Bull

AIX 4.3 Problem Solving Guide and Reference

AIX

ORDER REFERENCE
86 A2 32JX 02

Bull

AIX 4.3 Problem Solving Guide and Reference

AIX

Software

November 1999

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 32JX 02

The following copyright notice protects this book under the Copyright laws of the United States of America and other countries which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

Copyright © Bull S.A. 1992, 1999

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

To order additional copies of this book or other Bull Technical Publications, you are invited to use the Ordering Form also provided at the end of this book.

Trademarks and Acknowledgements

We acknowledge the right of proprietors of trademarks mentioned in this book.

AIX® is a registered trademark of International Business Machines Corporation, and is being used under licence.

UNIX is a registered trademark in the United States of America and other countries licensed exclusively through the Open Group.

Year 2000

The product documented in this manual is Year 2000 Ready.

The information in this document is subject to change without notice. Groupe Bull will not be liable for errors contained herein, or for incidental or consequential damages in connection with the use of this material.

Table of Contents

About This Book

The *AIX Version 4.3 Problem Solving Guide and Reference* provides information to help you investigate, define, and fix problems with the operating system, software or hardware, or other programs installed on the system. Follow the procedures in Chapter 1, 'Problem Solving Overview,' before placing a service call.

How to Use This Book

Note: Before beginning problem determination, make a copy of the Problem Summary Form located in Chapter 1 to record problem information.

If a three–digit value is displayed on your system unit or if a message is displayed on your screen, go directly to the *AIX Messages Guide and Reference* for information about the message and recovery procedures.

This book enables you to locate problem solving instructions and tools according to your needs and knowledge level. Chapter 1 contains prerequisite information for the rest of the book including the Symptom Index and the Problem Summary Form. Chapters 2 through 8 provide troubleshooting procedures for specific problems. Chapters 9 through 12 provide tools for investigating system problems and require some programming knowledge.

Overview of Contents

Directed Problem Analysis

- Chapter 1, 'Problem Solving Overview,' is the starting point for problem determination. You locate the general symptom of your problem in the Symptom Index and are directed to problem resolution steps in one of the following chapters. Instructions for installing the Software Service Aids Package are also included
- Chapter 2, 'Inactive System,' describes how to check your system hardware and the processes running on the system and then restart your system after diagnosing the problem area.
- Chapter 3, 'Terminal Problems,' describes how to free a terminal taken over by processes and respond to screen messages.
- Chapter 4, 'Inaccurate System Clock,' describes how to reset the system clock by testing the system battery or resetting the date and setclock commands.
- Chapter 5, 'Device Problems,' describes how to check the hardware and software that could affect device performance.
- Chapter 6, 'Media Problems,' describes how to check for media errors and recover data from a damaged diskette.
- Chapter 7, 'Network Problems,' describes how to solve network problems by diagnosing network hardware problems, network software problems, and network communication adapter problems.
- Chapter 8, 'Printing Problems,' describes how to reactivate the queue daemon, clear a print queue backlog, reallocate printer resources, and delete unnecessary files in the printer directory.

Problem Solving Tools

- Chapter 9, 'Hardware Diagnostics,' describes how to diagnose your system by loading a diagnostics program, choosing a test option, and reporting a service request number

(SRN). Information is also included to help you identify your hardware, interpret codes in the three–digit display, and gather information about your system configuration.

- Chapter 10, 'Error Logging Facility,' describes how the error logging facility can be used to record hardware and software problems and provide informational messages for system operation. Also included is information on error logging and alerts and programs used to run an automatic error log analysis.
- Chapter 11, 'Trace Facility,' describes how to use the trace facility to isolate system problems by monitoring system events.
- Chapter 12, 'System Dump Facility,' describes the system dump facility and how to configure a dump device, start a system dump, copy a system dump, and return to normal operations.
- Chapter 13, 'Remote Reboot Facility,' describes how the remote reboot facility allows the system to be rebooted through a native (integrated) serial port.

Reference Information

- Appendix A, 'Error Message Acronyms,' describes the acronyms used in this book.
- Appendix B, 'Software Validation,' describes how to verify that an installed software product (or licensed product) has not been damaged or corrupted.
- Appendix C, 'Error Identifiers for the Error Log,' lists the possible error log entries by error label.
- Appendix D, 'Recovering Volume Groups,' describes how to recover volume groups when they are corrupted.

Highlighting

The following highlighting conventions are used in this book:

Bold	Identifies commands, keywords, files, directories and other items whose names are predefined by the system.
Italics	Identifies parameters whose actual names or values are to be supplied by the user.
`Monospace`	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Related Publications

The following books contain information about or related to problem solving:

- *AIX and Related Products Documentation Overview*, order number 86 A2 71WE.
- The operator guide for your system unit contains the checkout procedures for problem determination, system verification, and using the diagnostics.
- *AIX Messages Guide and Reference*, order number 86 A2 33JX.
- *AIX 4.3 System Management Guide: Operating System and Devices*, order number 86 A2 99HX.
- *AIX 4.3 System Management Guide: Communications and Networks*, order number 86 A2 31JX.

- *AIX Performance Tuning Guide*, order number 86 A2 72AP.
- *Performance Toolbox 1.2 and 2.1 for AIX: User's Guide*, order number 86 A2 10AQ.
- *AIX General Programming Concepts : Writing and Debugging Programs*, order number 86 A2 34JX.
- *Common Diagnostics Information Manual for MCA Systems*, order number 86 A2 75AT.
- *AIX Communications Programming Concepts*, order number 86 A2 35JX
- *AIX Guide to Printers and Printing*, order number 86 A2 37JX
- *AIX Commands Reference*, order number 86 A2 38JX to 86 A2 43JX
- *AIX Kernel Extensions and Device Support Programming Concepts*, order number 86 A2 36JX
- *3270 Information Display System: 3274 Control Unit Description and Programmer's Guide*

Ordering Publications

You can order publications from your sales representative or from your point of sale.

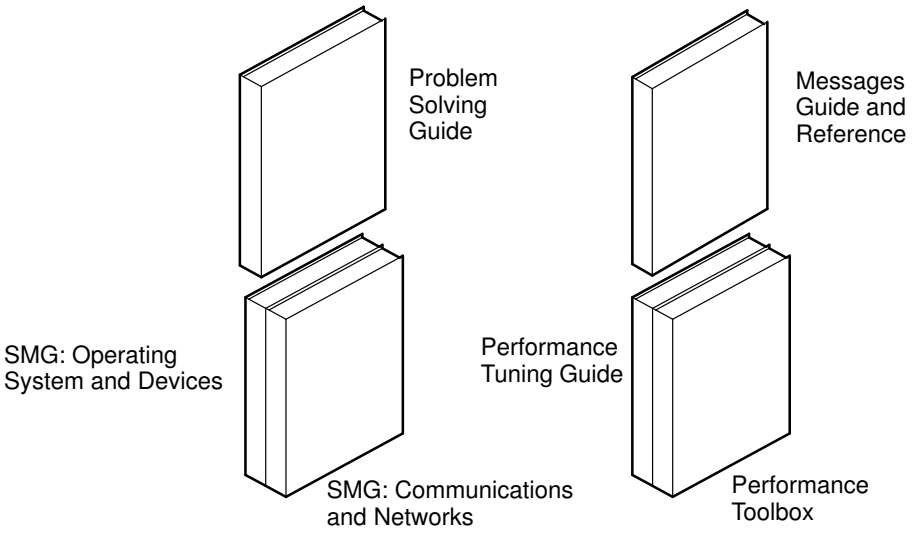
To order additional copies of this book, use order number 86 A2 32JX.

Chapter 1. Problem Solving Overview

This chapter directs you to the troubleshooting information most applicable to your system's problem. Before beginning any problem solving procedures, refer to the following:

Library Contents

This book addresses various system problems but is not a substitute for troubleshooting procedures included in other reference books. Other problem solving references include the following:



Problem Solving Source Overview

- The *AIX Messages Guide and Reference* includes reference and recovery information for seven–digit error messages and three–digit values in the operator panel display.
- The *AIX 4.3 System Management Guide: Operating System and Devices* contains procedures for troubleshooting logical volume, file system, system accounting, printing and general device problems.
- The *AIX 4.3 System Management Guide: Communications and Networks* contains procedures for troubleshooting general communications and network problems.
- The *AIX Performance Tuning Guide* describes the concepts and techniques involved in assessing and tuning the performance of AIX systems.
- The *Performance Toolbox 1.2 and 2.1 for AIX: User's Guide* describes a graphical interface to a collection of performance monitoring and tuning tools.

Refer to the Symptom Index to start your problem solving procedures or to determine the best source of information for your system's problem.

Symptom Index

Use the Symptom Index as your problem solving starting point. Select the symptom most closely matching the problem with your system, record this symptom in item 3 of the Problem Summary Form, and follow the indicated problem solving procedures.

Seven–Digit Error Number with Message Text	Refer to the *AIX Messages Guide and Reference*.
Three–Digit Value (in Operator Panel Display)	Refer to the *AIX Messages Guide and Reference*.
Inactive System	Refer to Check the Hardware, on page 2-2 Refer to Check the Processes, on page 2-3 Refer to Restart the System, on page 2-5
Terminal Problems	Refer to Free a Terminal Taken Over by Processes, on page 3-2 Refer to Respond to Screen Messages, on page 3-3
Inaccurate System Clock	Refer to Test the System Battery, on page 4-2 Refer to Reset the System Clock, on page 4-3
Device Problems	Refer to Check the Device Hardware, on page 5-2 Refer to Check the Device Software, on page 5-3
Media Problems	Refer to Check for Media Errors, on page 6-2 Refer to Recover Data from a Damaged Diskette, on page 6-4
Network Problems	Refer to Resolve Network Software Problems, on page 7-2 Refer to Resolve Network Hardware Problems, on page 7-7 Refer to Resolve Communication Adapter Problems, on page 7-8
Printing Problems	Refer to Reactivate the Queue Daemon, on page 8-2 Refer to Clear a Print Queue Backlog, on page 8-3 Refer to Reallocate Printer Resources, on page 8-5 Refer to Delete Unnecessary Directory Files, on page 8-6
Hardware Diagnostics	Refer to Selecting and Running Diagnostics, on page 9-4 Refer to Identify Your Hardware, on page 9-11 Refer to Interpret Values in the Operator Panel, on page 9-12 Refer to Gather Information about Your System Configuration, on page 9-16
You Need to Retrieve/Generate an Error Log Report	Refer to Error Logging Facility, on page 10-1
You Need to Generate a Trace Report	Refer to Trace Facility, on page 11-1
You Need to Retrieve a System Dump	Refer to System Dump Facility, on page 12-1

Software Service Aids Package

In AIX Version 4, some of the error log and dump commands are delivered in an optionally installable package called **bos.sysmgt.serv_aid**. The base system (**bos.rte**) includes the services for logging errors to the error log file. This includes the **errlog** subroutines, the **errsave** and **errlast** kernel service, the error device driver (/**dev**/**error**), the error daemon, and the **errstop** command. The commands required for licensed program installation (**errinstall** and **errupdate**) are also included in **bos.rte**. System dump commands included in the **bos.sysmgt.serv_aid** include the **sysdumpstart** command.

To determine if this package is installed, run the following command:

```
lslpp -l | grep bos.sysmgt.serv_aid
```

If this package is installed, a line containing the following will be displayed:

```
bos.sysmgt.serv_aid
```

If the package is not installed, you can use the System Management Interface Tool (SMIT) dialogs to install the **bos.sysmgt.serv_aid** package. If SMIT is not installed, use the **installp** command:

```
installp -xad <installation device> bos.sysmgt.serv_aid
```

Related Information

Because the **errclear**, **errdead**, **errlogger**, **errmsg**, and **errpt** commands are part of the optionally installable Software Service Aids package (**bos.sysmgt.serv_aid**), you need the Software Service Aids package to generate reports from the error log or delete entries from the error log. For information on transferring your system's error log to another system that has the Software Service Aids package installed, see "Transferring Your Errror Log to Another System" , on page 10-4.

Problem Summary Form

Background Information

1. Record the Current Date and Time _____
2. Record the System Date and Time (if available) _____
3. Record the Symptom _____
4. Record the Service Request Number (SRN) or error codes
   _____
5. Record the Three–Digit Codes (if available) __ __ __ – __ __ __ – __ __ __ –__ __ __
6. Record the Location Codes

- First FRU__ __–__ __–__ __–__ __
- Second FRU__ __–__ __–__ __–__ __
- Third FRU__ __–__ __–__ __–__ __
- Fourth FRU__ __–__ __–__ __–__ __

Problem Description

Data Captured

(Describe data captured, such as system dumps, core dumps, error IDs, error logs, or messages, that needs to be examined by your service organization.)

(After completing this form, copy it and keep it on hand for future problem solving reference.)

Chapter 2. Inactive System

A hardware problem, a software problem, or a combination of both can make your system inactive. This procedure guides you through steps to correct the problem and restart your system. If your system is still inactive after completing the procedure, refer to the hardware problem determination procedure in your system operator guide.

Procedures to reactivate an inactive system includes:

- Check the Hardware, on page 2-2
- Check the Processes, on page 2-3
- Restart the System, on page 2-5

If your system will not boot after performing the preceding procedures, refer to 'Accessing a System That Does Not Boot' in the *AIX Messages Guide and Reference*.

Check the Hardware

Check your hardware by doing the following:

- Checking the Power, on page 2-2
- Checking the Operator Panel Display if available, on page 2-2
- Activating Your Display or Terminal, on page 2-2

Checking the Power

Is the Power–On light on your system?

Yes	Go to the next question.
No	Check if the power is on and the system is plugged in. If you do not find a problem, go to ”Hardware Diagnostics”, on page 9-1.

Checking the Operator Panel Display

Is the Operator Panel Display on your system blank?

Yes	Go to the next question.
No	Go to the *AIX Messages Guide and Reference* for information concerning digits in the Operator Panel Display.

Activating Your Display or Terminal

Check several parts of your display or terminal:

- Make sure the display cable is securely attached to the display and to the system unit.
- Make sure the keyboard cable is securely attached.
- Make sure the mouse cable is securely attached.
- Make sure the display is turned on and that its Power–On light is lit.
- Adjust the brightness control on the display.
- Make sure the terminal’s communication settings are correct.

Is your system now active?

Yes	Your hardware checks have corrected the problem.
No	Did your system become inactive while you were trying to restart the system?

Yes	Go to ”Restart the System”, on page 2-5.
No	Go to ”Check the Processes”, on page 2-3.

Check the Processes

A stopped or stalled process might make your system inactive. Check your system processes by:

- Restarting Line Scrolling, on page 2-3
- Using the Ctrl–D Key Sequence , on page 2-3
- Using the Ctrl–C Key Sequence, on page 2-3
- Logging In from a Remote Terminal or Host, on page 2-3
- Ending Stalled Processes Remotely, on page 2-4

Restarting Line Scrolling

Restart line scrolling halted by the Ctrl–S key sequence by doing the following:

1. Activate the window or shell with the problem process.
2. Press the Ctrl–Q key sequence to restart scrolling.

The Ctrl–S key sequence stops line scrolling and the Ctrl–Q key sequence restarts scrolling.

Is your system now active?

Yes	Your scroll check corrected the problem with your inactive system.
No	Go to the next step, ”Using the Ctrl–D Key Sequence” , on page 2-3.

Using the Ctrl–D Key Sequence

End a stopped process by doing the following:

1. Activate the window or shell with the problem process.
2. Press the Ctrl–D key sequence. The Ctrl–D key sequence sends an end of file (EOF) to the process. The Ctrl–D key sequence may close the window or shell and log you out.

Is your system now active?

Yes	The Ctrl–D key sequence corrected the problem with your inactive system.
No	Go to the next step,”Using the Ctrl–C Key Sequence” , on page 2-3.

Using the Ctrl–C Key Sequence

End a stopped process by doing the following:

1. Activate the window or shell with the problem process.
2. Press the Ctrl–C key sequence. The Ctrl–C key sequence stops the current search or filter.

Is your system now active?

Yes	The Ctrl–C key sequence corrected the problem with your inactive system.
No	Go to the next step, ”Logging In from a Remote Terminal or Host”, on page 2-3.

Logging In from a Remote Terminal or Host

Log in remotely in either of two ways:

- Log in to the system from another terminal if more than one terminal is attached to your system.

- Log in from another host on the network (if your system is connected to a network) by entering the following **tn** command:

  ```
  tn YourSystemName
  ```

  The system asks for your regular login name and password when you use the **tn** command.

Were you able to log in to the system from another terminal or host?

Yes	Go to the next step, ”Ending Stalled Processes Remotely”, on page 2-4.
No	Go to ”Restart the System”, on page 2-5. Or, you can start a system dump to find out why your system became inactive. Go to ”System Dump Facility”, on page 12-1 for more information.

Ending Stalled Processes Remotely

End a stalled process from a remote terminal by doing the following:

1. List active processes by entering the following **ps** command:

   ```
   ps –ef
   ```

   The **–e** and **–f** flags identify all active and inactive processes.

2. Identify the process ID of the stalled process.

   For help identifying processes, you can use the **grep** command with a search string. For example, to quit the **xlock** process, enter the following to find the process ID:

   ```
   ps –ef | grep xlock
   ```

   The **grep** command allows you to search on the output from the **ps** command to identify the process ID of a specific process.

3. End the process by entering the following **kill** command:

   Note: You must have root user authority to use the **kill** command on processes you did not initiate.

   ```
   kill –9 ProcessID
   ```

   If you cannot identify the problem process, the most recently activated process might be the cause of your inactive system. End the most recent process if you think that is the problem.

Is your system now active?

Yes	Your process checks have corrected the problem with your inactive system.
No	Go to ”Restart the System”, on page 2-5. Or, you can start a system dump to find out why your system became inactive. Go to ”System Dump Facility”, on page 12-1 for more information.

Restart the System

You need to restart your system if the first two procedures fail to correct the problem that makes your system inactive.

Note: Before restarting your system, you need to complete a system dump. Go to ”System Dump Facility”, on page 12-1 for more information.

This procedure involves:

- Checking the Position of the Mode Switch, if available, on page 2-5
- Checking the State of the Boot Device, on page 2-5
- Loading the Operating System, on page 2-5

Checking the Position of the Mode Switch, if available

The correct position for the Mode Switch depends on the type of software you want to load.

Position the Mode Switch according to one of the following conditions:

- Use the Normal position to load the operating system.
- Use the Service position to boot the system from maintenance mode or hardware diagnostics (refer to ”Hardware Diagnostics”, on page 9-1).

Go to ”Checking the State of the Boot Device”, on page 2-5.

Checking the State of the Boot Device

Your system boots with either a removable medium, an external device, a small computer system interface (SCSI) device, an integrated device electronics (IDE) device, or a local area network (LAN). Decide which method applies to your system and use the following instructions to check the boot device:

- For a removable medium, such as tape, make sure the medium is inserted correctly.
- For IDE devices, verify that the IDE device id settings are unique per adapter. If only one device is attached to the adapter, the IDE device id must be set to the master device.
- For an externally attached device, such as a tape drive, make sure:
  - The power to the device is turned on.
  - The device cables are correctly attached to the device and to the system unit.
  - The ready indicator is on (if the device has one).
- For external SCSI devices, verify that the SCSI address settings are unique.
- For a LAN, verify that the network is up and operable.

Is the boot device working correctly?

Yes	Go to ”Loading the Operating System”, on page 2-5.
No	Go to ”Hardware Diagnostics”, on page 9-1.

Loading the Operating System

Load your AIX operating system by doing the following:

1. Switch off your system’s power.
2. Wait 1 minute.
3. Switch on your system’s power.
4. Wait about 5 minutes while the system boots.

Did your system boot successfully?

Yes Your steps to restart the system have corrected the problem with your inactive system.

No If the AIX operating system failed to load, boot the hard disk from Maintenance Mode or load or hardware diagnostics.

To load hardware diagnostics, refer to ”Hardware Diagnostics”, on page 9-1.

If you are still unable to restart the system, you can use an SRN to report the problem with your inactive system to your service representative. Go to ”Hardware Diagnostics”, on page 9-1 to learn how to get an SRN.

Chapter 3. Terminal Problems

Stalled or unwanted processes can cause problems with your terminal. Some problems produce messages on your screen that give information about possible causes.

To perform the following procedures, you must have either a second terminal, a modem, or a network login. If you do not have any of these, fix the terminal problem by rebooting your machine.

Choose the appropriate procedure for fixing your terminal problem:

Free a Terminal Taken Over by Processes

Identify and stop stalled or unwanted processes by doing the following:

1. Determine the active processes running on the screen by using the following **ps** command:

   ```
   ps -ef | pg
   ```

   The **ps** command shows the process status. The **–e** flag writes information about all processes (except kernel processes), and the **–f** flag generates a full listing of processes including what the command name and parameters were when the process was created. The **pg** command limits output to a single page at a time, so information does not quickly scroll off the screen.

   Suspicious processes include system or user processes that use up excessive amounts of a system resource such as CPU or disk space. System processes such as **sendmail**, **routed**, and **lpd** frequently become runaways. Use the **ps u** command to check CPU usage.

2. Determine who is running processes on this machine by using the **who** command:

   ```
   who
   ```

   The **who** command displays information about all users currently on this system, such as login name, workstation name, date, and time of login.

3. Determine if you need to stop, suspend, or change the priority of a user process.

   Note: You must have root authority to stop processes other than your own. If you terminate or change the priority of a user process, contact the process owner and explain what you have done.

   – Stop the process using the **kill** command. For example:

   ```
   kill 1883
   ```

   The **kill** command sends a signal to a running process. To stop a process, specify the process ID (PID), which is 1883 in this example. Use the **ps** command to determine the PID number of commands.

   – Suspend the process and run it in the background by using the ampersand (&). For example:

   ```
   /u/bin1/prog1 &
   ```

   The **&** signals that you want this process to run in the background. In a background process, the shell does not wait for the command to complete before returning the shell prompt. When a process requires more than a few seconds to complete, run the command in background by typing an **&** at the end of the command line. Jobs running in the background appear in the normal **ps** command.

   – Change the priority of the processes that have taken over by using the following **renice** command:

   ```
   renice 20 1883
   ```

   The **renice** command alters the scheduling priority of one or more running processes. The higher the number, the lower the priority with 20 being the lowest priority.

   In the previous example, **renice** reschedules process number 1883 to the lowest priority. It will run when there is a small amount of unused processor time available.

4. If your terminal does not return to normal operation after you complete the above procedures, follow the diagnostic procedures in ”Hardware Diagnostics”, on page 9-1.

