client*	
To find out the following, and then write down:	
OS client name	**Packages loaded**

Type: domainname
To find out domain name, and then write down:

End of Appendix

E Data General Software Trouble Report (STR) form

Service Agreement Number _____
Originator ID Number _____

Customer Information

Customer name: _____ Today's date: _____

Job title: _____ Date of problem: _____

Company name: _____ Phone number: _____

E-mail address: _____ Internal mail code: _____

Postal address: _____ City: _____

State/province: _____ Country: _____ Zip: _____

If a Data General employee files the STR on your behalf, please supply the following:

Data General originator's name: _____ E-mail address: _____

Office location: _____ Postal address: _____

Country: _____ Zip: _____ Phone number: _____

Product and System Information

Product name: _____ Model: _____ Revision: _____

Operating system: _____ Revision: _____

Other software in operation: _____ Revision: _____

_____ Revision: _____

_____ Revision: _____

_____ Revision: _____

Which patches/MUs have you applied? _____

List any software obtained/installed from a non-Data General source: _____

_____

Hardware Configuration

CPU Model/Type: _____ Memory Size: _____ Firmware Revision: _____

Data General peripherals (e.g., disks/tapes, async/sync boards, terminals/printers): _____

_____

_____

_____

Non-Data General suppliers and peripherals: _____

Problem Description

Problem type:

_____ Enhancement request

_____ Documentation

_____ Unknown

_____ Question/clarification

_____ System failure

_____ Hang

_____ Software error

_____ If hang, method used to force a memory dump?

_____

Briefly summarize the problem: _____

_____

_____

Frequency:

_____ Frequent

_____ Occasional

_____ Erratic

_____ Reproducible

_____ Not applicable

What is the impact of this problem? _____

_____

What caused the problem? How can we reproduce it? Did you change your configuration prior to the onset of the problem? If so, describe the change:

_____
_____

Attachments and Quantities

_____ None _____ Additional pages _____ Listings

_____ Magnetic tape _____ Cartridge tape _____ Diskette

_____ Console output _____ Other (specify): _____

Please label attachments using the following model as a guide:

BLUE DAEMON SYSTEMS, INC., Durham, N.C. 27709
Originator ID: 300456
Halt code: 3400002
Date: July 6, 1995
File 1: Kernel executable
Density: QIC-525 tape at high density
cpio format: cpio –oBcv
Tape 1 of 3

NOTE: If you have the DG/UX information security product (B2 or C2) installed on your system, the **cpio** command options would be modified as follows: **cpio –oZBcv**.

Recovery Procedures for Typical Problems

Hang:

1. Force a dump memory condition with the ABORT key or hot-key sequence.

   Hot key sequence: Press and hold the Ctrl key while producing these keystrokes:

   ] [ ] [ ] [ ] [

   NOTE: Do not press the space bar between keystrokes.

2. Using the strategy you selected for handling halts, dump memory to tape. For more information on halt behavior, see *Preventing and Fixing Problems on the DG/UX System*.

3. Copy the kernel executable (**/dgux**) to tape. Assuming that the memory dump file occupies file 0, copy the kernel executable to file 1.

   ex. # **mt –f /dev/rmt/0n fsf 1**↵
   # **cd /** ↵
   # **echo dgux | cpio –oBcv > /dev/rmt/0n** ↵

   If you have the DG/UX information security product (B2 or C2) installed on your system, you must include the Z option with the **cpio** command:
   # **echo dgux | cpio –oZBcv > /dev/rmt/0n** ↵

4. Submit a copy of **/var/adm/messages** and other materials that can be used to reproduce the problem.

Halt:

1. Record the halt code: _____

2. Using the strategy you selected for handling halts, dump memory to tape. For more information on halt behavior, see *Preventing and Fixing Problems on the DG/UX System*.

3. Copy the kernel executable (**/dgux**) to tape. Assuming that the memory dump file occupies file 0, copy the kernel executable to file 1.

   ex. # **mt –f /dev/rmt/0n fsf 1**↵
   # **cd /** ↵
   # **echo dgux | cpio –oBcv > /dev/rmt/0n** ↵

   If you have the DG/UX information security product (B2 or C2) installed on your system, you must include the Z option with the **cpio** command:

   # **echo dgux | cpio –oZBcv > /dev/rmt/0n** ↵

User program or DG/UX system core dump:

1. Copy the core file to tape file location 0, using the following command formats:

   e.g., # **find / –name core –print**↵
   # **mt –f /dev/rmt/0n fsf** ↵
   # **cd** *location-of-core-file* ↵
   # **echo** *core-file-name* **| cpio –oBcV > /dev/rmt/0n**↵

   If you have the DG/UX information security product (B2 or C2) installed on your system, you must include the Z option with the **cpio** command:

   # **echo dgux | cpio –oZBcv > /dev/rmt/0n** ↵

   NOTE: If you already have dumped memory and the kernel executable to tape, make sure that you specify the correct tape file number in the **mt** command. To copy the core dump to the third file location on the tape, use the command **mt –f /dev/rmt/0n fsf 2**.

2. Copy the command executable that caused the core dump. Assuming that the core file occupies file 0, copy the command executable to file 1, using the following command formats:

   e.g., # **mt –f /dev/rmt/0n fsf 1** ↵
   # **cd /** ↵
   # **echo** *script-name* **| cpio –oBcV > /dev/rmt/0n**↵

3. Identify the command executable that caused the core dump, using the following command format:

   e.g., # **what** *command-path* ↵

This STR form also is available on your DG/UX system at **/usr/release/STR_form**.

See *Preventing and Fixing Problems on the DG/UX® System* for more information on filing an STR.

Return to: Data General Corporation, 1626 Juergens Ct., Norcross, GA 30093, Attn: STR Administrator

End of Appendix

Index

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X

Z

Preventing and Fixing Problems on the DG/UX® System

069–701145–02

Preventing and Fixing Problems on the DG/UX® System