Respond to Screen Messages

Respond to and recover from screen messages by doing the following:

1. If the terminal displays any error messages, look up the messages in the *AIX Messages Guide and Reference*.

1. Make sure the **DISPLAY** environment variable is set correctly. Use either of the following methods to check the **DISPLAY** environment:

   – Use the **setsenv** command to display the environment variables.

   ```
   setsenv
   ```

   The **setsenv** command displays the protected state environment when you logged in.

   Determine if the **DISPLAY** variable has been set. In the following example, the **DISPLAY** variable does not appear, which indicates that the **DISPLAY** variable is not set to a specific value.

   ```
   SYSENVIRON:
   NAME=casey
   TTY=/dev/pts/5
   LOGNAME=casey
   LOGIN=casey
   ```

   OR

   – Change the value of the **DISPLAY** variable. For example, to set it to the machine named `bastet` and terminal 0, enter:

   ```
   DISPLAY=bastet:0
   export DISPLAY
   ```

   If not specifically set, the **DISPLAY** environment variable defaults to `unix:0` (the console). The value of the variable is in the format *name:number* where *name* is the host name of a particular machine, and *number* is the X server number on the named system.

2. Reset the terminal to its defaults using the following **stty** command:

   ```
   stty sane
   ```

   The **stty sane** command restores the 'sanity' of the terminal drivers. The command outputs an appropriate terminal resetting code from the /**etc**/**termcap** file (or /**usr**/**share**/**lib**/**terminfo** if available).

3. If the Return key does not work correctly, reset it by entering:

   ```
   ^J stty sane ^J
   ```

   The ^J represents the Ctrl–J key sequence.

4. If your screen does not return to normal operation after you complete the preceding procedures, follow the diagnostic procedures in 'Hardware Diagnostics', on page 9-1.

Chapter 4. Inaccurate System Clock

The system clock records the time of system events, allows you to schedule system events (such as running hardware diagnostics at 3:00 a.m.), and tells when you first created or last saved files. To reset or reactivate an inaccurate clock, you can:

- Test the System Battery, on page 4-2
- Reset the System Clock, on page 4-3

Test the System Battery

If your system is losing track of time, the cause may be a depleted or disconnected battery. To determine the status of your system battery, enter the following **diag** command:

```
diag -B -c
```

When the Diagnostics main menu appears, select the **Problem Determination** option. If the battery is disconnected or depleted, a problem menu will be displayed with a service request number (SRN). Record the SRN on Item 4 of the Problem Summary Form and report the problem to your hardware service organization.

If your system battery is operational, your system time may have been reset incorrectly because either the **date** or **setclock** command was run incorrectly or unsuccessfully. Refer to ”Reset the System Clock”, on page 4-3 to correct the problem.

Reset the System Clock

Use the **date** command to set your system clock. Use the **setclock** command to set the time and date for a host on a network. Reset your system clock by either:

- Using the date Command, on page 4-3
- Using the setclock Command, on page 4-3

Using the date Command

The **date** command displays or sets the date and time. Enter the following command to determine your system's date and time:

```
/usr/bin/date
```

Attention: Do not change the date when the system is running with more than one user.

The following formats can be used when setting the date with the *Date* parameter:

- *mmddHHMM*[.*SSyy*] (default)
- *yymmddHHMM*[.*SS*]
- *ddmmHHMMyy*[.*SS*]

The variables to the *Date* parameter are defined as follows:

mm	Specifies the month number.
dd	Specifies the number of the day in the month.
HH	Specifies the hour in the day (using a 24–hour clock).
MM	Specifies the minute number.
SS	Specifies the number of seconds.
yy	Specifies the last two numbers of the year.

Note: If the *yymmdd* format is specified, the value of the *yy* variable must be 88 to 99.

The **date** command writes the current date and time to standard output if called with no flags or with a flag list that begins with a **+** (plus sign). Otherwise, it sets the current date. Only a root user can change the date and time. The **date** command prints out the usage message on any unrecognized flags or input.

If you follow the **date** command with a + (plus sign) and a field descriptor, you can control the output of the command. You must precede each field descriptor with a **%** (percent sign). The system replaces the field descriptor with the specified value. Enter a literal % as %% (two percent signs). The **date** command copies any other characters to the output without change. The **date** command always ends the string with a new–line character.

Flags

–n	Does not set the time globally on all machines in a local area network that have their clocks synchronized.
–u	Displays or sets the time in Coordinated Universal Time (UTC).

Using the setclock Command

```
/usr/sbin/setclock
```

The /**usr**/**sbin**/**setclock** command gets the time from a network time server, and if run by a user with root user authority, sets the local time and date accordingly.

The **setclock** command takes the first response from the time server, converts the calendar clock reading found there, and shows the local date and time. If the **setclock** command is run by the root user, it calls the standard workstation entry points to set the system date and time.

If no time server responds, or if the network is not operational, the **setclock** command displays a message to that effect and leaves the date and time settings unchanged.

Note: Any host running the **inetd** daemon can act as a time server.

Parameter

TimeServer The host name or address of a network host that services TIME requests. The **setclock** command sends a public network TIME service request to a time server host. If the *TimeServer* name is omitted, the **setclock** command sends the request to the default time server. The default time server in a DOMAIN environment is specified by the name server. Otherwise, the default time server is specified in the /**etc**/**hosts** file.

Chapter 5. Device Problems

Problems with devices, such as drives and printers, can affect your system's operation. Find the cause of your device problem by doing the following:

Check the Device Hardware

Correct a device hardware problem by:

- Checking the Device Connections, on page 5-2
- Checking the Ready State of a Device, on page 5-2
- Running Diagnostics on a Device, on page 5-2

Checking the Device Connections

Follow these steps to check your device connections:

1. Check that power is available at the electrical outlet.
2. Check that the device power cable is correctly attached to the device and to the electrical outlet.
3. Check that the device signal cable is attached correctly to the device and to the correct connection on the system unit.
4. For SCSI devices, check that the SCSI terminator is correctly attached and the SCSI address setting is correct.
5. For communications devices, check that the device is correctly attached to the communications line.
6. Check that the device is turned on.

Refer to the publications for the specific device for cabling and configuring procedures and for further troubleshooting information.

Did your checks correct the problem with the device?

Yes	Your checks of the device connections have corrected the problem.
No	Go to the next step, ”Checking the Ready State of a Device”, on page 5-2.

Checking the Ready State of a Device

To determine whether the device is in a ready state:

1. Check that the device’s Ready indicator is on.
2. Check that removable media, such as tape, diskette, and optical devices, are inserted correctly.
3. Check the ribbon, the paper supply, and the toner supply for printers and plotters.
4. Check that the write medium is write–enabled if you are trying to write to the device.

Did your checks correct the problem with the device?

Yes	Your check of the device’s ready state corrected the problem.
No	Go to the next step, ”Running Diagnostics on a Device”, on page 5-2.

Running Diagnostics on a Device

You might have a defective device. Go to ”Hardware Diagnostics” , on page 9-1 to find the hardware problem with your device.

If running hardware diagnostics fails to find a problem with your device, go to ”Check the Device Software”, on page 5-3. If your device passes the diagnostic tests, you might have a problem with the way your device works with your system software. If it is possible that the preceding problem exists, report the problem to your software service organization.

Check the Device Software

Correct a device software problem by:

- Checking the Error Log, on page 5-3
- Listing All Devices, on page 5-3
- Checking the State of a Device, on page 5-3
- Checking the Attributes of a Device, on page 5-3
- Changing the Attributes of a Device, on page 5-4
- Using a Device with Another Application, on page 5-4
- Defining a New Device, on page 5-4

Checking the Error Log

Check the error log to see whether any errors are recorded for either the device, its adapter, or the application using the device. Go to ”Error Logging Facility”, on page 10-1 to perform this check. Return to this step after completing the procedures.

Did you correct the problem with the device?

Yes	Your check of the error log identified the problem.
No	Go to the next step, ”Listing All Devices”, on page 5-3.

Listing All Devices

Use the **lsdev –C** command to list all defined or available devices.

The command shows the characteristics of all the devices in your system.

Is the device in the list of devices?

Yes	Go to the next step, ”Checking the State of a Device”, on page 5-3.
No	Go to ”Defining a New Device”, on page 5-4.

Checking the State of a Device

Find the device in the list generated from the **lsdev –C** command. Check whether the device is in the Available state.

Is the device in the Available state?

Yes	Go to the next step, ”Checking the Attributes of a Device”, on page 5-3.
No	Go to ”Defining a New Device”, on page 5-4.

Checking the Attributes of a Device

Enter the **lsattr –E –l** *DeviceName* command to list the attributes of your device.

The **lsattr** command shows attribute characteristics and possible values of attributes for devices in the system. Refer to the publications for the specific device for the correct settings.

Are the device attributes set correctly?

Yes	Go to ”Using a Device with Another Application”, on page 5-4.
No	Go to the next step, ”Changing the Attributes of a Device”, on page 5-4.

Changing the Attributes of a Device

Enter the **chdev –l** *Name* **–a** *Attribute*=*Value* command to change device attributes.

The **chdev** command changes the characteristics of the device you name with the **–l** *Name* flag. Refer to the *AIX Commands Reference* for information before running this command.

Did you correct the problem with the device?

Yes	Changing the device attributes has corrected the problem.
No	Go to ”Using a Device with Another Application”, on page 5-4.

Using a Device with Another Application

Try using the device with another application. If the device works properly with another application, there might be a problem with the first application.

Did the device work correctly with another application?

Yes	You might have a problem with the first application. Report the problem to your software service representative.
No	Go to ”Defining a New Device”, on page 5-4.

Defining a New Device

Note: You must either have root user authority or be a member of the security group to use the **mkdev** command.

Use the **mkdev** command to add a device to the system.

The **mkdev** command can either define and make available a new device or make available a device that is already defined. You can uniquely identify the predefined device by using any combination of the **–c**, **–s**, and **–t** flags.

Refer to the *AIX Commands Reference* for information before running this command.

Did you correct the problem with the device?

Yes	Defining the device has corrected the problem.
No	You can either stop and report the problem to your service representative or use a diagnostics program to test your device. If you want to diagnose the problem with your device, go to ”Hardware Diagnostics”, on page 9-1

Chapter 6. Media Problems

A media error on a diskette is just like a sector error on a hard drive. Some portion of the diskette has been contaminated or destroyed and has become unusable. Damaged media should not be used to store data.

Media problems can develop in media that has exceeded its useful life. A table of tapes and their life expectancy follows:

Tape	Expectancy
4–mm DAT	Five years or 650 full–tape read or write operations.
8–mm DATA grade	Four years or 500 full–tape read or write operations.
1/4–inch	Not available.
1/2–inch	Not available.

The following information will enable you to:

- Check for Media Errors , on page 6-2
- Recover Data from a Damaged Diskette , on page 6-4

See ”Logical Volume Storage Overview” in *AIX 4.3 System Management Concepts: Operating System and Devices* for problem solving information on logical volume storage (storage system based on the hard disk drive).

Check for Media Errors

Check for media errors by:

- Reading from the Media, on page 6-2
- Writing to the Media , on page 6-2

Reading from the Media

Reading from the media involves detecting the data and format information on a media.

- You can use the **dd** command to check a disk or diskette.

  For example, to read a diskette on drive `rfd0` , enter:

  ```
  dd if=/dev/rfd0 of=/dev/null bs36b
  ```

  where the **bs** flag specifies the internal buffer size for **dd**.

  Note: A value of 36b for this flag greatly enhances the performance of this operation. With a buffer size of 18KB, **dd** can read both tracks of a diskette in only two revolutions of the diskette.

- You can also use the **tcopy** command to read a tape.

  For example, to read a tape on device `rmt0` , enter:

  ```
  tcopy /dev/rmt0
  ```

- Media errors are also recorded in the error log. For example, entries labeled TAPE_ERR1 indicate tape media error. For further information, see ”Error Logging Facility” , on page 10-1 and ”Error Identifiers for the Error Log” , on page C-1.

  If the media can be read successfully with the **dd** or the **tcopy** command, a syntax or usage problem possibly exists. Syntax or usage problems could include:

- Incorrect archive utility was used.
- **cpio** backup was not written with ASCII headers and the command specifies ASCII headers (**–c**).
- **tar** command has calculated a checksum different from what is recorded on the tape indicating that the incorrect volume may be in the drive. (Use the **–i** flag to ignore checksums.)
- For tapes, the DEVICE block size is different than the size of the tape file’s blocks.
- For tapes, the archive utility does not have enough buffer space to hold an entire block from the tape file.

Writing to the Media

Attention: The reformatting procedure destroys all data on the media.

Writing to the media also helps you check for media problems.

- Use the **format** or **dosformat** (formats a DOS diskette) commands to format diskettes.

  For example, to format a diskette in the diskette drive `rfd0` , enter:

  ```
  format –d /dev/rfd0
  ```

  OR

```
dosformat -D /dev/rdf0
```

- Use the **dd** command to write data to a tape.

  For example, to write the file `motd` to the device `rmt0` , enter:

  ```
  dd if=/etc/motd of=/dev/rmt0
  ```

  Note: Writing data to a tape ensures that only that portion of the tape that is written to is not damaged.

Recover Data from a Damaged Diskette

Diskettes can suffer actual physical damage (bending, folding) or internal damage. To recover data from a damaged diskette, do the following:

1. Make sure that you have a backup of the diskette.
2. Get another copy of the data on the diskette, if possible.
3. Restore as many files as possible from the diskette by determining which of the following methods you used to copy the file to the diskette.
   - If the files were written with the **tar –f** command, enter the following command to restore the file:

     ```
     tar -xvf/dev/rfd0
     ```

     The **tar** command writes files to or retrieves files from diskettes as well as tape media.

     The /**dev**/**rfd0** specifies that you want to use removable floppy drive zero (`rfd0`) for your input.

     Note: If you get checksum errors while restoring the data, use the **–i** flag to ignore header checksum errors.

     If the data was written onto the diskette with relative paths, the **tar** command extracts data from a diskette in removable `rfd0` and places it in the current file system.

     If the data was not written to the diskette with relative paths, then the **tar** command attempts to extract the data from the diskette in `rfd0` and place each item at the location specified by the absolute path. For example, if the relative path file is `./myfile`, the file will be written in the current directory.

     If the file was written on the diskette with an absolute path, such as `/u/diane/myfile`, then the file will be copied from the diskette into `/u/diane/myfile`. If the file `/u/diane/myfile` already exists, it will be overwritten.

     Note: An *absolute path name* is the full path name of the file. It begins with a slash (/) immediately followed by the root directory and contains all the directories leading to the file. The *relative path name* starts from the current directory and is the same as its base name if the file is in the **$HOME** directory. If the file is in a subdirectory of **$HOME**, the relative path name is the path from the working directory to the file.

   - If the files were written with the **cpio** command, enter the following command to restore the file:

     ```
     cpio -iv < /dev/rfd0
     ```

     The **cpio** command copies files into and out of storage and moves directory trees.

     Note that absolute path names will be written to their original locations (not to the current directory).

   - If the files were written with the **dd** command, enter the following command to restore the file:

     ```
     dd if=/dev/rfd0 of=myfile
     ```

The **dd** command copies files. It is most useful for reading files that are written in non–UNIX format. In the previous example, **dd** copied the entire diskette to `myfile`.

– If the files were written with the **backup** command, enter the following command to restore the file:

```
restore –xvf/dev/rfd0
```

The **restore** command restores files backed up by the **backup** command.

– Use SMIT to restore the file:

```
smit restore
```

4. After restoring the file, open it and make sure it looks okay. Beware that data in a file restored from a damaged diskette may be incorrect.

Prevent your diskette data from being deleted inadvertently. Put write–protect tabs on the diskettes. Keep diskettes away from magnets, old rotary phones (they contain magnets), and the front or top of CRT screens.

Related Information

'Backup Overview' in *AIX 4.3 System Management Concepts: Operating System and Devices*.

Chapter 7. Network Problems

Network problems could include an inability to contact a host (including name resolution problems), an inability to reach a remote network (a routing problem), or problems with accessing a particular service on an otherwise cooperative host.

Refer to the following procedures to correct your network problems:

Resolve Network Software Problems

Determine if your software is the source of your network problems by:

Resolving the Host Name into an IP Address

1. Find out if you are using the **/etc/resolv.conf** file to resolve host names.

   If other machines on your network have **/etc/resolv.conf** files and yours does not, you probably should have a **/etc/resolv.conf** file.

   For more information about using the **/etc/resolv.conf** file to resolve host names, refer to ”Understanding Domain Name Resolution” in *AIX Communications Programming Concepts*.

2. Find out if you are using an NIS server to resolve host names. For example:

   ```
   ps -ef | grep yp
   ```

   Look for the **ypserv** and **ypbind** processes. If found, you are probably using a network information server (NIS) server to resolve host names.

   You can also enter the **ypwhich** command to display the name of the NIS server for the local machine:

   ```
   ypwhich
   ```

   If you get the message `the domainname has not been set on this machine` , then you are probably not using NIS to resolve host names.

3. Find out if you are using the **/etc/hosts** file to resolve host names. For example:

   ```
   ls /etc/hosts
   ```

   If you can find the **/etc/hosts** file, you are using it to resolve host names. Check your **/etc/hosts** file to see if the host name is listed and not commented out.

Finding Your /etc/resolv.conf File

1. If you know you are using a **/etc/resolv.conf** file to resolve host names, make sure the file exists. For example:

   ```
   ls /etc/resolv.conf
   ```

2. If your **/etc/resolv.conf** is missing and you attempt to connect to a host, you get the following message:

   ```
   telnet: Unknown host SystemName
   ```

3. If your **/etc/resolv.conf** is missing and you attempt to send an echo request with the **ping** command, you get the following message:

   ```
   ping: HostName SystemName NOT FOUND
   ```

   The **ping** command sends an echo request to a network host. If the host is operational and on the network, it responds to the echo. By default, the **ping** command will continue to send echo requests to the display until you send an interrupt (Ctrl–C).

Correcting Your /etc/resolv.conf File

If you know you are using a **/etc/resolv.conf** file to resolve host names and the **/etc/resolv.conf** file exists, make sure it contains the correct host name and IP address.

If you have a bad **/etc/resolv.conf** file, your **ping** command hangs indefinitely when you send an echo request with the **ping** command.

Note: The **/etc/resolv.conf** file contains the IP address, or addresses, of the nameserver(s) you can use and your domain name. Your system administrator should be able to provide you with the correct domain and IP addresses.

Finding a Missing Interface

If you know you can resolve the host name to an address and issue the **telnet** command to another machine, you get the following message if the interface is down or missing:

```
telnet: connect: Network is down
```

1. Check that all interfaces are up and none are missing by using the **netstat** command. For example:

   ```
   netstat –i
   ```

   The **netstat** command displays the status of interfaces on the network.

   The interfaces marked with an asterisk (*) are down. Look for some kind of network interface such as `en0` for Ethernet.

2. If the interface is not up, missing, or detached, bring it up, recreate it, or attach it, respectively, by using the **ifconfig** command. For example, if the Ethernet `en0` is not up, enter:

   ```
   ifconfig en0 inet 129.35.147.121 up
   ```

   If the Ethernet `en0` is missing, enter:

   ```
   ifconfig en0 129.35.147.121 netmask 255.255.255.0 up
   ```

   The **ifconfig** command configures network interface parameters.

Enabling the inetd Daemon

If naming works and the interface is up and configured but you get the following message, do the procedures following the example message:

```
Trying...
telnet: connect: Connection refused.
```

1. Determine if the **inetd** daemon is up. For example:

   ```
   telnet SystemName echo
   ```

   If it echoes back, the **inetd** daemon is up; if it does not echo back, then you must start the **inetd** daemon.

2. Start the **inetd** daemon:

   – If you are using the system resource controller (SRC), enter:

   ```
   startsrc -s inetd
   ```

   Note: If you are using SRC, make sure it is running:

   ```
   ps -eaf | grep srcmst
   ```

   – If you are not using SRC, enter:

   ```
   /etc/inetd
   ```

3. If you have issued the **telnet** command and cannot make a connection, make sure the **inetd** daemon has the **telnet** service enabled by viewing the **inetd.conf** file and looking for the **telnet** service:

   a. Make sure the **telnet** service exists in the file.

   b. Make sure the line describing the **telnet** service is not commented out.

   c. Make sure the path to the **telnet** service is correct.

4. If you make edits to the **inetd.conf** file, refresh the file:

   – If you are using SRC, enter:

   ```
   refresh -s inetd
   ```

   – If you are not using SRC, kill the **inetd** daemon:

   ```
   kill -1 `ps -e | grep /etc/inetd | cut -c1-7`
   ```

Debugging the inetd Daemon

If the **inetd** daemon is up and running correctly and the appropriate service seems to be correct but you still cannot connect, try running the **inetd** daemon processes through a debugger.

1. Stop the **inetd** daemon temporarily:

   ```
   stopsrc -s inetd
   ```

   The **stopsrc** command stops subsystems like the **inetd** daemon.

2. Edit the **syslog.conf** file to add a debugging line at the bottom. For example:

   ```
   vi /etc/syslog.conf
   ```

   a. Add the line " `*.debug /tmp/myfile` " at the bottom of the file and exit.

   b. The file that you specify must exist (`/tmp/myfile` in this example). You can use the **touch** command to make your file exists.

3. Refresh the file:

   – If you are using SRC, enter:

   ```
   refresh -s syslogd
   ```

   – If you are not using SRC, kill the **syslogd** daemon:

   ```
   kill -1 `ps -e | grep /etc/syslogd | cut -c1-7`
   ```

4. Start the **inetd** daemon backup with debugging enabled:

   ```
   startsrc -s inetd -a "-d"
   ```

   The **–d** flag enables debugging.

5. Try to make a connection to log errors in the `/tmp/myfile` debugging file. For example:

   ```
   tn bastet
   Trying...
   connected to bastet
   login:>
   Connection closed
   ```

6. See if anything shows up as a problem in the debugging file. For example:

   ```
   tail -f /tmp/myfile
   ```

Debugging the telnetd Daemon

If the **inetd** daemon could execute the **telnet** service but you still cannot connect using the **telnet** command, there may be something wrong with the **telnet** interface.

1. Verify that **telnet** is using the correct terminal type.

   a. Check the **$TERM** variable on your machine:

   ```
   echo $TERM
   ```

   b. Log in to the machine to which you are trying to attach and check the **$TERM** variable:

   ```
   echo $TERM
   ```

2. Use the **telnet** interface's debugging capabilities by entering the **telnet** command without flags.

   ```
   telnet
   tn>
   ```

   a. Enter `open host` where *host* is the name of the machine.

   b. Enter Ctrl–T to get to the `tn%gt;` prompt.

   c. At the `tn>` prompt, enter `debug` for debugging mode.

3. Try to connect to another machine using the **telnet** interface:

   ```
   telnet bastet
   Trying...
   Connected to bastet
   Escape character is '^T'.
   ```

   Watch the display as the various commands scroll up the screen. For example:

   ```
   SENT do ECHO
   SENT do SUPPRESS GO AHEAD
   SENT will TERMINAL TYPE (reply)
   SENT do SUPPORT SAK
   SENT will SUPPORT SAK (reply)
   RCVD do TERMINAL TYPE (don't reply)
   RCVD will ECHO (don't reply)
   RCVD will SUPPRESS GO AHEAD (don't reply)
   RCVD wont SUPPORT SAK (reply)
   SENT dont SUPPORT SAK (reply)
   RCVD do SUPPORT SAK (don't reply)
   SENT suboption TELOPT_NAWS Width 80, Height 25
   RCVD suboption TELOPT_TTYPE SEND
   RCVD suboption TELOPT_TTYPE aixterm
   ...
   ```

4. Check **/etc/termcap** or **/usr/lib/terminfo** for the `aixterm` definition. For example:

   ```
   ls -a /usr/lib/terminfo
   ```

5. If the `aixterm` definition is missing, add it by building the **ibm.ti** file. For example:

   ```
   tic ibm.ti
   ```

   The **tic** command is a terminal information compiler.

Resolve Network Hardware Problems

Resolve network hardware problems by looking for any impedance variations along the cable with a time domain reflectometer (TDR), including:

- An open cable with no terminator
- A shorted cable or bad transceiver tap
- A healthy transceiver tap
- A healthy connector and barrel adapter
- A sharp bend, crunch, or kink in the cable

1. Analyze the packets and data coming across the cable with a network analyzer.
2. If you are using Token Ring, make sure that it is set to the right ring speed (4MB to 16MB). To do this, enter the **smit tcpip** command:

   ```
   smit tcpip
   ```

   You will access the TCP/IP submenu directly.

   From the TCP/IP submenu, choose the **Minimum Configuration & Startup** option, choose the **Token Ring Network Interface** option (such as **tr0**), and look at the RING Speed field.
3. If you are using Ethernet, make sure it is connected correctly (bnc or bix). To do this, enter the **smit devices** command:

   ```
   smit devices
   ```

   You will access the Devices submenu directly.

   From the Devices submenu, choose the **List Devices** option, choose the **Show Characteristics of a Supported Device** option, choose **adapter** in the Device Class list, choose an interface (such as **mca**) in the Device Interface list, choose **ethernet** in the Device Type list, and look at the Adapter CONNECTOR field.

Resolve Communication Adapter Problems

Determine if your communication adapter is the source of your network problems by:

- Checking for Software Problems , on page 7-8
- Checking for Hardware Problems , on page 7-9

Checking for Software Problems

1. Use the **netstat** command to make sure there is a complete route from your machine to the target machine. For example:

   ```
   netstat -r
   ```

   The **netstat** command displays the status and various statistics of the network.

   The error `no route to remote host available` may indicate that there is not a complete route from your machine to the target or a machine on the route is down.

2. Check for input and output errors. For example:

   ```
   netstat -i
   ```

   Input and output errors can be caused by bad cables or by bad packets sent from other machines to your machine. Input errors can also indicate that a packet was sent to an unknown protocol.

3. Check if your machine needs more buffers allocated to networking. For example:

   ```
   netstat -m
   ```

   If a large amount of memory has been denied in the statistics listing, consider reconfiguring the number of memory buffers allocated to networking. You can find more information about tuning these types of parameters in *AIX Performance Tuning Guide*.

4. Check for dropped connections. For example:

   ```
   netstat -s
   ```

   Look at the statistics on protocol layers. Dropped connections could indicate a problem with the network or with disconnected cables.

5. Use the **arp** command to make sure you have the complete IP address for the target machine. For example:

   ```
   arp -a
   ```

   The **arp** command looks for the physical adapter address. This command might show an incomplete address. For example:

   ```
   ? (192.100.61.210) at (incomplete)
   ```

This could be due to an unplugged machine, a stray address with no machine at that particular address, or a hardware problem (such as a machine that connects and receives packets but is not able to send packets back).

6. Look for errors on the adapter card. For example:

```
netstat -v
```

The **netstat –v** command shows statistics for the Ethernet, Token Ring, X.25, and 802.3 adapter device drivers. The command also shows network and error logging data for all device drivers active on an interface including: `No Mbufs Errors` , `No Mbuf Extension Errors` , and `Packets Transmitted and Adapter Errors Detected` .

Checking for Hardware Problems

To effectively test a network adapter, the adapter must not be figured on the network. Testing of the network adapter should be done from a standalone mode or by unconfiguring the adapter. Run diagnostics in either of two ways to verify that the adapter card is good:

- Use the **diag** command to perform hardware problem determination:

  Note: You must have root user authority to run the **diag** command.

  ```
  diag
  ```

- Using the **smit diag** command, choose the **Current Shell Diagnostics** option, and then select the adapter from the Diagnostic Selection menu using the Function Selection menu:

  ```
  smit diag
  ```

Chapter 8. Printing Problems

Printing problems occur when the /**var** file system is full. This usually happens when print jobs sent to the print queue begin to back up for some reason, causing the spooling directory within the file system to grow too large. The spooling directories usually affected are /**var**/**spool**/**lpd** and /**var**/**spool**/**qdaemon**.

The print queue can back up if the queue daemon has stopped functioning, the printer has gone down or has been turned off, or a large print job sent to the printer has occupied all resources. The /**var** file system can also fill up if other directories in the file system besides the spooling directory grow too large.

When the /**var** file system is full, you must do one or more of the following:

Reactivate the Queue Daemon

The queue daemon (or **qdaemon**) process tracks print job requests and the printers available to handle these requests. The **qdaemon** maintains queues of outstanding requests and, as devices become available, sends them to the proper device at the proper time. If the **qdaemon** stops functioning, you will experience printing problems and will then need to restart the **qdaemon** using the following procedure.

Note: Some commands may require root user or system group authority.

1. Determine if the **qdaemon** has stopped functioning by entering the following **ps** command:

   ```
   ps -ef | grep qdaemon
   ```

   If you do not see a process called **/var/sbin/qdaemon**, **qdaemon**, or **/etc/qdaemon** running, the **qdaemon** is not running.

2. Restart the **qdaemon** by entering the following **startsrc** command:

   ```
   startsrc -s qdaemon
   ```

   If you are not using the system resource controller (SRC), you can also restart the queue daemon with the **qdaemon** command.

   Let the **qdaemon** print all the jobs in the print queue.

3. Make sure the **lpd** daemon is up and running by entering:

   ```
   startsrc -s lpd
   ```

   The **lpd** daemon provides the remote print server on a network.

Clear a Print Queue Backlog

To clear the queuing system, you must stop **qdaemon** and check if the /**var** file system is full. If so, use the following procedure to clear the queue directories and restart the **qdaemon**.

1. Make sure you are logged in as root.
2. If possible, let all current print jobs finish printing or cancel them. To cancel a print job, issue the **lpstat** command to get the print job number. Then use the **enq** command to cancel each job:

   ```
   enq -x JobNumber
   ```

   The **lpstat** command displays information about the current status of the line printer. The **enq** command enqueues a file.
3. Issue the following command to stop **qdaemon**:

   ```
   stopsrc -s qdaemon
   ```

4. Issue the following commands to verify that **qdaemon** did not fork other processes:

   ```
   ps -ef | grep qdaemon

   ps -ef | grep pio
   ```

   The **ps** command shows the current status of processes. The **grep** command searches a file for a pattern.

   If you get one line back from each of the above **grep** commands, skip step 5 and go to step 6. If you get more than one line, go to step 5.
5. If other qdaemons or pios were returned by the **ps –ef** command, kill these processes by issuing the following command with each process ID:

   ```
   kill -9 pid
   ```

   The following example shows a qdaemon returned by **ps –ef**. The process ID is 3357.

   ```
   root 3357 2288 0 13:32:21 - 0:04 dtterm
   ```

   To kill this process ID, enter `kill -9 3357` at the command line.
6. Perform this step only if it is necessary to save the current print jobs from being deleted. Otherwise, proceed to step 7.

   If your print job is queued in one of the following directories, make a copy of it, and place it in /**tmp**; you can print it when the queuing system is running again.

   Note: In these directories, the files will have unfamiliar system names.

   ```
   /var/spool/qdaemon
   /var/spool/lpd
   ```

7. If the /**var** file system gets too full, you may experience problems with **qdaemon** or the spooler. Large print jobs may fail, or **00root** files with zero lengths may appear in your **qdir** directory. Rebooting the system in this case may not clear out the files or restart **qdaemon**.

   Enter the **df** command and look in the `%used` column for `/var` to see if the file system is too full. Free space in the file system as necessary.

The **df** command displays information about total space and available space on a file system.

8. Change the directory as follows:

   ```
   cd /var/spool/lpd/qdir
   ```

9. Issue a **pwd** command to verify that you are in the proper directory. Then, remove all files in this directory using the **del** command:

   ```
   del *
   ```

   The **pwd** command writes to standard output the full path name of your current directory (from the root directory). The **del** command removes the entries for the specified file or files from a directory.

10. Change the directory again:

    ```
    cd /var/spool/lpd/stat
    ```

11. Issue a **pwd** command to verify that you are in the proper directory. Then, remove all files in this directory:

    ```
    del *
    ```

12. Change the directory again:

    ```
    cd /var/spool/qdaemon
    ```

13. Issue a **pwd** command to verify that you are in the proper directory. Then, remove all files in this directory:

    ```
    del *
    ```

14. Follow this step if you are having trouble with the remote queue or **lpd**. Change the directory:

    ```
    cd /var/spool/lpd
    ```

    Issue a **pwd** to verify that you are in the proper directory. Then, remove all files in this directory using the **del** command:

    ```
    del *
    ```

    Note: The **del** command will not remove the subdirectories.

15. Start **qdaemon**:

    ```
    startsrc -s qdaemon
    ```

    The queuing system should start normally. If some queues are still down, bring them up by entering:

    ```
    enable QueueName
    ```

Reallocate Printer Resources

Use the following procedures to avoid having one print job use all of the printer resources.

Note: Some commands may require root user or system group authority.

1. Determine if a print job is using all resources in one of two ways:
   - Use the following **lpq** command:

     ```
     lpq
     ```

     The **lpq** command, when entered without flags, reports the status of the default queue.
   - Use the following **enq** command:

     ```
     enq -q
     ```

     The **enq** command enqueues a file to a shared resource, typically a printer (that is, it puts files into a queue for a particular resource). The **–q** flag displays the status of the default queue.
2. Use one of the following commands to remove the job from the print queue (you must have root user authority to cancel jobs other than your own):
   - Use the following **enq** command.

     ```
     enq -x 21
     ```

     In this example, the **enq** command uses the **–x** flag to cancel job number `21` .
   - Use the following **lprm** command:

     ```
     lprm -P lp0 42
     ```

     In this example, the **lprm** command removes job number `42` from the `lp0` printer queue, named with the **–P** flag. You can also remove jobs for a specific user by naming the user on the command line.
   - Use the following **qadm** command:

     ```
     qadm -X lp0
     ```

     In this example, the **qadm** command uses the **–x** flag to cancel all jobs on the `lp0` printer.
   - Use the following SMIT fast path for the **qcan** command:

     ```
     smit qcan
     ```

     In this example, you can choose the **By Print Queue** option to cancel either all of a particular user's jobs or all jobs on a particular printer.
3. Tell the sender of the print job to first divide it into smaller pieces by using the following **split** command, and then send the file as a series of jobs:

   ```
   split -50 bigfile
   ```

   The **split** command reads the specified file and writes it into segments to a set of output files. In the previous example, `bigfile` is split into 50–line segments named `bigfileaa` , `bigfileab` , `bigfileac` , and so forth.

Delete Unnecessary Directory Files

Use the following procedures to clean out unnecessary files stored in the spooling directory.

Note: Some commands require root user or system group authority.

1. Determine if there are unnecessary files stored in the spooling directory by entering the following **du** command:

   ```
   du –rs /var/spool
   ```

   The **du** command summarizes disk usage. The **–s** flag instructs the **du** command to display only the total disk usage of the /**var**/**spool** directory and the files it contains. The **–r** flag tells the **du** command to display an error message if it cannot read a file or directory.

2. Delete or move files in a full directory by doing either of the following:

   – Delete any extraneous files. For example:

   ```
   rm extrafile
   ```

   – Move files that are a few hours old to a safe temporary directory. For example:

   ```
   mv extrafile /u/spoolhold
   ```

   Note: You must have root user authority to remove or move files other than your own.

3. Prevent users from storing files in your spooling directories by doing the following:

   – Set permissions on the spooling directory using the **chmod** command. Change the directory to exclude general users. For example:

   ```
   chmod go–rw /var/spool/lp0
   ```

   – Create a **cron** job to clean out the directory (you must have root user authority). Edit the **crontab** file. For example, you might add the following line to your **crontab** file:

   ```
   find /spool –mtime +7 –a –exec rm –f
   ```

   This line removes any file in the /**var**/**spool** directory one week after the last modification.

   For more information about creating **cron** jobs and using the **crontab** command, see ”How to Set Up an Accounting System” in *AIX 4.3 System Management Guide: Operating System and Devices*.

   – Establish policy for the whole user group.

   For example, use e–mail to make a general announcement or add to the message of the day (/**etc**/**motd**) asking users to clean up their files. To change the message of the day, see ”How to Change the Message of the Day” in *AIX 4.3 System Management Guide: Operating System and Devices*.

   Create a script to identify all users whose disk holdings are above a certain threshold and send them e–mail requesting that they clean up their files.

- Provide an alternate way to store files, such as a tape drive in a public area, so users can archive infrequently used files.

4. As a last resort, mount more space to the spool directory. There are two ways to do this:
   - Use the **mount** command, which makes a file system available for use at a specified location. For example:

     ```
     mount /var/spool morespool
     ```

   - Use the **smit mount** command, choose the **Mount a File System** option, and specify the file system name and attributes.

Related Information

For more information about mounting file systems, see ”How to Mount or Unmount a File System” in *AIX 4.3 System Management Guide: Operating System and Devices*.

For more information about printers and printing problems, see ”Printer Overview for System Management” in *AIX Guide to Printers and Printing*.

Chapter 9. Hardware Diagnostics

Online and Standalone Diagnostics are used to resolve problems with your hardware. Online Diagnostics are optionally installed with the AIX operating system and can be run in concurrent or service mode. Standalone Diagnostics are stored on removable media and are run in service mode.

Some system types do not have an operator panel display or a mode switch, and some system types do not support Online and Standalone Diagnostics. ???If needed, refer to your system guides to determine what your system supports and how to load the diagnostics in service mode. If your system does not support Online and Standalone Diagnostics, then refer to the system guides to do diagnostics.

This chapter discusses the following :

Consider the following before running this procedure:

- The diagnostic can use either direct–attached displays or a tty terminal attached to the S1 port.

  Note: See the operator guide for your type tty terminal to find the key sequences you need when responding to the diagnostic.

- ISA adapters that are plugged in cannot be detected by the system. When running Standalone Diagnostics, the ISA adapter Configuration Service Aid allows the identification and configuration of ISA adapters. ISA adapters must be identified and configured before they can be tested.
- The DIAGNOSTIC OPERATING INSTRUCTIONS screen does not appear on a console display attached to a **5085** or **5086**. It appears on the console display attached to the system.
- If the procedure instructs you to stop the operating system and you are operating a system that is attached to one or more other systems, consider how actions you take with one system can affect the attached systems. Consult with the operator of the attached system whenever you think your operations might affect another system.

  The following actions can affect or be affected by the operation of an attached system: starting and stopping communication with the other system, running diagnostics on the system, using wrap plugs with a diagnostic program, and analyzing error log information.

  Isolate a system unit from any attached systems before stopping the operating system or running diagnostic programs. Some system cabling changes (such as installing wrap plugs or removing a device from the configuration) may require action by the operator of the attached system before you make the cabling changes on your system.

Selecting and Running Diagnostics

It is important that you select and run the diagnostic tests that meet your needs. This can be achieved by following these steps:

1. Check Your Operating System
2. Choose the Right Diagnostics
3. Run Diagnostic Routines
4. Choose a Specific Diagnostic Test

1.

Check Your Operating System, on page 0If your operating system will accept commands, go to Choosethe Right Diagnostics., on page 0 If your system is not responding to commands, first, try to execute the **shutdown** command to stop the operating system. If this does not work, switch off your system's power and go to Loading the Online Diagnostics in Service Mode, on page 9-3.

2.

Choose the Right Diagnostics, on page 0 Even though concurrent diagnostics can test many resources when your operating system is running and users are accessing the system, using the online diagnostics in service mode can isolate and test problems more accurately.

If you want to run the online diagnostics in service mode, then use the **shutdown** command to stop the operating system, switch off your system's power, and go to Loading the Online Diagnostics in Service Mode., on page 9-3 Otherwise, go to Running the Online Diagnostics in Concurrent Mode., on page 0

Running the Online Diagnostics in Concurrent Mode Run the concurrent diagnostic program by following these steps:

a. Enter the **diag**command to start the diagnostics.

Note: You must either have root user authority or be a user with administrative role 'RunDiagnostics' to run diagnostics. If you are not a root user, you must also have 'system' as primary group and a group set that includes 'shutdown'. Group 'shutdown' is necessary to perform shutdown and reboot operations that are required for certain diagnostic operations. Users in the Run Diagnostics role can change the system configuration, update microcode etc. It is important that users in this role understand the responsibility it requires.

To setup a non–root user, Customer Engineer (CE) or Service Support Representative (SSR) who can run diagnostics, create a unique user using System Management Interface Tool (SMIT). The primary group of this user must be 'system'. This user must also have role 'RunDiagnostics' and a group set that includes 'shutdown'.

b. Wait for the Diagnostic Operating Instructions screen to appear.

c. The diagnostics programs should display the Diagnostic Operating Instructions screen.

If the appearance of the Diagnostic Operating Instructions screen is distorted or unreadable, you may have a problem with the display console. Go to "Identifying Your Console Display" for directions on correcting display problems. If this section does not help you resolve your display problem, then record and report SRN (Service Request Number) 111–121 to your service representative.

d. Press the Enter key to continue. The Function Selection screen should be displayed.

If the Function Selection screen does not appear after you press the Enter key, you might have a problem with your keyboard. If you have a tty, you may need to press a key other than the Enter key. Go to Identifying Your Keyboard, on page 9-11 to continue.

Loading the Online Diagnostics in Service Mode

Note: If you select the Online Diagnostic in Service Mode method, you may not be able to test either your system disk or your SCSI adapter. If this occurs, go to Loading the Standalone Diagnostics in Service Mode, on page 9-3 for a more comprehensive test.

Load the Online Diagnostics by following these steps:

a. Stop your system with the **shutdown**command.

b. Switch off your system's power.

c. If your system has a mode switch, set it to the Service position.

d. Switch on the power for all attached devices.

e. Switch on your system's power.

f. If your system does NOT have a mode switch, and the first device icon or mnemonic displays, press the F6 key on the direct attached keyboard or 6 on the tty keyboard. Enter the password if the password icon or prompt appears.

g. If the Maintenance menu is displayed, select **System Boot**—>**Boot From List**.

h. Follow any instructions to select a console.

i. The diagnostics programs should display the Diagnostic Operating Instructions screen.

   – If the system appears to stop before the Diagnostic Operating Instructions screen screen is displayed, go to Interpret Values in the Operator Display Panel.

   – If the appearance of the Diagnostic Operating Instructions screen is distorted or unreadable, you may have a problem with the display console. Go to "Identifying Your Console Display" for directions on correcting display problems. If this section does not help you resolve your display problem, then record and report SRN (Service Request Number) 111–121 to your service representative.

j. Press the Enter key to continue. The Function Selection screen should be displayed.

   – If the Function Selection screen does not appear after you press the Enter key, you might have a problem with your keyboard. If you have a tty, you may need to press a key other than the Enter key. Go to Identifying Your Keyboard, on page 9-11 to continue.

   – If the system appears to stop before the Function Selection screen is displayed, go to Interpret Values in the Operator Panel Display.

k. Go to Run Diagnostic Routines for information on continuing your diagnostic tests.

Loading the Standalone Diagnostics in Service Mode

Note: If you are using a tty terminal to run standalone diagnostics, check for the following conditions before loading the diagnostics:

– The tty terminal must be attached to the S1 port on your system.

– The attributes for the terminal must be set to match the defaults of the diagnostics.

For more information about loading diagnostics with a tty, see your system operator's guide. It is important that you set your terminal attributes to work with the diagnostics.

Return to this section when you finish checking the attributes. Record any settings that are changed.

Load the diagnostics by doing the following:

a. Stop your system with the **shutdown** command.

b. Switch off your system's power.

c. If your system has a mode switch, set it to the **Service** position.

d. Switch on the power for all attached devices.

e. Switch on your system's power.

f. Insert the Standalone Diagnostics media in the proper drive.

g. If your system does NOT have a mode switch, and the first device icon or mnemonic displays, press the F5 key on the direct attached keyboard or 5 on the tty keyboard. Enter the password if the password icon or prompt appears.

h. If the Maintenance Menu is displayed, select **System Boot**—>**Boot From List**, then continue with this step.

i. Respond to screen messages, followed by the Enter key. The diagnostics program defines your system console and shows the Diagnostic Operating Instructions screen.

j. Press the Enter key to continue. The diagnostics program shows the Function Selection screen.

k. Select **Initialize Terminal** from the menu on the Function Selection screen.

l. Enter your terminal type. The diagnostics program initializes your system environment and shows the Function Selection screen again.

m. Go to Run Diagnostic Routines, on page 9-4 for information about continuing with your diagnostic tests.

3.

Run Diagnostic Routines

Concurrent Diagnostics or Online Diagnostics in Service Mode

Select Diagnostic Routines from the menu on the Function Selection screen.

Use the procedures listed in the table below, according to the system response.

System Response	User Action
Diagnostic Mode Selection menu is displayed	Select Problem Determination from the menu and press Enter. Go to Choose a Specific Diagnostic Test.
DIAGNOSTIC SELECTION menu is displayed	If the DIAGNOSTIC SELECTION menu is displayed when you select **Problem Determination** , go to Choose a Specific Diagnostic Test, on page 9-6.

System Response	User Action
The MISSING RESOURCE menu or the NEW RESOURCE menu is displayed.	Follow the displayed instructions until either the DIAGNOSTIC SELECTION menu or an SRN is displayed. **Notes:** – Run any supplemental media which may have been supplied with the adapter or device, and then select the **DIAGNOSTICS ROUTINES** option. – If you are running Standalone Diagnostics, be sure that all adapters and SCSI devices are listed on the NEW RESOURCE menu. – Resources attached to serial and parallel ports may not appear in the NEW RESOURCE menu. ISA adapters cannot be detected by the system. The ISA Adapter Configuration Service Aid in Standalone Diagnostics allows the identification and configuration of ISA adapters. If the DIAGNOSTIC SELECTION menu is displayed appears, go to Choose a Specific Diagnostic Test, on page 9-6 . If you get an SRN, record and report it to your service representative. If you get a code in the operator panel display, go to Interpret Values in the Operator Panel Display, on page 9-12 .
The message `The system will now continue the boot process` is displayed continuously on the system unit's console.	Shutdown the system, and report the problem to your service representative.
The diagnostics begin testing a resource. **Note:** If the Problem Determination Option was selected from the DIAGNOSTIC MODE SELECTION menu, and if a recent error has been logged in the error log, the diagnostics automatically begins testing the resource.	Follow the displayed instructions. If the No Trouble Found screen is displayed, press Enter. If another resource is tested, repeat this step. If the DIAGNOSTIC SELECTION menu is displayed appears, go to Choose a Specific Diagnostic Test, on page 9-6 . If you get an SRN, record and report it to your service representative. If you get a code in the operator panel display, go to Interpret Values in the Operator Panel Display, on page 9-12 .
The system stopped with an `888` message, a 3–digit code, or a 4–digit code displayed in the operator panel display.	Go to Interpret Values in the Operator Panel Display, on page 9-12 .

Standalone Diagnostics in Service Mode

Select Diagnostic Routines from the menu on the Function Selection screen.

The first time you select either Diagnostics Routines or Advanced Diagnostics Routines, the diagnostics lists all resources that were detected with the exception of ISA adapters and those resources attached to serial and parallel ports. The ISA Adapter Configuration Service Aid task allows the identification and configuration of an ISA adapter. All ISA adapters must be identified and configured before they can be tested.

Follow the instructions displayed on the screen. Be sure and check that the resources you are having problems with are listed. , on page 9-6, on page 9-6

System Response	User Action
Diagnostic Mode Selection	Select Problem Determination from the menu and press Enter. Go to Choose a Specific Diagnostic Test.
Diagnostic Selection	If the Diagnostic Selection screen appears when you select **Problem Determination** , go to Choose a Specific Diagnostic Test
SRN Displayed	Record and report the SRN to your service representative
New Resource Menu	If the New Resource screen appears follow the instructions on your screen. ISA adapters cannot be detected by the system. The ISA adapter Configuration Service Aid tasks allow the identification and configuration of ISA adapters. ISA adapters must be identified and configured before they can be tested. **Note:** Devices attached to the built–in serial and parallel ports do not appear on the New Resource menu. If the Diagnostic Selection screen appears, go to Choose a Specific Diagnostic Test. If you get an SRN, ecord and report the SRN to your service representative.

4.

Choose a Specific Diagnostic Test

Select and run the diagnostic tests on the hardware you are having problems with.

Alternatively, you can run a full system checkout to check all of your configured hardware. Select these diagnostics from the menu on the Diagnostic Selection screen.

If the Diagnostic Mode Selection menu appears, select Problem Determination from the menu and press Enter.

The table below lists the various ways your system can respond when you choose a specific test.

System Response	User Action
A Corrective Action Appears	A corrective action is displayed for problems that do not require a part replacement. Follow the procedure that is displayed for the corrective action.
SRN Displayed	Record and report the SRN to your service representative.
System Stops	If your system stops with a code showing in the operator panel display, go to Interpret Values in the Operator Panel Display, on page 9-12. Choose your code from the list, and read how to interpret and act on the code.

System Response	User Action
Test Completion Notice Appears Before You Have Finished Testing	– If your console display shows the Testing Complete screen with the `No trouble was found message` and you have not tested all of your resources, press enter. – If the Diagnostic Selection screen appears, go to Choose a Specific Diagnostic Test, on page 9-6. – If you do not get an SRN, the diagnostics program did not find a hardware problem. Check your configuration, and then contact the IBM software support center if you still have a problem. If you are attached to another system, refer to hardware diagnostics considerations , on page 9-1. ISA adapters cannot be detected by the system. When running Standalone Diagnostics, the ISA adapter Configuration Service Aid allows the identification and configuration of ISA adapters. ISA adapters must be identified and configured before they can be tested. **Note:** If dials and lighted programmable function keyboards (LPFKs) are attached to built–in serial ports and you are running Online Diagnostics, the dials and LPFKs only appear on the selection screen if they have been configured by the user. Use the dials and LPFKs configuration service aid or SMIT to configure these devices. If the dials and LPFKs are attached to built–in serial ports, you must configure them using the dials and LPFKs configuration service aid before they can be tested by Standalone Diagnostics.
Test Completion Notice Appears After You Have Finished Testing	– If your console display shows the Testing Complete screen with the `No trouble was found` message, and you have tested all of your resources, then the diagnostics program did not find a hardware problem. – If you still have a problem, check your configuration, and then contact the IBM software support center. If you are attached to another system, refer to the multiple systems guidelines in the chapter on using diagnostics of your system unit operator guide.

Choose Test Options

Select the **Diagnostic Routines** option from the menu on the FUNCTION SELECTION screen. The DIAGNOSTIC MODE SELECTION screen appears.

Select the **Problem Determination** option from the menu on the DIAGNOSTIC MODE SELECTION screen.

Begin your diagnostic test by:

- Responding to Test Messages, on page 9-9
- Choosing a Specific Diagnostic Test, on page 9-9

Responding to Test Messages

Determine which test message applies to your system, and refer to the appropriate procedures:

- Diagnostic Selection Menu Appears, on page 9-9
- Missing Resource Menu Appears, on page 9-9
- New Resource Menu Appears, on page 9-9

Diagnostic Selection Menu Appears

If the DIAGNOSTIC SELECTION screen appears when you select the **Problem Determination** option, go to ”Choosing a Specific Diagnostic Test”, on page 9-9.

Missing Resource Menu Appears

If the MISSING RESOURCE screen appears follow the displayed instructions until either the DIAGNOSTIC SELECTION screen or an SRN is displayed.

If the DIAGNOSTIC SELECTION screen appears, go to ”Choosing a Specific Diagnostic Test”, on page 9-9.

If you get an SRN, record and report it to your service representative.

If you get a code in the operator panel display, go to ”Interpret Values in the Operator Panel Display”, on page 9-12.

New Resource Menu Appears

If the NEW RESOURCE screen appears follow the instructions on your screen.

ISA adapters cannot be detected by the system. When running Standalone Diagnostics, the ISA adapter Configuration Service Aid allows the identification and configuration of ISA adapters. ISA adapters must be identified and configured before they can be tested.

> **Note:** Devices attached to the S1 and S2 serial ports do not appear on the NEW RESOURCE menu.

If the DIAGNOSTIC SELECTION screen appears, go to ”Choosing a Specific Diagnostic Test”, on page 9-9.

If you get an SRN, record and report it to your service representative.

Choosing a Specific Diagnostic Test

Select and run the diagnostic tests on the hardware you are having problems with.

Alternatively, you can run a full system checkout to check all of your configured hardware. The system checkout is available, however, only when you use Online Diagnostic in Service Mode. Select the option from the menu on the DIAGNOSTIC SELECTION screen.

Your system can respond in one of several ways when you choose a specific test:

- An SRN Appears, on page 9-10

- Your System Stops, on page 9-10
- A Test Completion Notice Appears before You Have Finished Testing, on page 9-10
- A Test Completion Notice Appears after You Have Finished Testing, on page 9-10

An SRN Appears

If your console display shows an SRN when you run a specific test, record and report the SRN to your service representative.

Your System Stops

If your system stops with a code showing in the operator panel display, go to ”Interpret Values in the Operator Panel Display”, on page 9-12. Choose your code from the list, and read how to interpret and act on the code. Otherwise report SRN 111–905.

A Test Completion Notice Appears before You Have Finished Testing

If your console display shows the TESTING COMPLETE screen with the `No trouble was found message` and you have not tested all of your resources, press enter.

If the DIAGNOSTIC SELECTION screen appears, go to ”Choosing a Specific Diagnostic Test”, on page 9-9.

If the Missing Resource Menu appears, go to ”Missing Resource Menu Appears”, on page 9-9

If the New Resource Menu appears, go to ”New Resource Menu Appears”, on page 9-9

If you get an SRN, record and report it to your service representative.

If you do not get an SRN, the diagnostics program did not find a hardware problem. Check your configuration, and then contact the IBM software support center if you still have a problem. If you are attached to another system, refer to hardware diagnostics considerations , on page 9-1.

ISA adapters cannot be detected by the system. When running Standalone Diagnostics, the ISA adapter Configuration Service Aid allows the identification and configuration of ISA adapters. ISA adapters must be identified and configured before they can be tested.

> **Note:** If dials and lighted programmable function keyboards (LPFKs) are attached to serial ports S1 or S2 and you are running Online Diagnostics, the dials and LPFKs only appear on the selection screen if they have been configured by the user. Use the dials and LPFKs configuration service aid or SMIT to configure these devices.

If the dials and LPFKs are attached to serial ports S1 or S2, you must configure them using the dials and LPFKs configuration service aid before they can be tested by Standalone Diagnostics.

A Test Completion Notice Appears after You Have Finished Testing

If your console display shows the TESTING COMPLETE screen with the `No trouble was found` message and you have tested all of your resources, then the diagnostics program did not find a hardware problem.

If you still have a problem, check your configuration, and then contact the IBM software support center. If you are attached to another system, refer to the multiple systems guidelines in the chapter on using diagnostics of your system unit operator guide.

Identify Your Hardware

Identify your hardware by:

- Identify Your Console Display, on page 9-11
- Identify Your Keyboard, on page 9-11

Identify Your Console Display

Identify your console display in the following list. Follow the prescribed action if you think your console display is preventing you from completing your diagnostics test.

CONSOLE TYPE	ACTION
display device	See your display documentation for problem determination.
tty terminal	See your tty terminal documentation for problem determination.

Identify Your Keyboard

Identify your keyboard in the following list. Report the SRN (Service Request Number) if you think your keyboard is preventing you from completing your diagnostics test.

KEYBOARD TYPE	ACTION
101–key	Identify by the type of Enter key used. The Enter key is within one horizontal row of keys. Report SRN 111–921.
102–key	Identify by the type of Enter key used. The Enter key extends into two horizontal rows of keys. Report SRN 111–922.
Kanji	Identify by the Japanese characters. Report SRN 111–923.
tty terminal	Go to your hardware documentation for problem determination. This instruction applies to all attached terminals.

Interpreting Values in the Operator Panel Display

If the diagnostics program has trouble loading and the system has an operator panel display, a code will be displayed in the operator panel display. Some values are steady, some values alternate between two or more values, and one code flashes on and off.

Choose from the following list for information about interpreting and acting on your code:

DISPLAYED VALUE	DEFINITION
c31 (steady)	When the `c31` code appears, your diagnostics program cannot detect a console display. Follow the instructions on your screen to select a console display. If you do not have a console display, set the mode switch first to Normal and then back to Service to indicate to the diagnostics there is no console display. **Note:** Trouble with either your display or your keyboard can prevent you from selecting a console display. If your screen appearance is bad (distorted or unreadable), go to Identify Your Console Display, on page 9-11 . Follow the instructions for correcting the screen problem. If you cannot make a screen selection because your keyboard seems not to respond, you might have a problem with your keyboard. Go to Identify Your Keyboard, on page 9-11 to find the SRN (Service Request Number) you need to report your problem.
8–digit code	This is an error code, and should be recorded and reported to your service representative.
3–digit code starting with F	This is a three–digit firmware checkpoint. Record and report the code to your service representative.
3–digit code that Does not start With F	This is a three–digit AIX event indicator. Record and report SRN 111–xxx, where xxx is the three–digit value, to your service representative.
4–digit code starting with 8, 9, or A–F	This is a four–digit firmware checkpoint. Record and report the code to your service representative
4–digit code starting with a number from 0–7.	This is a four–digit AIX event indicator. Record and report SRN 111–xxx, where xxxx is the four–digit value, to your service representative

DISPLAYED VALUE	DEFINITION
200 (steady)	When you see a 200 code in the operator panel display, check that the mode switch is set to the Service position. If the mode switch is not in the Service position, set it to Service. Then see Loading the Online Diagnostics in Service Mode, on page 9-3 or Loading the Standalone Diagnostics in Service Mode, on page 9-3 for information about restarting your diagnostic tests. If the mode switch is in the Service position, record and report SRN 111–102 to your service representative.
260 (steady)	**Note:** Record and report SRN 111–101 to your service representative if you are loading Standalone Diagnostics on any machine other than a DPX/20 100 or DPX/20 105. If your machine is a DPX/20 100 or DPX/20 105, press the 1 key to continue the diagnostics. Continue with this section if you are loading Online Diagnostics. When you get a 260 code in the operator panel display, check whether the console display is directly attached to the system unit by means of a graphics adapter. If so, go to Responding to the 260, 261, and 262 Values, on page 9-15 . If the console display is not directly attached to the system, press the 1 key on your keyboard. Observe your system; then, go to Responding to the 260, 261, and 262 Values, on page 9-15 .
261 (steady)	**Note:** Record and report SRN 111–101 to your service representative if you are loading Standalone Diagnostics on any machine other than a DPX/20 100 or DPX/20 105. If your machine is a DPX/20 100 or DPX/20 105, press the 1 key to continue the diagnostic. Continue with this section if you are loading Online Diagnostics. Press the 1 key on your keyboard. Observe your system; then, go to Responding to the 260, 261, and 262 Values, on page 9-15 .

DISPLAYED VALUE	DEFINITION
262 (steady)	**Note:** Record and report SRN 111–101 to your service representative if you are loading Standalone Diagnostics on any machine other than a DPX/20 100 or DPX/20 105. If your machine is a DPX/20 100 or DPX/20 105, press the 1 key to continue the diagnostic. Continue with this section if you are loading Online Diagnostics. Your diagnostic program did not detect a keyboard. If a keyboard is attached to the keyboard connector (K) on the system unit, record and report SRN 101–262 to your service representative. Otherwise, press the 1 key on your keyboard; then, go to Responding to the 260, 261, and 262 Values, on page 9-15 .
888 (flashing)	A flashing 888 indicates that a problem was detected, but could not be displayed on the console. The 888 will be followed by either a 102, 103, or 105. The reset button is used to scroll the message. •A 102 code indicates that a software or hardware check occurred and the system could not continue. A 102 message may be followed by a 103 message. If this occurs, follow the procedure for the 103 message. If the 103 message is not displayed, record and report SRN 111–107 to your service representative. •A 103 message indicates that a SRN follows the 103. The SRN consists of the two sets of digits following the 103 message. Record and report the SRN to your service representative. •A 105 message indicates that an encoded SRN follows the 105. Record and report SRN 111–108 to your service representative.

DISPLAYED VALUE	DEFINITION
888 (flashing) followed by a string of numbers	A flashing 888 indicates that a problem was detected, but could not be displayed on the console. The 888 will be followed by either a 102, 103, or 105. •A 102 code indicates that a software or hardware check occurred and the system could not continue. A 102 message may be followed by a 103 message. If this occurs, follow the procedure for the 103 message. If the 103 message is not displayed, record and report SRN 111–107 to your service representative. •A 103 message indicates that a SRN follows the 103. The SRN consists of the two sets of digits following the 103 message. Record and report the SRN to your service representative. •A 105 message indicates that an encoded SRN follows the 105. Record and report SRN 111–108 to your service representative.
Two or More Values (alternating)	**Note:** Record and report SRN 111–101 to your service representative if you are loading diagnostics from either removable media. Continue with this section if you are loading Online Diagnostics. Your system stopped with two or more values between `221` and `296` alternating in the operator panel display. This code indicates the diagnostics could not be found. If you already tried to load the standalone diagnostics, record and report SRN 111–101.

Responding to the 260, 261, and 262 Values

When you get either a `260`, a `261`, or a `262` code, you can normally press the 1 key on your keyboard and observe your system. Watch for one of the following results, and perform the indicated procedures.

Note: See the descriptions of these three values, and perform any required steps before continuing with this section.

1. If information does not show correctly on the console display (the information is distorted, blurred, or not readable), then go to Identifying Your Console Display, on page 9-11.
2. If your console display shows an SRN, record and report the SRN to your service representative.
3. If your console display shows the `MAIN MENU`, you can either stop this problem determination procedure or load diagnostic programs from other media.

   If you want to stop, record and report SRN 111–103 to your service representative.

   If you want to continue, go to Loading the Standalone Diagnostics in Service Mode, on page 9-3.

Gather Information about Your System Configuration

You can gather information about your system configuration by:

- Using the lscfg Command, on page 9-16
- Using the snap Command, on page 9-16

Using the lscfg Command

Enter the following command to show your system configuration on your screen:

```
lscfg
```

The **lscfg** command shows the name, location, and description of all hardware in your system.

Using the snap Command

Note: You need root user authority to use the **snap** command.

Enter the following command to gather all configuration information in a **tar** file and copy it to diskette:

```
snap -a -o /dev/rfd0
```

The **snap** command in this example uses the **–a** flag to gather all information about your system configuration. The **–o** flag copies the compressed **tar** file to the device you name. `/dev/rfd0` names your disk drive.

Enter the following command to gather all configuration information in a **tar** file and copy it to tape:

```
snap -a -o /dev/rmt0
```

`/dev/rmt0` names your tape drive.

Note: If you intend to use a tape to send a snap image to IBM for software support. The tape must be one of the following formats: **8mm, 2.3 Gb** capacity, **8mm, 5.0 Gb** capacity, or **4mm, 4.0 Gb** capacity. Using other formats will prevent or delay software support from being able to examine the contents.

Periodic Diagnostics and Automatic Error Log Analysis

Periodic Diagnostics and Automatic Error Log Analysis are provided by the diagnostics.

Periodic Diagnostics

Periodic diagnosis of the disk drives and battery are enabled by default. The disk diagnostics will perform disk error log analysis on all disks. The battery diagnostics will test the real time clock and NV–RAM battery.

Periodic Diagnostics are performed in different ways depending on the diagnostic version.

AIX Version 4

Periodic diagnostics in AIX Version 4 is controlled by the Periodic Diagnostic Service Aid. The Periodic Diagnostic Service Aid allows error log analysis to be run on hardware resources once a day. By default, the battery and all disk drives are enabled to run. The battery diagnostic is run at 4:00 a.m. each day, and error log analysis is performed on all the disk drives at 3:00 a.m. each day. Other devices can be added to the Periodic Diagnostic Device list, and error log analysis can be directed to run at different times.

Problems are reported by a message to the system console, and a mail message is sent to all members of the system group. The message contains the SRN.

Running diagnostics in this mode for base system devices is similar to using the **diag –c –d** *device* command. All other devices are invoked with the **–e** flag appended.

Automatic Error Log Analysis (diagela)

Automatic Error Log Analysis (**diagela**) provides the capability to do error log analysis whenever a permanent hardware error is logged. If the **diagela** program is enabled and a permanent hardware resource error is logged, the **diagela** program is started. Automatic Error Log Analysis is enabled by default on all platforms.

The **diagela** program determines whether the error should be analyzed by the diagnostics. If the error should be analyzed, a diagnostic application will be invoked and the error will be analyzed. No testing is done. If the diagnostics determines that the error requires a service action, it sends a message to your console and to all system groups. The message contains the SRN, or a corrective action.

Running diagnostics in this mode is similar to using the **diag –c –e –d** device command.

Notification can also be customized by adding a stanza to the PDiagAtt object class. The following example illustrates how a customer's program can be invoked in place of the normal mail message:

```
PDiagAtt:

DClass = ""
DSClass = ""
DType = ""
attribute = "diag_notify"
value = "/usr/bin/customer_notify_program $1 $2 $3 $4 $5"
rep = "s"
```

If DClass, DSClass, and DType are blank, then the customer_notify_program will apply for ALL devices. Filling in the DClass, DSClass, and DType with specifics will cause the customer_notify_program to be invoked only for that device type.

Once the above stanza is added to the ODM data base, problems will be displayed on the system console and the program specified in the value field of the **diag_notify** predefined attribute will be invoked. The following keywords will be expanded automatically as

arguments to the notify program: $1 the keyword ”diag_notify”
$2 the resource name that reported the problem
$3 the Service Request Number
$4 the device type
$5 the error label from the error log entry

In the case where no diagnostic program is found to analyze the error log entry, or analysis is done but no error was reported, a separate program can be specified to be invoked. This is accomplished by adding a stanza to the **PDiagAtt** object class with an attribute = ”diag_analyze”. The following example illustrates how a customer’s program can be invoked for this condition:

```
PDiagAtt:
DClass = ””
DSClass = ””
DType = ””
attribute = ”diag_analyze”
value = ”/usr/bin/customer_analyzer_program $1 $2 $3 $4 $5”
rep = ”s”
```

If DClass, DSClass, and DType are blank, then the **customer_analyzer_program** will apply for ALL devices. Filling in the **DClass**, **DSClass**, and **DType** with specifics will cause the **customer_analyzer_program** to be invoked only for that device type.

Once the above stanza is added to the ODM data base, the program specified will be invoked if there is no diagnostic program specified for the error, or if analysis was done, but no error found. The following keywords will be expanded automatically as arguments to the analyzer program:

Keyword	Definition
$1	the keyword ”diag_analyze”
$2	The resource name that reported the problem.
$3	Can be one of the following: •The error label from the error log entry if invoked for ELA •The keyword ”PERIODIC” if invoked for Periodic Diagnostics. •The keyword ”REMINDER” if invoked for providing a Diagnostic Reminder.
$4	the device type
$5	Can be either of the following keywords: •”no_trouble_found” if analyzer was run and no trouble was found •”no_analyzer” if analyzer was not available

To activate the Automatic Error Log Analysis feature, log in as root and type the following command:

```
/usr/lpp/diagnostics/bin/diagela ENABLE
```

To disable the Automatic Error Log Analysis feature, log in as root and type the following command:

```
/usr/lpp/diagnostics/bin/diagela DISABLE
```

The **diagela** program can also be enabled and disabled using the Periodic Diagnostic Service Aid.

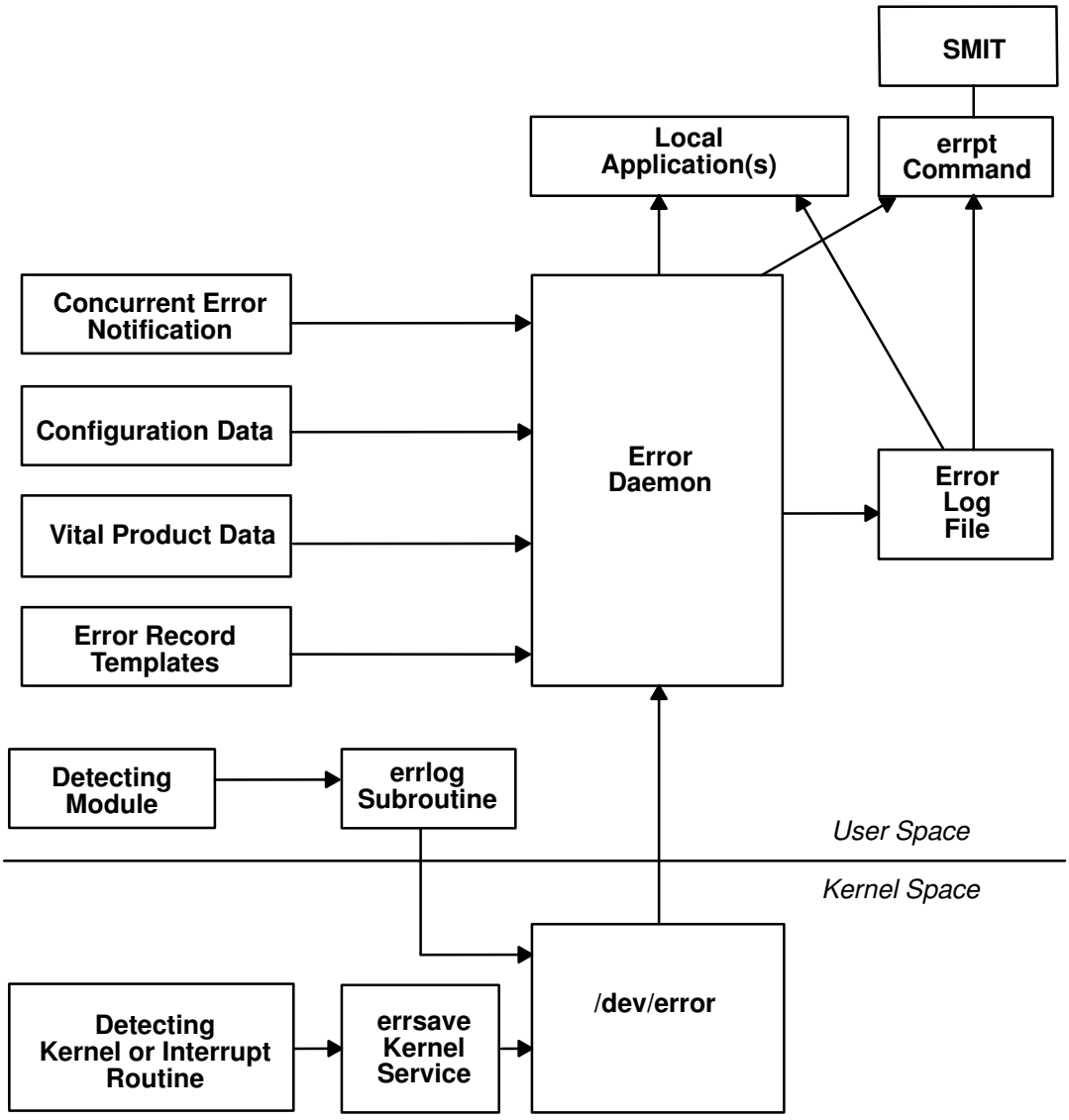
Chapter 10. Error Logging Facility

The error logging facility records hardware and software failures in the error log for information purposes or for fault detection and corrective action.

Refer to the following to use the error logging facility:

In AIX Version 4, some of the error log commands are delivered in an optionally installable package called **bos.sysmgt.serv_aid**. The base system (**bos.rte**) includes the services for logging errors to the error log file. This includes the errlog subroutines, the **errsave** and **errlast** kernel service, the error device driver (/**dev**/**error**), the **error** daemon, and the **errstop** command. The commands required for licensed program installation (**errinstall** and **errupdate**) are also included in **bos.rte**. See 'Software Service Aids Package', on page 1-4 for more information. Also, for information on transferring your system's error log file to a system that has the Software Service Aids package installed, see 'Transferring Your Error Log to Another System', on page 10-4.

The following diagram illustrates the error logging process:



Flow of the Error Logging Facility

Error Logging Overview

The error logging process begins when an operating system module detects an error. The error–detecting segment of code then sends error information to either the **errsave** and **errlast** kernel service or the **errlog** application subroutine, where the information is in turn written to the /**dev**/**error** special file. This process then adds a time stamp to the collected data. The **errdemon** daemon constantly checks the /**dev**/**error** file for new entries, and when new data is written, the daemon conducts a series of operations.

Before an entry is written to the error log, the **errdemon** daemon compares the label sent by the kernel or application code to the contents of the Error Record Template Repository. If the label matches an item in the repository, the daemon collects additional data from other parts of the system.

To create an entry in the error log, the **errdemon** daemon retrieves the appropriate template from the repository, the resource name of the unit that detected the error and detail data. Also, if the error signifies a hardware–related problem and hardware vital product data (VPD) exists, the daemon retrieves the VPD from the Object Data Manager. When you access the error log, either through SMIT or with the **errpt** command, the error log is formatted according to the error template in the error template repository and presented in either a summary or detailed report. Most entries in the error log are attributable to hardware and software problems, but informational messages can also be logged.

The **diag** command uses the error log in part to diagnose hardware problems. To correctly diagnose new system problems, the system deletes hardware–related entries older than 90 days from the error log. The system deletes software–related entries 30 days after they are logged.

Terms to help you use the error logging facility include the following:

error ID	A 32–bit CRC hexadecimal code used to identify a particular failure. Each error record template has a unique error ID.
error label	The mnemonic name for an error ID.
error log	The file that stores instances of errors and failures encountered by the system.
error log entry	A record in the system error log that describes a hardware failure, a software failure, or an operator message. An error log entry contains captured failure data.
error record template	A description of what will be displayed when the error log is formatted for a report, including information on the type and class of the error, probable causes, and recommended actions. Collectively, the templates comprise the Error Record Template Repository.

Managing Error Logging

Error logging is automatically started during system initialization by the **rc.boot** script and is automatically stopped during system shutdown by the **shutdown** script. The error log analysis performed by the diagnostics (**diag** command) analyzes hardware error entries up to 90 days old. If you remove hardware error entries less than 90 days old, you can limit the effectiveness of this error log analysis.

To manage error logging efficiently, see:

Transferring Your Error Log to Another System

The **errclear**, **errdead**, **errlogger**, **errmsg**, and **errpt** commands are part of the optionally installable Software Service Aids package (**bos.sysmgt.serv_aid**). You need the Software Service Aids package to generate reports from the error log or delete entries from the error log. You can install the Software Service Aids package on your system or you can transfer your system's error log file to a system that has the Software Service Aids package installed.

Determine the path to your system's error log file by running the following command:

```
/usr/lib/errdemon -l
```

There are a number of ways to transfer the file to another system. For example, you can copy the file to a remotely mounted file system using the **cp** command; you can copy the file across the network connection using the **rcp**, **ftp**, or **tftp** commands; or you can copy the file to removable media using the **tar** or **backup** command and restore the file onto another system.

You can format reports for an error log copied to your system from another system by using the **–i** flag of the **errpt** command. The **–i** flag allows you to specify the path name of an error log file other than the default. Likewise, you can delete entries from an error log file copied to your system from another system by using the **–i** flag of the **errclear** command.

Configuring Error Logging

You can customize the name and location of the error log file and the size of the internal error buffer to suit your needs.

Listing the Current Settings

To list the current settings, run **/usr/lib/errdemon –l**. The values for the error log file name, error log file size, and buffer size that are currently stored in the error log configuration database display on your screen.

Customizing the Log File Location

To change the filename used for error logging run the **/usr/lib/errdemon –i** *FileName* command. The specified file name is saved in the error log configuration database and the error daemon is immediately restarted.

Customizing the Log File Size

To change the maximum size of the error log file enter:

```
/usr/lib/errdemon -s LogSize
```

The specified log file size limit is saved in the error log configuration database and the error daemon is immediately restarted. If the log file size limit is smaller than the size of the log file currently in use, the current log file is renamed by appending **.old** to the file name and a new log file is created with the specified size limit. The amount of space specified is reserved for the error log file and is not available for use by other files. Therefore, you should be careful not to make the log excessively large. But, if you make the log too small, important information may be overwritten prematurely. When the log file size limit is reached, the file *wraps*, that is, the oldest entries are overwritten by new entries.

Customizing the Buffer Size

To change the size of the error log device driver's internal buffer, enter:

```
/usr/lib/errdemon -B BufferSize
```

The specified buffer size is saved in the error log configuration database and, if it is larger than the buffer size currently in use, the in–memory buffer is immediately increased. If it is smaller than the buffer size currently in use, the new size is put into effect the next time the error daemon is started after the system is rebooted. The buffer cannot be made smaller than the hard–coded default of 8KB. The size you specify is rounded up to the next integral multiple of the memory page size (4KBs). The memory used for the error log device driver's in–memory buffer is not available for use by other processes (the buffer is pinned).

You should be careful not to impact your system's performance by making the buffer excessively large. But, if you make the buffer too small, the buffer may become full if error entries are arriving faster than they are being read from the buffer and put into the log file. When the buffer is full, new entries are discarded until space becomes available in the buffer. When this situation occurs, an error log entry is created to inform you of the problem, and you should correct the problem by enlarging the buffer.

Removing Error Log Entries

Entries are removed from the error log when the root user runs the **errclear** command, when the **errclear** command is automatically invoked by a daily **cron** job, and when the error log file wraps as a result of reaching its maximum size. When the error log file reaches the maximum size specified in the error log configuration database, the oldest entries are overwritten by the newest entries.

Automatic Removal

The system is shipped with a **crontab** file to delete hardware errors older than 90 days and other errors older than 30 days. To display the **crontab** entries for your system, enter:

```
crontab -l Command
```

To change these entries, enter:

```
crontab -e Command
```

See the **crontab** command.

errclear Command

The **errclear** command can be used to selectively remove entries from the error log. The selection criteria you may specify include the error id number, sequence number, error label, resource name, resource class, error class, and error type. You must also specify the age of entries to be removed. The entries that match the selection criteria you specified and are older than the number of days you specified will be removed.

Enabling and Disabling Logging for an Event

You can disable logging or reporting of a particular event by modifying the Log or the Report field of the error template for the event. You can use the **errupdate** command to change the current settings for an event.

Showing Events for Which Logging is Disabled

To list all events for which logging is currently disabled, enter:

```
errpt -t -F Log=0
```

Events for which logging is disabled are not saved in the error log file.

Showing Events for which Reporting is Disabled

To list all events for which reporting is currently disabled, enter:

```
errpt -t -F Report=0
```

Events for which reporting is disabled are saved in the error log file when they occur, but they are not displayed by the **errpt** command.

Changing the Current Setting for an Event

You can use the **errupdate** command to change the current settings for an event. The necessary input to the **errupdate** command can be in a file or from standard input.

In the following example, standard input is used. To disable the reporting of the **ERRLOG_OFF** event (error id number 192AC071), enter the following lines to run the **errupdate** command:

```
errupdate <Enter>
=192AC071: <Enter>
Report=False <Enter>
<Ctrl-D>
<Ctrl-D>
```

Setting Up Error Notification

Refer to ”Error Notification Object Class in ” in *AIX General Programming Concepts : Writing and Debugging Programs*.

Logging Maintenance Activities

The **errlogger** command allows the system administrator to record messages in the error log. Whenever you perform a maintenance activity, such as clearing entries from the error log, replacing hardware, or applying a software fix, it is a good idea to record this activity in the system error log.

Redirecting syslog Messages to Error Log

Some applications use syslog for logging errors and other events. Some administrators find it desirable to be able to list error log messages and syslog messages in a single report. This can be accomplished by redirecting the syslog messages to the error log. You can do this by specifying *errlog* as the destination in the syslog configuration file **(/etc/syslog.conf**). See the **syslogd** daemon for more information.

Directing Error Log Messages to Syslog

You can log error log events in the **syslog** file by using the **logger** command with the concurrent error notification capabilities of error log. For example, to log system messages (syslog), add an errnotify object with the following contents:

```
errnotify:
        en_name = ”syslog1”
        en_persistenceflg = 1
        en_method = ”logger Msg from Error Log: ‘errpt –l $1 |
grep –v ’ERROR_ID TIMESTAMP’ ‘”
```

For example, create a file called /**tmp**/**syslog.add** with these contents, then run the command **odmadd** /**tmp**/**syslog.add** (you must be logged in as root to do this).

For more information about concurrent error notification, see the ”Error Notification Object Class in ” in *AIX General Programming Concepts : Writing and Debugging Programs*.

Error Logging Tasks

Error logging tasks and information to assist you in using the error logging facility include:

Reading an Error Report

To obtain a report of all errors logged in the 24 hours prior to the failure, enter:

```
errpt -a -s mmddhhmmyy | pg
```

where *mmddhhmmyy* represents the month, day, hour, minute, and year 24 hours prior to the failure.

An error log report contains the following information:

Note: Not all errors will generate information for each of the following categories.

LABEL	Predefined name for the event.
ID	Numerical identifier for the event.
Date/Time	Date and time of the event.
Sequence Number	Unique number for the event.
Machine ID	Identification number of your system processor unit.
Node ID	Mnemonic name of your system.
Class	General source of the error. The possible error classes are:

H	Hardware. (When you receive a hardware error, refer to your system operator guide for information about performing diagnostics on the problem device or other piece of equipment. The diagnostics program tests the device and analyze the error log entries related to it to determine the state of the device.)
S	Software.
O	Informational messages.
U	Undetermined (for example, a network).

Type Severity of the error that has occurred. Five types of errors are possible:

PEND	The loss of availability of a device or component is imminent.
PERF	The performance of the device or component has degraded to below an acceptable level.
PERM	Condition that could not be recovered from. Error types with this value are usually the most severe errors and are more likely to mean that you have a defective hardware device or software module. Error types other than `PERM` usually do not indicate a defect, but they are recorded so that they can be analyzed by the diagnostics programs.
TEMP	Condition that was recovered from after a number of unsuccessful attempts. This error type is also used to record informational entries, such as data transfer statistics for DASD devices.
UNKN	It is not possible to determine the severity of the error.
INFO	The error log entry is informational and was not the result of an error.

Resource Name Name of the resource that has detected the error. For software errors. this is the name of a software component or an executable program. For hardware errors, this is the name of a device or system component. It does not indicate that the component is faulty or needs replacement. Instead, it is used to determine the appropriate diagnostic modules to be used to analyze the error.

Resource Class General class of the resource that detected the failure (for example, a device class of `disk`).

Resource Type Type of the resource that detected the failure (for example, a device type of `355mb`).

Location Code Path to the device. There may be up to four fields, which refer to drawer, slot, connector, and port, respectively.

VPD Vital product data. The contents of this field, if any, vary. Error log entries for devices typically return information concerning the device manufacturer, serial number, Engineering Change levels, and Read Only Storage levels.

Description Summary of the error.

Probable Cause Listing of some of the possible sources of the error.

User Causes List of possible reasons for errors due to user mistakes. An improperly inserted disk and external devices (such as modems and printers) that are not turned on are examples of user–caused errors.

Recommended Actions Description of actions for correcting a user–caused error.

Install Causes	List of possible reasons for errors due to incorrect installation or configuration procedures. Examples of this type of error include hardware and software mismatches, incorrect installation of cables or cable connections becoming loose, and improperly configured systems.
Recommended Actions	Description of actions for correcting an installation–caused error.
Failure Causes	List of possible defects in hardware or software. **Note:** A failure causes section in a software error log usually indicates a software defect. Logs that list user or install causes or both, but not failure causes, usually indicate that the problem is not a software defect. If you suspect a software defect, or are unable to correct user or install causes, report the problem to your software service department.
Recommended Actions	Description of actions for correcting the failure. For hardware errors, `PERFORM PROBLEM DETERMINATION PROCEDURES` is one of the recommended actions listed. For hardware errors, this will lead to running the diagnostic programs.
Detailed Data	Failure data that is unique for each error log entry, such as device sense data.

Reporting may be turned off for some errors. To show which errors have reporting turned off, enter:

```
errpt -t -F report=0 | pg
```

If reporting is turned off for any errors, enable reporting of all errors using the **errupdate** command.

Logging may also have been turned off for some errors. To show which errors have logging turned off, enter:

```
errpt -t -F log=0 | pg
```

If logging is turned off for any errors, enable logging for all errors using the **errupdate** command. Logging all errors is useful if it becomes necessary to recreate a system error.

Examples of Detailed Error Reports

The following are sample error report entries that are generated by issuing the **errpt –a** command.

An error–class value of **H** and an error–type value of **PERM** indicate that the system encountered a problem with a piece of hardware (the SCSI adapter device driver) and could not recover from it.

```
LABEL:      SCSI_ERR1
ID:         0502F666
```

```
Date/Time:        Jun 19 22:29:51
Sequence Number:  95
Machine ID:       123456789012
Node ID:          host1
Class:            H
Type:             PERM
Resource Name:    scsi0
Resource Class:   adapter
Resource Type:    hscsi
Location:         00-08
VPD:
     Device Driver Level.........00
     Diagnostic Level............00
     Displayable Message.........SCSI
     EC Level....................C25928
     FRU Number..................30F8834
     Manufacturer................IBM97F
     Part Number.................59F4566
     Serial Number...............00002849
     ROS Level and ID............24
     Read/Write Register Ptr.....0120



Description
ADAPTER ERROR



Probable Causes
ADAPTER HARDWARE CABLE
CABLE TERMINATOR DEVICE



Failure Causes
ADAPTER
CABLE LOOSE OR DEFECTIVE



          Recommended Actions
          PERFORM PROBLEM DETERMINATION PROCEDURES
          CHECK CABLE AND ITS CONNECTIONS



Detail Data
SENSE DATA
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

An error–class value of **H** and an error–type value of **PEND** indicate that a piece of hardware (the Token Ring) may become unavailable soon due to numerous errors detected by the system.

```
LABEL:    TOK_ESERR
ID:       AF1621E8
```

```
Date/Time:       Jun 20 11:28:11
Sequence Number: 17262
Machine Id:      123456789012
Node Id:         host1
Class:           H
Type:            PEND
Resource Name:   TokenRing
Resource Class:  tok0
Resource Type:   Adapter
Location:        TokenRing



Description
EXCESSIVE TOKEN-RING ERRORS



Probable Causes
TOKEN-RING FAULT DOMAIN



Failure Causes
TOKEN-RING FAULT DOMAIN



        Recommended Actions
        REVIEW LINK CONFIGURATION DETAIL DATA
        CONTACT TOKEN-RING ADMINISTRATOR RESPONSIBLE FOR THIS LAN



Detail Data
SENSE DATA
0ACA 0032 A440 0001 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 2080 0000 0000 0010 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 78CC 0000 0000 0005 C88F 0304 F4E0 0000 1000 5A4F 5685
1000 5A4F 5685 3030 3030 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000
```

An error–class value of **S** and an error–type value of **PERM** indicate that the system encountered a problem with software and could not recover from it.

```
LABEL:    DSI_PROC
ID:       20FAED7F
```

```
Date/Time:       Jun 28 23:40:14
Sequence Number: 20136
Machine Id:      123456789012
Node Id:         123456789012
Class:           S
Type:            PERM
Resource Name:   SYSVMM



Description
Data Storage Interrupt, Processor




Probable Causes
SOFTWARE PROGRAM




Failure Causes
SOFTWARE PROGRAM




        Recommended Actions
        IF PROBLEM PERSISTS THEN DO THE FOLLOWING
        CONTACT APPROPRIATE SERVICE REPRESENTATIVE




Detail Data
Data Storage Interrupt Status Register
4000 0000
Data Storage Interrupt Address Register
0000 9112
Segment Register, SEGREG
D000 1018
EXVAL
0000 0005
```

An error–class value of **S** and an error–type value of **TEMP** indicate that the system encountered a problem with software. After several attempts, the system was able to recover from the problem.

```
LABEL:          SCSI_ERR6
ID:             52DB7218
```

```
Date/Time:       Jun 28 23:21:11
Sequence Number: 20114
Machine Id:      123456789012
Node Id:         host1
Class:           S
Type:            INFO
Resource Name:   scsi0



Description
SOFTWARE PROGRAM ERROR




Probable Causes
SOFTWARE PROGRAM




Failure Causes
SOFTWARE PROGRAM




        Recommended Actions
        IF PROBLEM PERSISTS THEN DO THE FOLLOWING
        CONTACT APPROPRIATE SERVICE REPRESENTATIVE




Detail Data
SENSE DATA
0000 0000 0000 0000 0000 0011 0000 0008 000E 0900 0000 0000 FFFF
FFFE 4000 1C1F 01A9 09C4 0000 000F 0000 0000 0000 0000 FFFF FFFF
0325 0018 0040 1500 0000 0000 0000 0000 0000 0000 0000 0000 0800
0000 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000
```

An error class value of **O** indicates that an informational message has been logged.

```
LABEL:     OPMSG
ID:        AA8AB241
```

```
Date/Time:       Jul 16 03:02:02
Sequence Number: 26042
Machine Id:      123456789012
Node Id:         host1
Class:           O
Type:            INFO
Resource Name:   OPERATOR



Description
OPERATOR NOTIFICATION




User Causes
errlogger COMMAND



        Recommended Actions
        REVIEW DETAILED DATA




Detail Data
MESSAGE FROM errlogger COMMAND
hdisk1 : Error log analysis indicates a hardware failure.
```

Example of a Summary Error Report

The following is an example of a summary error report generated using the **errpt** command. One line of information is returned for each error entry.

Note: ”Error Identifiers for the Error Log”, on page C-1 lists the possible error log entries by error label.

```
ERROR_
IDENTIFIER TIMESTAMP  T CL RESOURCE_NAME ERROR_DESCRIPTION
192AC071   0101000070 I 0  errdemon      Error logging turned off
0E017ED1   0405131090 P H  mem2          Memory failure
9DBCFDEE   0101000070 I 0  errdemon      Error logging turned on
038F2580   0405131090 U H  scdisk0       UNDETERMINED ERROR
AA8AB241   0405130990 I O  OPERATOR      OPERATOR NOTIFICATION
```

Generating an Error Report

Use the following procedure to create an error report of software or hardware problems.

1. Determine if error logging is on or off. To do this, determine if the error log contains entries:

   ```
   errpt -a
   ```

The **errpt** command generates an error report from entries in the system error log.

If the error log does not contain entries, error logging has been turned off. Activate the facility by entering:

```
/usr/lib/errdemon
```

Note: You must have root user access to run this command.

The **errdemon** daemon starts error logging and writes error log entries in the system error log. If the daemon is not running, errors are not logged.

2. Generate an error log report using the **errpt** command. For example, to see all the errors for the `hdisk1` disk drive, enter:

   ```
   errpt -N hdisk1
   ```

3. Generate an error log report using SMIT. For example, use the **smit errpt** command:

   ```
   smit errpt
   ```

   Select **1** to send the error report to standard output or **2** to send the report to the printer.

   Select **yes** to display or print error log entries as they occur; otherwise, select `no` .

   Specify the appropriate device name in the **Select resource names** option (such as `hdisk1`).

   Select **Do**.

Stopping an Error Log

This procedure describes how to stop the error logging facility. Ordinarily, you would not want to turn off the error logging facility. Instead, you should clean the error log of old or unnecessary entries. For instructions about cleaning the error log, refer to ”Cleaning an Error Log”, on page 10-16.

You should turn off the error logging facility when installing or experimenting with new software or hardware. This way the error logging daemon does not use CPU time to log problems you know you are causing.

Note: You must have root user authority to use the command in this procedure.

Enter the **errstop** command to turn off error logging:

```
errstop
```

The **errstop** command stops the error logging daemon from logging entries.

Cleaning an Error Log

This procedure describes how to strip old or unnecessary entries from your error log. Cleaning is normally done for you as part of the daily **cron** command.

If it is not done automatically, you should probably clean the error log yourself every couple of days after you have examined the contents to make sure there are no significant errors.

You can also clean up specific errors. For example, if you get a new disk and you do not want the old disk’s errors in the log to confuse you, you can clean just the disk errors.

Delete all entries in your error log by doing either of the following:

- Use the **errclear –d** command. For example, to delete all software errors enter:

  ```
  errclear -d S 0
  ```

  The **errclear** command deletes entries from the error log that are older than a certain number of days. The `0` in the previous example means that you want to delete entries for all days.

- Use the **smit errclear** command:

  ```
  smit errclear
  ```

Copying an Error Log to Diskette or Tape

Copy an error log by:

- Use the **ls** and **backup** commands to copy the error log to diskette. Place a formatted diskette into the diskette drive and enter:

  ```
  ls /var/adm/ras/errlog | backup -ivp
  ```

- To copy the error log to tape, place a tape in the drive and enter:

  ```
  ls /var/adm/ras/errlog | backup -ivpf/dev/rmt0
  ```

  OR

- Use the **snap** command to gather system configuration information in a **tar** file and copy it to diskette. Place a formatted diskette into the diskette drive and enter:

  Note: You need root user authority to use the **snap** command.

  ```
  snap -a -o /dev/rfd0
  ```

  The **snap** command in this example uses the **–a** flag to gather all information about your system configuration. The **–o** flag copies the compressed **tar** file to the device you name. `/dev/rfd0` names your disk drive.

  Enter the following command to gather all configuration information in a **tar** file and copy it to tape:

  ```
  snap -a -o /dev/rmt0
  ```

  `/dev/rmt0` names your tape drive.

  See the **snap** command in *AIX Commands Reference* for more information.

Error Logging and Alerts

If the `Alert` field of an error record template is set to `True`, programs which process alerts use the following fields in the error log to build an alert:

- `Class`
- `Type`
- `Description`
- `Probable Cause`
- `User Cause`
- `Install Cause`
- `Failure Cause`
- `Recommended Action`
- `Detail Data`

These template fields must be set up according to the SNA Generic Alert Architecture. Alerts that are not set up according to the architecture cannot be processed properly by a receiving program, such as NetView.

Messages added to the error logging message sets must not conflict with the SNA Generic Alert Architecture. When the **errmsg** command is used to add messages, the command selects message numbers that do not conflict with the architecture.

If the `Alert` field of an error record template is set to `False`, you can use any of the messages in the error logging message catalog.

Error Logging Controls

You can control the error logging facility by using the following:

- Error Logging Commands, on page 10-19
- Error Logging Subroutines and Kernel Services, on page 10-20
- Error Logging Files, on page 10-20

Error Logging Commands

errclear

Deletes entries from the error log. This command can erase the entire error log. Removes entries with specified error ID numbers, classes, or types.

errdead

Extracts errors contained in the /dev/error buffer captured in the system dump. The system dump will contain error records if the **errdemon** daemon was not active prior to the dump.

errdemon

Reads error records from the /**dev**/**error** file and writes error log entries to the system error log. The **errdemon** also performs error notification as specified in the error notification objects in the Object Data Manager (ODM). This daemon is started automatically during system initialization.

errinstall

Can be used to add or replace messages in the error message catalog. Provided for use by software installation procedures. The system creates a backup file named *File*.**undo**. The **undo** file allows you to cancel the changes you made by issuing the **errinstall** command.

errlogger

Writes an operator message entry to the error log.

errmsg

Implements error logging in in–house applications. The **errmsg** command lists, adds, or deletes messages stored in the error message catalog. Using this command, text can be added to the `Error Description`, `Probable Cause`, `User Cause`, `Install Cause`, `Failure Cause`, `Recommended Action`, and `Detailed Data` message sets.

errpt

Generates an error report from entries in the system error log. The report can be formatted as a single line of data for each entry, or the report can be a detailed listing of data associated with each entry in the error log. Entries of varying classes and types can be omitted from or included in the report.

errstop

Stops the **errdemon** daemon, which is initiated during system initialization. Running the **errstop** command also disables some diagnostic and recovery functions of the system.

errupdate

Adds or deletes templates in the Error Record Template Repository. Modifies the Alert, Log, and Report attributes of an error template. Provided for use by software installation procedures.

Error Logging Subroutines and Kernel Services

errlog

Writes an error to the error log device driver.

errsave and **errlast**

Alllows the kernel and kernel extensions to write to the error log.

Error Logging Files

/dev/error

Provides standard device driver interfaces required by the error log component.

/dev/errorctl

Provides nonstandard device driver interfaces for controlling the error logging system.

/usr/include/sys/err_rec.h	Contains structures defined as arguments to the **errsave** kernel service and the **errlog** subroutine.
/var/adm/ras/errlog	Stores instances of errors and failures encountered by the system.
/var/adm/ras/errtmplt	Contains the Error Record Template Repository.

Related Information

The 'Error Notification Object Class in ' in *AIX General Programming Concepts : Writing and Debugging Programs* allows applications to be notified when particular errors are recorded.

Chapter 11. Trace Facility

> **Note:** In AIX Version 4, the trace facility subsystem is stored in an option package entitled Software Trace Service Aids (**bos.sysmgt.trace**). Although the base system includes minimal services for trace (including the **trcstart**, **trcstop**, **trcon**, and **trcoff** subroutines, and the **trcupdate** command), you need to install the Software Trace Service Aids package to activate the **trace** daemon and produce trace reports.

The trace facility helps you isolate system problems by monitoring selected system events. Events that can be monitored include: entry and exit to selected subroutines, kernel routines, kernel extension routines, and interrupt handlers. The trace facility captures a sequential flow of system events, providing a fine level of detail on system activity. Events are shown in time sequence and in the context of other events. The trace facility is useful in expanding the trace event information to understand who, when, how, and even why the event happened.

When the trace facility is active, information about system events is recorded in a system **trace** log file. The trace facility includes commands for activating and controlling traces and generating trace reports. Applications and kernel extensions can use several subroutines to record additional events.

Use the following procedures to activate the trace facility:

- Start the Trace Facility, on page 11-2
- Control the Trace Facility, on page 11-6

Start the Trace Facility

Use the following procedures to configure and start a system trace:

Configuring the trace Command

The **trace** command starts the tracing of system events and controls the size of and manages the **trace** log file, as well as the internal trace buffers that collect trace event data. The syntax of this command is:

```
trace [-fl] [-ad] [-s] [-h] [-jk events] [,events] [-m message]
[-o outfile][-g] [-T buf_sz] [-L log_sz]
```

The various options of the **trace** command are:

–f or –l — Controls the capture of trace data in system memory. If you specify neither the **–f** nor **–l** option, the trace facility creates two buffer areas in system memory to capture the trace data. The **trace** log files and the internal **trace** buffers that collect trace event data can be managed, including their size, by this command. The **–f** or **–l** flag provides the ability to prevent data from being written to the file during data collection. The options are to collect data only until the memory buffer becomes full (**–f** for first), or to use the memory buffer as a circular buffer that captures only the last set of events that occurred before **trace** was terminated (**–l**). The **–f** and **–l** options are mutually exclusive. With either the **–f** or **–l** option, data is not transferred from the memory collection buffers to file until **trace** is terminated.

–a — Runs the **trace** collection asynchronously (as a background task), returning a normal command line prompt. Without this option, the **trace** command runs in a subcommand mode (similar to the **crash** command) and returns a > prompt. You can issue subcommands and regular shell commands from the **trace** subcommand mode by preceding the shell commands with an ! (exclamation point).

–d — Delays data collection. The trace facility is only configured. Data collection is delayed until one of the collection trigger events occurs. Various methods for triggering data collection on and off are provided. These include the following:

- **trace** subcommands
- **trace** commands
- **trace** subroutines.

–j events or –k events Specifies a set of events to include (**–j**) or exclude (**–k**) from the collection process. Specifies a list of events to include or exclude by a series of three–digit hexadecimal event IDs separated by a space.

–s Terminate trace data collection if the **trace** log file reaches its maximum specified size. The default without this option is to wrap and overwrite the data in the log file on a FIFO basis.

–h Does not write a **date**/**sysname**/**message** header to the **trace** log file.

–m message Specifies a text string (message) to be included in the **trace** log header record. The message is included in reports generated by the **trcrpt** command.

–o outfile Specifies a file to use as the log file. If you do not use the **–o** option, the default log file is **/var/adm/ras/trcfile**. To direct the trace data to standard output, code the **–o** option as **–o –**. Use this technique only to pipe the data stream to another process since the trace data contains raw binary events that are not displayable.

–g Duplicates the **trace** design for multiple channels. Channel 0 is the default channel and is always used for recording system events. The other channels are generic channels, and their use is not predefined. There are various uses of generic channels in the system. The generic channels are also available to user applications. Each created channel is a separate events data stream. Events recorded to channel 0 are mixed with the predefined system events data stream. The other channels have no predefined use and are assigned generically.

A program typically requests that a generic channel be opened by using the **trcstart** subroutine. A channel number is returned, similar to the way a file descriptor is returned when a file is opened (the channel ID). The program can record events to this channel and, thus, have a private data stream. Less frequently, the **trace** command allows a generic channel to be specifically configured by defining the channel number with this option.

–T size and –L size Specifies the size of the collection memory buffers and the maximum size of the log file in bytes.

> **Note:** Because the trace facility pins the data collection buffers, making this amount of memory unavailable to the rest of the system, the trace facility can impact performance in a memory–constrained environment. If the application being monitored is not memory–constrained, or if the percentage of memory consumed by the **trace** routine is small compared to what is available in the system, the impact of **trace** ”stolen” memory should be small.

If you do not specify a value, trace uses a default size.

Recording Trace Event Data

The data recorded for each traced event consist of a word containing the trace hook identifier and the hook type followed by a variable number of words of trace data optionally followed by a time stamp. The word containing the trace hook identifier and the hook type is called the hook word. The remaining two bytes of the hook word are called hook data and are available for recording event data.

Trace Hook Identifiers

A trace hook identifier is a three–digit hexadecimal number that identifies an event being traced. You specify the trace hook identifier in the first 12 bits of the hook word. The values 0x010 through 0x0FF are available for use by user applications. All other values are reserved for system use. The trace hook identifiers for the installed software can be listed using the **trcrpt –j** command.

The trace hook IDs, which are stored in the **/usr/include/sys/trchkid.h** file, and the trace formatting templates, which are stored in the **/etc/trcfmt** file, are shared by all the trace channels.

Hook Types

The hook type identifies the composition of the event data and is user–specified. Bits 12 through 16 of the hook word constitute the hook type. For more information on hook types, refer to the **trcgen**, **trcgenk**, and **trchook** subroutines.

Using Generic Trace Channels

The trace facility supports up to eight active trace sessions at a time. Each trace session uses a channel of the multiplexed trace special file, **/dev/systrace**. Channel 0 is used by the trace facility to record system events. The tracing of system events is started and stopped by the **trace** and **trcstop** commands. Channels 1 through 7 are referred to as generic trace channels and may be used by subsystems for other types of tracing such as data link tracing.

To implement tracing using the generic trace channels of the trace facility, a subsystem calls the **trcstart** subroutine to activate a trace channel and to determine the channel number. The subsystem modules can then record trace events using the **trcgen**, **trcgent**, **trcgenk**, or **trcgenkt** subroutine. The channel number returned by the **trcstart** subroutine is one of the parameters that must be passed to these subroutines. The subsystem can suspend and resume trace data collection using the **trcoff** and **trcon** subroutines and can deactivate a trace channel using the **trcstop** subroutine. The trace events for each channel must be written to a separate **trace** log file, which must be specified in the call to the **trcstart** subroutine. The subsystem must provide the user interface for activating and deactivating subsystem tracing.

Starting a Trace

Use the one of the following procedures to start the trace facility.

- Start the trace facility by using the **trace** command.

  Start the trace asynchronously. For example:

  ```
  trace –a
  mycmd
  trcstop
  ```

  When using the trace facility asynchronously, use the **trace** daemon to trace the selected system events (such as the **mycmd** command); then, use the **trcstop** command to stop the trace.

  OR

  Start the trace interactively. For example:

  ```
  trace
  –>!mycmd
  –>quit
  ```

When using the trace facility interactively, get into the interactive mode as denoted by the -> prompt, and use the **trace** subcommands (such as **!**) to trace the selected system events. Use the **quit** subcommand to stop the trace.

- Use **smit trace**, and choose the **Start Trace** option.

```
smit trace
```

Stopping a Trace

Use one of the following procedures to stop the trace you started earlier.

- When using **trace** asynchronously at the command line, use the **trcstop** command:

```
trace -a
mycmd
trcstop
```

When using the trace facility asynchronously, use the **trace** daemon to trace the selected system events (such as the **mycmd** command); then, use the **trcstop** command to stop the trace.

- When using **trace** interactively at the command line, use the **quit** subcommand:

```
trace
->!mycmd
->quit
```

The interactive mode is denoted by the -> prompt. Use the **trace** subcommands (such as **!**) to trace the selected system events. Use the **quit** subcommand to stop the trace.

- Use **smit trace** and choose the **Stop Trace** option:

```
smit trace
```

Generating a Trace Report

Use either of the following procedures to generate a report of events that have been traced.

- Use the **trcrpt** command:

```
trcrpt>/tmp/NewFile
```

The previous example formats the **trace** log file and sends the report to /**tmp**/**newfile**. The **trcrpt** command reads the **trace** log file, formats the trace entries, and writes a report.

- Use the **smit trcrpt** command:

```
smit trcrpt
```

Go to ”Control the Trace Facility”, on page 11-6 for information about **trace** commands, calls and subroutines, and files.

Control the Trace Facility

When the trace is configured, the trace facility controls trigger the collection of data on or off and stop the trace facility (stop deconfigures the **trace** command and unpins buffers). Trace facility controls include:

Trace Facility Commands and Subcommands

trace

Starts the tracing of system events. The **trace** log file and the internal **trace** buffers that collect trace event data can be managed, including their size, by this command.

trcdead

Extracts trace information from a system dump. If the system halts while the trace facilities are active, the contents of the internal **trace** buffers are captured. This command extracts the trace event data from the dump and writes it to the **trace** log file.

trcrpt

The **trcrpt** command formats reports of trace event data contained in the **trace** log file. You can specify the events to be included (or omitted) in the report, as well as determine the presentation of the output with this command. The **trcrpt** command uses the trace formatting templates stored in the **/etc/trcfmt** file to determine how to interpret the data recorded for each event.

trcstop

Stops the tracing of system events.

trcupdate

Updates the trace formatting templates stored in the **/etc/trcfmt** file. When you add applications or kernel extensions that record trace events, templates for these events must be added to the **/etc/trcfmt** file. The **trcrpt** command uses the trace formatting templates to determine how to interpret the data recorded for each event. Software products that record events usually run the **trcupdate** command as part of the installation process.

Trace Facility Calls and Subroutines

trcgen, trcgent

Records trace events consisting of more than five words of data. The **trcgen** subroutine may be used to record an event as part of the system event trace (trace channel 0) or to record an event on a generic trace channel (channels 1 through 7). You specify the channel number in a subroutine parameter when you record the trace event. The **trcgent** subroutine appends a time stamp to the event data.

trcgenk, trcgenkt

Used by kernel extensions to record trace events consisting of more than five words of data. May be used to record an event as part of the system event trace (trace channel 0) or to record an event on a generic trace channel (channels 1 through 7). You specify the channel number in a subroutine parameter when you record the trace event. The **trcgenkt** subroutine appends a time stamp to the event data.

trchook, utrchook

Records up to five words of data. These subroutines may be used to record an event as part of the system event trace (trace channel 0). The **utrchook** subroutine uses a special FAST–SVC path to improve performance and should be used by programs at the user (application) level.

trcoff

Suspends the collection of trace data on either the system event trace channel (channel 0) or a generic trace channel (1 through 7). The trace channel remains active and trace data collection can be resumed by using the **trcon** subroutine.

trcon

Starts the collection of trace data on a trace channel. The channel may be either the system event trace channel (0) or a generic channel (1 through 7). The trace channel, however, must have been previously activated by using the **trace** command or the **trcstart** subroutine. You can suspend trace data collection by using the **trcoff** subroutine.

trcstart

Requests a generic trace channel. This subroutine activates a generic trace channel and returns the channel ID to the calling application to use in recording trace events using the **trcgen**, **trcgent**, **trcgenk**, and **trcgenkt** subroutines.

trcstop

Frees and deactivates a generic trace channel.

Trace Facility Files

/etc/trcfmt	Contains the trace formatting templates used by the **trcrpt** command to determine how to interpret the data recorded for each event.
/usr/include/sys/trcmacros.h	Contains commonly used macros for recording trace events.
/var/adm/ras/trcfile	Default trace log file. The **trace** command allows you to specify a different trace log file.

Chapter 12. System Dump Facility

Your system generates a system dump when a severe error occurs. System dumps can also be user–initiated by users with root user authority. A system dump creates a picture of your system's memory contents. System administrators and programmers can generate a dump and analyze its contents when debugging new applications.

If your system stops with an `888` number flashing in the operator panel display, the system has generated a dump and saved it to a dump device. Go to "Copy a System Dump", on page 12-7 for information about copying the contents of the dump to other media.

To generate a system dump see:

In AIX Version 4, some of the error log and dump commands are delivered in an optionally installable package called **bos.sysmgt.serv_aid**. System dump commands included in the **bos.sysmgt.serv_aid** include the **sysdumpstart** command. See "Software Service Aids Package" , on page 1-4 for more information.

Configure a Dump Device

When an unexpected system halt occurs, the system dump facility automatically copies selected areas of kernel data to the primary dump device. These areas include kernel segment 0 as well as other areas registered in the Master Dump Table by kernel modules or kernel extensions. If the dump to the primary dump device fails and you're using AIX release 4.2.1 or later, an attempt is made to dump to the secondary dump device if it has been defined.

When you install the operating system, the dump device is automatically configured for you. By default, the primary device is /**dev**/**hd6**, which is a paging logical volume, and the secondary device is /**dev**/**sysdumpnull**.

For systems migrated from versions of AIX earlier than 4.1, the primary dump device is what it formerly was, /**dev**/**hd7**.

If a dump occurs to paging space, the system will automatically copy the dump when the system is rebooted. By default, the dump is copied to a directory in the root volume group, /**var**/**adm**/**ras**. See the **sysdumpdev** command for details on how to control dump copying.

Note: Diskless systems automatically configure a remote dump device.

If you are using AIX Version 4.3.2 or later, compressing your system dumps before they are written to the dump device will reduce the size needed for dump devices. Refer to the **sysdumpdev** command for more details.

Start a System Dump

Attention: Do not start a system dump if the flashing `888` number shows in your operator panel display. This number indicates your system has already created a system dump and written the information to your primary dump device. If you start your own dump before copying the information in your dump device, your new dump will overwrite the existing information. Follow the procedures in Chapter 2, '888 in the Operator Panel Display,' of the *AIX Messages Guide and Reference* to record the message encoded as a sequence of operator panel display values. Refer also to 'Check the Status of a System Dump', on page 12-5.

A user–initiated dump is different from a dump initiated by an unexpected system halt because the user can designate which dump device to use. When the system halts unexpectedly, a system dump is initiated automatically to the primary dump device.

You can start a system dump by using one of the methods listed below.

If you have the Software Service Aids Package installed (see 'Software Service Aids Package', on page 1-4 for more information), you have access to the **sysdumpstart** command and can start a dump using one of these methods:

- Using the Command Line, on page 12-3
- Using SMIT, on page 12-3

If you do not have the Software Services Aids Package installed, you must use one of these methods to start a dump:

- Using the Reset Button, on page 12-4
- Using Special Key Sequences, on page 12-4

Using the Command Line

Use the following steps to choose a dump device, initiate the system dump, and determine the status of the system dump:

Note: You must have root user authority to start a dump by using the **sysdumpstart** command.

1. Check which dump device is appropriate for your system (the primary or secondary device) by using the following **sysdumpdev** command:

   ```
   sysdumpdev -l
   ```

   This command lists the current dump devices. You can use the **sysdumpdev** command to change device assignments.

2. Start the system dump by entering the following **sysdumpstart** command:

   ```
   sysdumpstart -p
   ```

   This command starts a system dump on the default primary dump device. You can use the **–s** flag to specify the secondary dump device.

3. If a code shows in the operator panel display, refer to 'Check the Status of a System Dump', on page 12-5. If the operator panel display is blank, the dump was not started. Try again using the Reset button.

Using SMIT

Use the following SMIT commands to choose a dump device and start the system dump:

Note: You must have root user authority to start a dump using SMIT. SMIT uses the **sysdumpstart** command to start a system dump.

1. Check which dump device is appropriate for your system (the primary or secondary device) by using the following SMIT fast path command:

   ```
   smit dump
   ```

2. Choose the **Show Current Dump Devices** option and write the available devices on notepaper.
3. Enter the following SMIT fast path command again:

   ```
   smit dump
   ```

4. Choose either the primary (the first example option) or secondary (the second example option) dump device to hold your dump information:

   Start a Dump to the Primary Dump Device

   OR

   Start a Dump to the Secondary Dump Device

   Base your decision on the list of devices you made in step 2.

5. Refer to 'Check the Status of a System Dump', on page 12-5 if a value shows in the operator panel display. If the operator panel display is blank, the dump was not started. Try again using the Reset button.

   Note: To start a dump with the reset button or a key sequence you must have the key switch, or mode switch, in the Service position, or have set the Always Allow System Dump value to true. To do this:

   a. Use the following SMIT fast path command:

   ```
   smit dump
   ```

   b. Set the Always Allow System Dump value to true.

   This is essential on systems that do not have a mode switch.

Using the Reset Button

Start a system dump with the Reset button by doing the following (this procedure works for all system configurations and will work in circumstances where other methods for starting a dump will not):

1. Turn your machine's mode switch to the Service position, or set Always Allow System Dump to true.
2. Press the Reset button.

   Your system writes the dump information to the primary dump device.

Using Special Key Sequences

Start a system dump with special key sequences by doing the following:

1. Turn your machine's mode switch to the Service position, or set Always Allow System Dump to true.
2. Press the Ctrl–Alt 1 key sequence to write the dump information to the primary dump device, or press the Ctrl–Alt 2 key sequence to write the dump information to the secondary dump device..

Note: You can start a system dump by this method ONLY on the native keyboard.

Check the Status of a System Dump

When a system dump is taking place, status and completion codes are displayed in the operator panel display on the operator panel. When the dump is complete, a `0cx` status code displays if the dump was user initiated, a flashing `888` displays if the dump was system initiated.

You can check whether the dump was successful, and if not, what caused the dump to fail. If a `0cx` is displayed, see 'Status Codes' below. If a flashing `888` is displayed, refer to the chapter on 888 in the operator panel display in the *AIX Messages Guide and Reference*.

Note: If the dump fails and upon reboot you see an error log entry with the label `DSI_PROC` or `ISI_PROC`, and the Detailed Data area shows an **EXVAL** of `000 0005`, this is probably a paging space I/O error. If the paging space (probably /**dev**/**hd6**) is the dump device or on the same hard drive as the dump device, your dump may have failed due to a problem with that hard drive. You should run diagnostics against that disk.

Status Codes

Find your status code in the following list, and follow the instructions:

000 The kernel debugger is started. If there is an ASCII terminal attached to one of the native serial ports, enter `q dump` at the debugger prompt (`>`) on that terminal and then wait for flashing `888`s to appear in the operator panel display. After the flashing `888` appears, go to 'Check the Status of a System Dump', on page 12-5.

0c0 The dump completed successfully. Go to 'Copy a System Dump', on page 12-7.

0c1 An I/O error occurred during the dump. Go to 'System Dump Facility', on page 12-1.

0c2 A user–requested dump is not finished. Wait at least 1 minute for the dump to complete and for the operator panel display value to change. If the operator panel display value changes, find the new value on this list. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form, and report the problem to your software service department.

0c4 The dump ran out of space . A partial dump was written to the dump device, but there is not enough space on the dump device to contain the entire dump. To prevent this problem from occurring again, you must increase the size of your dump media. Go to 'Increase the Size of a Dump Device', on page 12-10.

0c5 The dump failed due to an internal error. Wait at least 1 minute for the dump to complete and for the operator panel display value to change. If the operator panel display value changes, find the new value on the list. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form and report the problem to your software service department.

0c7 A network dump is in progress, and the host is waiting for the server to respond. The value in the operator panel display should alternate between `0c7` and `0c2` or `0c9`. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form, and report the problem to your software service department.

0c8 The dump device has been disabled. The current system configuration does not designate a device for the requested dump. Enter the **sysdumpdev** command to configure the dump device.

0c9 A dump started by the system did not complete. Wait at least 1 minute for the dump to complete and for the operator panel display value to change. If the operator panel display value changes, find the new value on the list. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form and report the problem to your software service department.

0cc (For AIX Version 4.2.1 and later only) An error occured dumping to the primary device; the dump has switched over to the secondary device. Wait at least 1 minute for the dump to complete and for the three–digit display value to change. If the three–digit display value changes, find the new value on this list. If the value does not change, then the dump did not complete due to an unexpected error. Complete the Problem Summary Form and report the problem to your software service department.

c20 The kernel debugger exited without a request for a system dump. Enter the **quit dump** subcommand. Read the new three–digit value from the LED display.

Copy a System Dump

Your dump device holds the information that a system dump generates, whether generated by the system or a user. You can copy this information to either diskette or tape and deliver the material to your service department for analysis.

Note: If you intend to use a tape to send a snap image to IBM for software support. The tape must be one of the following formats: **8mm, 2.3 Gb** capacity, **8mm, 5.0 Gb** capacity, or **4mm, 4.0 Gb** capacity. Using other formats will prevent or delay software support from being able to examine the contents.

There are two procedures for copying a system dump, depending on whether you're using a dataless workstation or a non–dataless machine:

- Copying a System Dump on a Dataless Workstation, on page 12-7
- Copying a System Dump on a Non–Dataless Machine, on page 12-8

Copying a System Dump on a Dataless Workstation

On a dataless workstation, the dump is copied to the server when the workstation is rebooted after the dump. The dump may not be available to the dataless machine.

Copy a system dump on a dataless workstation by performing the following tasks:

1. Reboot in Normal mode, on page 12-7.
2. Locate the System Dump, on page 12-7.
3. Copy the System Dump from the Server, on page 12-8.

Reboot in Normal mode

1. Switch off the power on your machine.
2. Turn the mode switch to the Normal position.
3. Switch on the power on your machine.

Locate the System Dump

Locate the dump by this procedure :

1. Log on to the server .
2. Use the **lsnim** command to find the dump object for the workstation. (For this example, the workstation's object name on the server is `worker` .)

   ```
   lsnim –l worker
   ```

   The dump object appears on the line:

   ```
   dump = dumpobject
   ```

3. Use the **lsnim** command again to determine the path of the object:

   ```
   lsnim –l dumpobject
   ```

   The path name displayed is the directory containing the dump. The dump usually has the same name as the object for the dataless workstation.

Copy the System Dump from the Server

The dump is copied like any other file. To copy the dump to tape, use the **tar** command:

```
tar -c
```

or, to copy to a tape other than /**dev**/**rmt0** :

```
tar -cftapedevice
```

To copy the dump back from the external media (such as a tape drive), use the **tar** command. Enter the following to copy the dump from /**dev**/**rmt0** :

```
tar -x
```

To copy the dump from any other media, enter:

```
tar -xftapedevice
```

Copying a System Dump on a Non–DatalessMachine

Copy a system dump on a non–dataless machine by performing the following tasks:

1. Reboot Your Machine, on page 12-8
2. Copy the System Dump using one of the following methods:
   - Copy a System Dump after Rebooting in Normal Mode, on page 12-8
   - Copy a System Dump after Booting from Maintenance Mode, on page 12-9

Reboot Your Machine

Reboot in Normal mode using the following steps:

1. Switch off the power on your machine.
2. Turn the mode switch to the Normal position.
3. Switch on the power on your machine.

If your system brings up the login prompt, go to 'Copy a System Dump after Rebooting in Normal Mode', on page 12-8.

If your system stops with a number in the operator panel display instead of bringing up the login prompt, reboot your machine from Maintenance mode, then go to 'Copy a System Dump after Booting from Maintenance Mode', on page 12-9.

Copy a System Dump after Rebooting in NormalMode

After rebooting in Normal mode, copy a system dump by doing the following:

1. Log in to your system as root user.
2. Copy the system dump to diskette (the first example) or tape (the second example) using the following **snap** command:

   ```
   /usr/sbin/snap -gfkD -o /dev/rfd0
   ```

   or

```
/usr/sbin/snap -gfkD -o /dev/rmt#
```

where # (pound sign) is the number of your available tape device (the most common is /**dev**/**rmt0**) . To find the correct number, enter the following **lsdev** command, and look for the tape device listed as `Available`:

```
lsdev -C -c tape -H
```

Note: If your dump went to a paging space logical volume, it has been copied to a directory in your root volume group, /**var**/**adm**/**ras**. See 'Configure a Dump Device' and the **sysdumpdev** command for more details. These dumps are still copied by the **snap** command. The **sysdumpdev –L** command lists the exact location of the dump.

3. To copy the dump back from the external media (such as a tape drive), use the **tar** command. Enter the following to copy the dump from /**dev**/**rmt0**:

```
tar -x
```

To copy the dump from any other media, enter:

```
tar -xftapedevice
```

Copy a System Dump after Booting from MaintenanceMode

Note: Use this procedure *only* if you cannot boot your machine in Normal mode.

1. After booting from Maintenance mode, copy a system dump to diskette (the first example) or tape (the second example) using the following **snap** command:

```
/usr/sbin/snap -gfkD -o /dev/rfd0
```

or

```
/usr/sbin/snap -gfkD -o /dev/rmt#
```

2. To copy the dump back from the external media (such as a tape drive), use the **tar** command. Enter the following to copy the dump from /**dev**/**rmt0**:

```
tar -x
```

To copy the dump from any other media, enter:

```
tar -xftapedevice
```

Increase the Size of a Dump Device

Refer to the following to determine the appropriate size for your dump logical volume and to increase the size of either a logical volume or a paging space logical volume.

Determining the Size of a Dump Device

The size required for a dump is not a constant value because the system does not dump paging space; only data that resides in real memory can be dumped. Paging space logical volumes will generally hold the system dump. However, because an incomplete dump may not be usable, follow the procedure below to make sure that you have enough dump space.

When a system dump occurs, all of the kernel segment that resides in real memory is dumped (the kernel segment is segment 0). Memory resident user data (such as u–blocks) are also dumped.

The minimum size for the dump space can best be determined using the **sysdumpdev –e** command. This gives an estimated dump size taking into account the memory currently in use by the system. If dumps are being compressed, then the estimate shown is for the compressed size of thedump, not the original size. In general, compressed dump size estimates will be much higher than the actual size. This occurs because of the unpredictability of the compression algorithm's efficiency. You should still ensure your dump device is large enough to hold the estimated size in order to avoid losing dump data.

For example, enter:

```
sysdumpdev -e
```

If **sysdumpdev –e** returns the message, `Estimated dump size in bytes: 9830400` , then the dump device should be at least 9830400 bytes or 12MB (if you are using three 4MB partitions for the disk).

Note: When a client dumps to a remote dump server, the dumps are stored as files on the server. For example, the **/export/dump/kakrafon/dump** file will contain **kakrafon's** dump. Therefore, the file system used for the **/export/dump/kakrafon** directory must be large enough to hold the client dumps.

Determining the Type of Logical Volume

1. Enter the **sysdumpdev** command to list the dump devices. The logical volume of the primary dump device will probably be **/dev/hd6** or **/dev/hd7**.

   Note: You can also determine the dump devices using SMIT. Select the **Show Current Dump Devices** option from the System Dump SMIT menu.

2. Determine your logical volume type by using SMIT. Enter the SMIT fast path **smit lvm** or **smitty lvm**. You will go directly to Logical Volumes. Select the **List all Logical Volumes by Volume Group** option.

   Find your dump volume in the list and note its Type (in the second column). For example, this might be **paging** in the case of `hd6` or **sysdump** in the case of `hd7` .

Increasing the Size of a Dump Device

If you have confirmed that your dump device is a paging space, refer to Changing or Removing a Paging Space in *AIX 4.3 System Management Guide: Operating System and Devices* for more information.

If you have confirmed that your dump device type is sysdump, refer to the **extendlv** command for more information.

Chapter 13. Remote Reboot Facility

The remote reboot facility allows the system to be rebooted through a native (integrated) serial port. The system is rebooted when the reboot string is received at the port, followed by a confirmation response of **1**. This facility is useful when the system does not otherwise respond but is capable of servicing serial port interrupts. Remote reboot can be enabled on only one native serial port at a time. The user is expected to provide their own external security for the port. This facility runs at the highest device interrupt class and a failure of the UART to clear the transmit buffer quickly may have the effect of causing other devices to lose data if their buffers overflow during this time. It is suggested that this facility only be used to reboot a machine that is otherwise ”hung” and cannot be remotely logged into. File systems will NOT be sync’d, and a potential for some loss of data which has not been flushed exists. It is strongly suggested that when remote reboot is enabled that the port not be used for any other purpose, especially file transfer, to prevent an inadvertent reboot.

Two native serial port attributes control the operation of remote reboot.

reboot_enable

Indicates whether this port is enabled to reboot the machine on receipt of the **REMOTE** reboot **STRING**, and if so, whether or not to take a system dump prior to rebooting.

```
no        - Indicates remote reboot is disabled
reboot    - Indicates remote reboot is enabled
dump      - Indicates remote reboot is enabled, and prior to
rebooting a system dump will be taken on the primary dump device.
```

reboot_string

Specifies the remote reboot string that the serial port will scan for when the remote reboot feature is enabled. When the remote reboot feature is enabled and the reboot_string is received on the port, a **’>’** character is transmitted and the system is ready to reboot. If a **’1’** character is recieved the system is rebooted; any character other than **’1’** aborts the reboot process. The reboot string has a maximum length of 16 characters and must not contain a space, colon, equal sign, null, new line, or **cntrl–**\ character.

Remote reboot can be enabled through SMIT, or via the command line. For SMIT the path **System Environments –> Manage Remote Reboot Facility** may be used for a configured TTY. Alternatively, when configuring a new TTY, remote reboot may be enabled from the **Add a TTY** or **Change/Show Characteristics of a TTY** menus. These menus are accessed through the path **Devices –> TTY**.

From the command line, the **mkdev** or **chdev** commands are used to enable remote reboot. For example, the following command enables remote reboot (with the dump option) and sets the reboot string to **ReBoOtMe** on **tty1**.

```
chdev -l tty1 -a remreboot=dump -a reboot_string=ReBoOtMe
```

This example enables remote reboot on **tty0** with the current reboot string in the database only (will take effect on the next reboot).

```
chdev -P -l tty0 -a remreboot=reboot
```

If the **tty** is being used as a normal port, then you will have to use the **pdisable** command before enabling remote reboot. You may use **penable** to reenable the port afterwards.

Appendix A. Error Message Acronyms

This appendix provides the explanations of acronyms used in this guide.

ACLST	Alternating–Current Logic Self–Test
AST	Array Self–Test
BIST	Built–In Self–Test
BOS	Base Operating System
BSCRW	Bisync Read–Write
CRC	Cyclical Redundancy Check
CD–ROM	Compact Disc Read–Only Memory
DCLST	Direct–Current Logic Self–Test
DLC	Data Link Control
EPROM	Erasable Programmable Read–Only Memory
EPOW	Early Power–Off Warning
ESCON	Enterprise Systems Connection
FDDI	Fiber Distributed Data Interface
FSLA	F–Serial Link Adapter
IDE	Integrated Device Electronics
IP	Internet Protocol
IPL	Initial Program Load
I/O	Input/Output
LAN	Local Area Network
LED	Light–Emitting Diodes
LPFK	Lighted Program Function Keyboard
LVM	Logical Volume Manager
NVRAM	Nonvolatile Random Access Memory
NETBIOS	Network Basic Input/Output System
OCS	On–Chip Sequencer
IOCC	Input/Output Channel Controller
PTY	Pseudo Terminal
RAM	Random Access Memory
ROM	Read–Only Memory
SIO	Serial Input/Output
SNA	System Network Architecture
SVC	Switched Virtual Circuit
SCSI	Small Computer System Interface
SDLC	Synchronous Data Link Control
SIMM	Single In–line Memory Module

TCW	Translation Control Words
TTY	Teletypewriter
VME	Video Monitor Extended

Appendix B. Software Validation

Validate software when you want to make sure that the installed software product (or licensed product) has not been damaged or corrupted. The files of a product are checked to make sure they exist and then for total file size, checksum values, and symbolic links. They are then checked against the Software Vital Product Data (SWVPD) database. If the values do not match, the product has been altered or damaged, and you need to contact your software service center.

There are two ways to validate software:

- Use the **lppchk** command. To check for file existence, file length, and checksum value, enter:

  ```
  lppchk -c
  ```

  The **lppchk** command can also be used for a specific product, for example:

  ```
  lppchk -c bos.rte
  ```

  To check the file links, enter:

  ```
  lppchk -f
  ```

  The **lppchk** command verifies the files of an installable software product .

- Use the **smit problem** command and choose the **Validate Software** option:

  ```
  smit problem
  ```

Appendix C. Error Identifiers for the Error Log

Your system stores information about errors and failures in the error log. This information includes error identifiers, which the error logging facility uses to compile reports. Each unique error identifier is a 32–bit CRC hexadecimal code that determines which error record template the report uses.

System/370 Parallel Adapter Error Identifiers for the Error Log

CAT ERR1

Error Description: LICENSED INTERNAL CODE (LIC) PROGRAM ABNORMALLY TERMINATED

This error appears as a notification in the adapter's status queue when the LIC has stopped for any reason.

CAT ERR2

Error Description: LICENSED INTERNAL CODE (LIC) PROGRAM ERROR

This error occurs when the call to download LIC returns with an error code or times out.

CAT ERR3

Error Description: RESOURCE UNAVAILABLE

This error occurs when a call to the **m_get** or **m_clget** macro for the **mbuf** kernel service fails. It indicates that the driver could not get an **mbuf** during a receive of data from the System/370 host.

CAT ERR4

Error Description: RESOURCE UNAVAILABLE

This error occurs when a call to the **xmalloc** subroutine fails. The driver could not allocate pinned memory for an internal data structure.

CAT ERR5

Error Description: ADAPTER ERROR

This error can occur when setting the adapter's parameters or while downloading a control–unit table that either fails or times out.

CAT ERR6

Error Description: RESOURCE UNAVAILABLE

This error indicates that a call to the **d_init** or **i_init** kernel service failed. The driver cannot be configured.

CAT ERR7

Error Description: ADAPTER ERROR

This error is caused when the internal driver lacks direct access to memory or when the receiver indicates that data has been lost.

CAT ERR8

Error Description: ADAPTER ERROR

This error indicates that an unrecoverable program input/output error occurred when bus memory was accessed.

ATE Error Identifiers for the Error Log

ATE ERR1

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by ATE (asynchronous terminal emulation) if no pacing character is received.

ATE ERR2

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by ATE if excessive transmission errors occur.

ATE ERR3

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by ATE if no acknowledgement is received from the receiving site.

ATE ERR4

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by ATE if the receiving site is not ready.

ATE ERR5

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by ATE if the sending site is not sending.

ATE ERR6

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by ATE for no carrier signal.

ATE ERR7

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by ATE if checksum error is detected.

ATE ERR8

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by ATE if sector is received twice.

ATE ERR9

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by ATE if incorrect sector is received.

ATE ERR10

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by ATE if sector could not be verified.

BADISK Error Identifiers for the Error Log

This category of identifiers contains the following error labels:

```
BADISK ERR1   BADISK ERR2   BADISK ERR3


BADISK ERR4   BADISK ERR5   BADISK ERR8
```

Go back to the categories of Error Identifiers.

BADISK ERR1

Error Description: DISK FAILURE

This error is recorded by the BUS ATTACHED DASD device driver in the event of a recovered soft read error.

DETAIL DATA

Sense Data	The sense data consists of failure data which is analyzed by the Diagnostic Programs.

BADISK ERR2

Error Description: DISK OPERATION ERROR

This error is recorded by the BUS ATTACHED DASD device driver in the event of a hard read error.

Refer to the Detail Data under BADISK ERR 1.

BADISK ERR3

Error Description: DISK OPERATION FAILURE

This error is recorded by the BUS ATTACHED DASD device driver in the event of a soft equipment check.

Refer to the Detail Data under BADISK ERR 1.

BADISK ERR4

Error Description: DISK OPERATION ERROR

This error is recorded by the BUS ATTACHED DASD device driver in the event of a hard equipment check.

Refer to the Detail Data under BADISK ERR 1.

BADISK ERR5

Error Description: DISK FAILURE

This error is recorded by the BUS ATTACHED DASD device driver in the event of an attachment error.

Refer to the Detail Data under BADISK ERR 1.

BADISK ERR8

Error Description: DISK OPERATION ERROR

This error is recorded by the BUS ATTACHED DASD device driver in the event of a seek fault.

Refer to the Detail Data under BADISK ERR 1.

C327 Error Identifiers for the Error Log

This category of identifiers contains the following error labels:

```
C327 INTR   C327 START
```

Go back to the categories of Error Identifiers.

C327 INTR

Error Description: C327 ERROR INTERRUPT

This error is logged by the 3270 Connection Adapter device driver when the control unit detects an error condition, the error is corrected, or the control unit is not responding because the coaxial (COAX) cable is disconnected.

DETAIL DATA

Reason Code This field indicates where the error occurred. Most of the errors are described in the publication *3270 Information Display System: 3274 Control Unit Description and Programmer's Guide*. The following errors could also occur:

603	Control unit is not responding (COAX disconnected).
670	Control unit is not responding (COAX disconnected).
671	Adapter card not functioning correctly.
774	Session is not available.

Line Address The number of the session which had the error (session numbers start with 1).

C327 START

Error Description: C327 START ERROR

This error is logged by the 3270 Connection Adapter device driver when a session with a host could not be started.

DETAIL DATA

Reason Code Reason the session could not be started. The following errors could also occur:

8200	Control unit is not responding (COAX disconnected).
8300	Adapter card is not functioning correctly.
8303	SNA control unit—not supported.
8306	Control unit does not support extended AEDV status.
8601	Session is not available.

Line Address The number of the session which had the error (session numbers start with 1).

CHRP Error Identifiers for the Error Log

Note: This function is available beginning with AIX Version 4.2.1.

SCAN_ERROR_CHRP

Error Description: RTAS event–scan log.

This error is logged when a call to RTAS (runtime abstraction services) provided by system firmware reports an error or event.

DETAIL DATA

Problem Data This contains the raw RTAS log returned from event–scan.

RTAS_ERROR

Error Description: RTAS call failure.

This error is logged when a call to RTAS (runtime abstraction services) returns an unexpected error code to the operating system.

DETAIL DATA

Error Code Code returned by the failing RTAS service.

Malfunction Code Internal identifier of failing RTAS service.

Problem Data Optional RTAS error log providing additional information.

MACHINE_CHECK_CHRP

Error Description: Machine check on a CHRP system.

This error can indicate a variety of problems, including memory error, data or address bus parity error, internal cache parity error, or I/O bus error.

DETAIL DATA

Machine status save/restore register 0

Register containing the machine status saved.

Machine status save/restore register 1

Register containing the machine status saved.

Problem Data Optional error log returned by RTAS.

EPOW_SUS_CHRP

Error Description: Electrical Power Event interrupt occurred.

This error is logged to notify the system that an event has occurred that may affect the system's power.

DETAIL DATA

Power Status Register

Code describing state of system power when event occurred.

Problem Data Raw unformatted RTAS error log containing additional information on the failure. This data is optional.

UPDATE_FLASH_ERR

Error Description: An update of flash–memory failed.

This error indicates that an update of the flash–memory failed. It is possible that the failure happened at a time during the update that may cause the flash–memory to be corrupted. A reboot may not be possible.

DETAIL DATA

Error Code	Numbers correspond to errors returned by the RTAS call update–flash–and–reboot, with the exception of –200, which indicates that not enough real memory is available tin the real memory heap.
Problem Data	Raw unformatted RTAS error log containing additional information on the failure. This data is optional.

COM Error Identifiers for the Error Log

COM CFG ADPT

Error Description: CONFIGURATION FAILED: ADAPTER CONFIGURED

This error is logged by the TTY device driver during configuration if the adapter has already been configured.

DETAIL DATA

ERROR CODE System error code (see **sys/errno.h**).

COM CFG BUSI

Error Description: CONFIGURATION FAILED: BAD BUS ID

This error is logged by the TTY device driver if a bad bus ID is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG BUSID

Error Description: CONFIGURATION FAILED: BUS ID OUT OF RANGE

This error is logged by the TTY device driver if a bad bus ID that is out of range is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG BUST

Error Description: CONFIGURATION FAILED: BAD BUS TYPE

This error is logged by the TTY device driver if a bad bus type is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG DEVA

Error Description: CONFIGURATION FAILED: STR_INSTALL (LOAD) FAILED

This error is logged by the TTY device driver if call to **str_install ()** fails during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG DEVD

Error Description: CONFIGURATION FAILED: STR_INSTALL (UNLOAD) FAILED

This error is logged by the TTY device driver if call to **str_install ()** fails during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG DMA

Error Description: CONFIGURATION FAILED: DMA LEVEL CONFLICT

This error is logged by the TTY device driver if a DMA level conflict is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG IFLG

Error Description: CONFIGURATION FAILED: BAD INTERRUPT FLAG

This error is logged by the TTY device driver if a bad interrupt flag is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG ILVL

Error Description: CONFIGURATION FAILED: BAD INTERRUPT LEVEL

This error is logged by the TTY device driver if a bad interrupt level is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG INTR

Error Description: CONFIGURATION FAILED: INTERRUPT PRIORITY

This error is logged by the TTY device driver if bad interrupt priority is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG MNR

Error Description: CONFIGURATION FAILED: BAD MINOR NUMBER

This error is logged by the TTY device driver if a bad minor device number is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG NADP

Error Description: CONFIGURATION FAILED: ADAPTER NOT PRESENT

This error is logged by the TTY device driver during configuration if adapter–not–present is detected.

Refer to the Detail Data under COM CFG ADPT.

COM CFG PORT

Error Description: CONFIGURATION FAILED: PORT CONFIGURED

This error is logged by the TTY device driver during configuration if the port has already been configured.

Refer to the Detail Data under COM CFG ADPT.

COM CFG RESID

Error Description: CONFIGURATION FAILED: RESID NOT CORRECT

This error is logged by the TTY device driver if bad residual count is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG SLIH

Error Description: CONFIGURATION FAILED: I_INIT OF SLIH FAILED

This error is logged by the TTY device driver if call to **i_init()** fails during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM CFG UIO

Error Description: CONFIGURATION FAILED: RESID NOT CORRECT

This error is logged by the TTY device driver during configuration if **uiomove()** fails with incorrect residual.

Refer to the Detail Data under COM CFG ADPT.

COM CFG UNK

Error Description: CONFIGURATION FAILED: BAD ADAPTER TYPE

This error is logged by the TTY device driver if bad adapter type is detected during configuration.

Refer to the Detail Data under COM CFG ADPT.

COM MEM SLIH

Error Description: CAN NOT ALLOCATE MEMORY: SLIH STRUCTURE

This error is logged by the TTY device driver if **malloc()** for SLIH structure fails.

Refer to the Detail Data under COM CFG ADPT.

COM PERM PIO

Error Description: PIO EXCEPTION

This error is logged by the TTY device driver if a programming input or output exception has occurred, indicating possible adapter failure.

DETAIL DATA

PIO csr register	PIO channel status register.
PIO dsisr register	PIO data storage interrupt status register.
PIO srval register	PIO segment register contents.
PIO dar register	PIO data address register.
Adapter Check Status	
	Adapter check status.

COM TEMP PIO

Error Description: PIO EXCEPTION

This error is logged by the TTY device driver if a temporary PIO error is detected.

DETAIL DATA

PIO csr register	
	PIO channel status register.
PIO dsisr register	PIO data storage interrupt status register.
PIO srval register	PIO segment register contents.
PIO dar register	PIO data address register.
Adapter Check Status	
	Adapter check status.
Additional information	
	Additional failure data which should be made available to the service organization if this problem requires a service call.

CXMA (128–port) Error Identifiers for the Error Log

CXMA_ERR_ASSRT

Error Description: Driver assert message.

DETAIL DATA

The 128–port device driver has detected an internal error. The Driver Line Number field contains the line number in the device driver where the error occurred. Contact your service representative.

CXMA_CFG_FEPOS

Error Description: An error occurred executing the 128–port adapter FEPOS microcode.

DETAIL DATA

Perform the ”Hardware Diagnostics” , on page 9-1 or contact your service representative.

CXMA_CG_BIOS

Error Description: An error occurred executing the 128–port adapter BIOS microcode.

DETAIL DATA

Perform the ”Hardware Diagnostics” , on page 9-1 or contact your service representative.

CXMA_CFG_RST

Error Description: The 128–port adapter did not respond to a hardware reset.

DETAIL DATA

Perform the ”Hardware Diagnostics” , on page 9-1 or contact your service representative.

CXMA_ADP_FAIL

Error Description: The 128–port device driver has detected an unrecoverable error communicating with the adapter.

DETAIL DATA

Perform the ”Hardware Diagnostics” , on page 9-1 or contact your service representative.

CXMA_CFG_MEM

Error Description: The 128–port adapter memory address in the DDS structure passed from the cfgcmxa configuration method to the driver configuration entry point is not valid. This could be caused by a corrupted ODM database.

DETAIL DATA

Contact your service representative.

CXMA_CFG_PORT

Error Description: The 128–port adapter I/O port address in the DDS structure passed from the cfgcmxa configuration method to the driver configuration entry point is not valid. This could be caused by a corrupted ODM database.

DETAIL DATA

Contact your service representative.

CXMA_CFG_MTST

Error Description: The 128–port device driver detected an error during a memory test of the 128–port adapter's dual–ported memory.

DETAIL DATA

Perform the ”Hardware Diagnostics” , on page 9-1 or contact your service representative.

CXMA_CFG_TALLOC

Error Description: The 128–port device driver detected an error attempting to allocate a trb timer structure.

DETAIL DATA

Contact your service representative.

CXMA_IO_ATT

Error Description: The 128–port device driver detected an error attempting to attach to I/O memory.

DETAIL DATA

Perform the ”Hardware Diagnostics” , on page 9-1 or contact your service representative.

CXMA_MEM_ATT

Error Description: The 128–port device driver detected an error attempting to attach to bus memory.

DETAIL DATA

Perform the ”Hardware Diagnostics” , on page 9-1 or contact your service representative.

CD–ROM, DISK, and R/W OPTICAL Error Identifiers for the Error Log

DISK ERR1

Error Description: CD–ROM, DISK, or R/W OPTICAL DRIVE OPERATION ERROR

This error is recorded by the SCSI disk device driver in the event of a media error.

DETAIL DATA

Sense Data	The sense data consists of failure data which is analyzed by the Diagnostic Programs.

DISK ERR2

Error Description: CD–ROM, DISK, or R/W OPTICAL DRIVE FAILURE

This error is recorded by the SCSI disk device driver in the event of a physical hardware error.

DETAIL DATA

Sense Data	The sense data consists of failure data which is analyzed by the Diagnostic Programs.

DISK ERR3

Error Description: CD–ROM, DISK, or R/W OPTICAL DRIVE FAILURE

This error is recorded by the SCSI disk device driver in the event of an adapter detected failure.

DETAIL DATA

Sense Data	Sense data is not applicable for this error. Zeros are recorded.

DISK ERR4

Error Description: CD–ROM, DISK, or R/W OPTICAL RECOVERED ERROR

This error is recorded by the SCSI disk device driver in the event of a recovered error. Either the device or the device driver recovered the data.

DETAIL DATA

Sense Data	The sense data consists of failure data which is analyzed by the Diagnostic Programs.

DISK ERR5

Error Description: UNDETERMINED ERROR

This error is recorded by the SCSI disk device driver in the event of an unknown failure.

DETAIL DATA

Sense Data	Sense data is not applicable for this error. Zeros are recorded.

IDE DISK ERR1

Error Description: DISK OPERATION ERROR

This error is recorded by the IDE disk device driver when a media error has been detected.

DETAIL DATA

Status Validity and ATA Command Data

The status validity and ATA command data contain information about the failed command which may be analyzed by diagnostic programs.

IDE DISK ERR2

Error Description: SOFTWARE ERROR

This error is recorded by the IDE disk device driver when a disk has aborted a command. This can occur when a disk rejects an unsupported command. It can also indicate a failure in the disk drive or IDE controller electronics.

DETAIL DATA

Status Validity and ATA Command Data

The status validity and ATA command data contain information about the failed command which may be analyzed by diagnostic programs.

IDE DISK ERR3

Error Description: DISK FAILURE

This error is recorded by the IDE disk device driver in response to a failure that is neither a media error nor an aborted command. It can occur when a disk drive has suffered a hardware failure.

DETAIL DATA

Status Validity and ATA Command Data

The status validity and ATA command data contain information about the failed command which may be analyzed by diagnostic programs.

IDE DISK ERR4

Error Description: DISK FAILURE RECOVERED DURING RETRY

This error is recorded by the IDE disk device driver in the event of a recovered error. Either the disk or the device driver recovered the error.

DETAIL DATA

Status Validity and ATA Command Data

The status validity and ATA command data contain information about the failed command which may be analyzed by diagnostic programs.

DISKETTE Error Identifiers for the Error Log

DISKETTE ERR1

Error Description: DISKETTE OPERATION ERROR

This error is logged by the diskette device driver if a diskette permanent error occurs.

DETAIL DATA

Sense Data The sense data consists of failure data which is analyzed by the Diagnostic Programs.

DISKETTE ERR2

Error Description: DISKETTE DEVICE FAILURE

This error is logged by the diskette device driver if a diskette timeout error occurs.

Refer to Detail Data under DISKETTE ERR1.

DISKETTE ERR3

Error Description: DISKETTE MEDIA ERROR

This error is logged by the diskette device driver if a diskette media error occurs.

Refer to Detail Data under DISKETTE ERR1.

DISKETTE ERR4

Error Description: DISKETTE OPERATION ERROR

This error is logged by the diskette device driver if an undetermined diskette error occurs.

Refer to Detail Data under DISKETTE ERR1.

DISKETTE ERR5

Error Description: INPUT/OUTPUT DEVICE ERROR

This error is logged by the diskette device driver if a temporary diskette PIO error occurs.

Refer to Detail Data under DISKETTE ERR1.

DISKETTE ERR6

Error Description: INPUT/OUTPUT DEVICE ERROR

This error is logged by the diskette device driver if a permanent diskette PIO error occurs.

Refer to Detail Data under DISKETTE ERR1.

ENT Error Identifiers for the Error Log

ENT ERR1

Error Description: ADAPTER ERROR

This error is logged by the Ethernet device handler for the following permanent Ethernet adapter hardware errors:

- Mismatch between firmware version in microcode and ROS level from the vital product data (VPD).
- First start command to the adapter failed to complete in a specified amount of time.
- One or more of the fields in the VPD is invalid.
- Micro Channel bus I/O access failure.

DETAIL DATA

Sense Data	The sense data consists of failure data which is analyzed by the Diagnostic Programs.

ENT ERR2

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by the Ethernet device handler for the following temporary Ethernet adapter hardware errors:

- Adapter detected a parity error and reported this error to the device handler.
- One of the following transmission errors occurred: Maximum Collisions, FIFO Underrun, Carrier Sense Lost, Clear To Send Lost, Transmit Timeout, Packet Too Short, or Packet Too Large.
- Micro Channel bus I/O access failure.

Refer to Detail Data under ENT ERR1.

ENT ERR3

Error Description: ADAPTER ERROR

This error is logged by the Ethernet device handler in the event of one of the following permanent adapter firmware errors:

- Adapter **Execute** command returned with error bit set.
- Adapter **Execute** command ”Report Configuration” returned invalid data.

Refer to Detail Data under ENT ERR1.

ENT ERR4

Error Description: ADAPTER ERROR

- Incorrect VPD data.

Refer to Detail Data under ENT ERR1.

ENT ERR5

Error Description: RESOURCE UNAVAILABLE

This error is logged by the Ethernet device handler when one of the following system errors is detected by the Ethernet device handler:

- DMA kernel facilities failed.

- DMA region facilities failed.
- Receive buffer facilities (**mbuf**) failed.

Refer to Detail Data under ENT ERR1.

ENT ERR6

Error Description: CSMA/CD LAN COMMUNICATIONS LOST

This error is logged by the Ethernet device handler when an alert for lost data is received.

Refer to Detail Data under ENT ERR1.

EPOW Error Identifiers for the Error Log

EPOW RES

Error Description: ELECTRICAL POWER RESUMED

This error is logged by the EPOW (early power off warning) interrupt indicating that power resumed.

DETAIL DATA

Power Status Register

Contents of the power status register as documented in **/usr/include/sys/iocc.h**.

EPOW SUS

Error Description: LOSS OF ELECTRICAL POWER

This error is logged by the EPOW (early power off warning) interrupt indicating that primary power is about to be lost.

DETAIL DATA

Power Status Register

Contents of the power status register as documented in **/usr/include/sys/iocc.h**.

ERRLOG Error Identifiers for the Error Log

ERRLOG OFF

Error Description: ERRDEMON TURNED OFF

This error is logged by the **error** daemon when the error logging is stopped. The system automatically turns off error logging during shutdown.

ERRLOG ON

Error Description: ERRDEMON TURNED ON

This error is logged by the **error** daemon when the error logging is started. The system automatically turns on error logging during initialization.

EU Error Identifiers for the Error Log

EU BAD ADPT

Error Description: EXPANSION UNIT ERROR

This error is logged by the Async Expansion Unit if a bad adapter is detected during configuration.

DETAIL DATA

Expansion Unit Slot Number

The expansion unit slot containing the bad adapter.

EU CFG BUSY

Error Description: CONFIGURATION FAILED: IN USE

This error is logged by the Async Expansion Unit if a configuration operation fails because the device is already in use.

EU CFG GONE

Error Description: CONFIGURATION FAILED: UNCONFIGURED

This error is logged by the Async Expansion Unit device driver if a configuration operation fails because the device is already unconfigured.

EU CFG HERE

Error Description: CONFIGURATION FAILED: ALREADY CONFIGURED

This error is logged by the Async Expansion Unit device driver if a configuration operation fails because the device is already configured.

EU CFG NADP

Error Description: CONFIGURATION FAILED: ADAPTER MISSING

This error is logged by the Async Expansion Unit device driver if a configuration operation fails because the adapter is not present.

EU CFG NPLN

Error Description: CONFIGURATION FAILED: ADAPTER MISSING

This error is logged by the Async Expansion Unit device driver if the adapter cannot be detected during configuration.

EU DIAG ACC

Error Description: CANNOT PERFORM DESTRUCTIVE DIAGNOSTICS

This error is logged by the Async Expansion Unit device driver if diagnostics are requested while the device is in use.

EU DIAG MEM

Error Description: CAN NOT ALLOCATE MEMORY: WRAP BUFFER

This error is logged by the Async Expansion Unit device driver if **malloc**() for the wrap buffer fails.

Exceptions/Interrupts Error Identifiers forthe Error Log

CHECKSTOP

Error Description: CHECKSTOP

This error is logged when the **errdaemon** is started and when the checkstop count in nonvolatile random access memory (NVRAM) indicates that a checkstop has occurred. The **errdaemon** copies the checkstop data from NVRAM to a file and records the path name of the file and the checkstop count in the error log entry. The **errdaemon** resets the checkstop count to zero.

Note: When a machine checkstops, the machine will try to reboot itself. If the reboot is unsuccessful, the system will attempt to reboot itself two more times. If each subsequent reboot is unsuccessful, the machine will stop with the value 201 in the operator display panel. If this occurs, contact your service representative for assistance.

DETAIL DATA

Checkstop Count	The value of the checkstop count from NVRAM.
Checkstop File Pathname	Path name of file where checkstop data was saved.

CORE DUMP

Error Description: SOFTWARE PROGRAM ABNORMALLY TERMINATED

This error is logged when a software program abnormally terminates and causes a core dump.

DETAIL DATA

Signal Number	The decimal value of the signal number received at program termination.
User's Process ID	The decimal value of the user ID on which the terminated program was running.
File System Serial Number	The decimal value of the file system serial number for the core file generated.
I–node Number	The decimal value of the i–node number relating to the core file.

The file–system serial number is assigned to file systems at mount time and represents the order in which mounts occurred. This serial number identifies the file system in which the core dump occurred. The i–node number identifies the directory that contains the core dump file.

Mounted file systems are associated with a linked list of virtual file system (VFS) structures. You can locate the corresponding VFS serial number as follows:

- Write a program to list the mounted filesystems and call the **stat**subroutine on each one of them. Then, compare the `st_vfs` fieldof each file system to the VFS serial number from the error log.

  OR

- Execute the **crash**command with the **vfs** subcommand to list the mounted file systems.Then, compare the values in the NUMBER column of the VFS table with theVFS serial number from the error log.

Note: The filesystem serial numberis not guaranteed to be useful following a system restart since it dependson the mount sequence.

CORRECTED SCRUB

Error Description: MEMORY SCRUBBING CORRECTED ECC ERROR

DETAIL DATA

Single Bit Signature/Syndrome Register

Hardware register containing detailed information about the error.

Single Bit Status Register

Hardware register containing the error type.

Single Bit Address Register

Word address of where the error occurred.

DMA ERR

Error Description: UNDETERMINED ERROR

This error is logged for an undetermined DMA error. Check the error log for a device or adapter DMA error.

DETAIL DATA

Bus Number	Virtual memory handle I/O channel control.
Error Code	Channel Status Register.

DOUBLE PANIC

Error Description: SOFTWARE PROGRAM ABNORMALLY TERMINATED

This error is logged by the kernel panic routine when a second system panic occurs while processing the first one.

DETAIL DATA

Panic String	String corresponding to the last kernel print before the first panic. If the panic routine was called by the assert routine, this string will contain the file name and line number within the module where assert was called.
Panic String	Panic string for the second panic.

DSI IOCC

Error Description: DATA STORAGE INTERRUPT, IOCC

This error is logged by the interrupt handler for the I/O channel controller (IOCC) type of Data Storage Interrupt.

DETAIL DATA

Data Storage Interrupt Status Register (DSISR)

Defines the cause of the Data Storage Interrupt.

Data Status Address Register (DSAR)

Address of the storage access that caused the Data Storage Interrupt.

Segment Register (SEGREG)

Segment register of unit causing exception.

Channel Status Register 15 (CSR15)

Indicates where the error occurred.

DSI PROC

Error Description: DATA STORAGE INTERRUPT, PROCESSOR

This error is logged by the interrupt handler for a processor type of Data Storage Interrupt.

DETAIL DATA

Data Storage Interrupt Status Register

Defines the cause of the Data Storage Interrupt.

Data Status Address Register

Address of the storage access that caused the Data Storage Interrupt.

Segment Register Segment register of unit causing exception.

DSI SCU

Error Description: DATA STORAGE INTERRUPT, SCU

This error is recorded by the interrupt handler for the storage control unit type of Data Storage Interrupt.

DETAIL DATA

Data Storage Interrupt Status Register

Defines the cause of the Data Storage Interrupt.

Data Status Address Register

Address of the storage access that caused the Data Storage Interrupt.

Data Storage Interrupt Reason Register

Indicates reason for Data Storage Interrupt.

Segment Register Segment register of unit causing exception.

DSI SLA

Error Description: DATA STORAGE INTERRUPT, SLA

This error is logged by the interrupt handler when a serial link adapter type of Data Storage Interrupt occurs.

DETAIL DATA

Data Storage Interrupt Status Register

Defines the cause of the Data Storage Interrupt.

Data Status Address Register

Address of the storage access that caused the Data Storage Interrupt.

Segment Register Segment register of unit causing exception.

DUMP

Error Description: DUMP DEVICE ERROR

This error is logged by the dump pseudo–device driver when it is unable to open the dump device.

DETAIL DATA

Major/Minor Device Number
Major and minor device numbers of the dump device.

Error Code System error code (see **sys/errno.h**).

DUMP STATS

Error Description: SYSTEM DUMP

This error includes the information included in the **sysdumpdev –L** command.

DETAIL DATA

Major Device Number
Number of the major device.

Minor Device Number
Number of the minor device.

Dump Size Size of the dump in bytes.

Time Time at which the dump occurred.

Dump Type Primary or secondary (1 or 2).

Dump Status The last digit of the dump LED code.

EXCHECK DMA

Error Description: External check, DMA

This error is logged by the interrupt handler if a direct memory access (DMA) type of external check occurs. It can indicate that an unknown location in memory has gone bad.

DETAIL DATA

External Check Error Status Register
Indicates source and type of external check.

External Check Error Address Register
Address of non–error correction control (ECC) type of error indicated in EESR.

External Check ECC Address Register
Address of ECC type of error indicated in EESR.

Memory Configuration Registers
Configuration registers 0, 2, 6, 8, 10, 12, 14.

EXCHECK SCRUB

Error Description: OPERATOR NOTIFICATION

This error indicates that an unrecoverable double–bit memory error has occurred.

DETAIL DATA

External Check Error Status Register

Indicates source and type of external check.

External Check Error Address Register

Address of non–ECC type of error indicated in EESR.

External Check ECC Address Register

Address of ECC type of error indicated in EESR.

Memory Configuration Registers

Configuration registers 0, 2, 6, 8, 10, 12, 14.

FLPT UNAVAIL

Error Description: FLOATING POINT UNAVAILABLE

This error is logged by the interrupt handler when the Floating Point Unavailable Interrupt occurs while in kernel mode. This can be caused by an attempted use of the floating point processor while it is in an unavailable state.

DETAIL DATA

Segment Register Segment register of the instruction that caused the exception.

Machine Status Save/Restore Register 0

SRR0 is used to save machine status on interrupts and to restore machine status when a Return From Interrupt instruction is executed. In general, SRR0 contains the instruction address which caused the interrupt or the instruction address to return to after an interrupt is serviced. For this interrupt, SRR0 contains the address of the instruction that caused the Floating Point Unavailable interrupt.

Machine State Register

The state of the processor.

INTR ERR

Error Description: UNDETERMINED ERROR

This error is logged for undetermined interrupt error.

DETAIL DATA

Bus Number Virtual memory handle (IOCC).

Bus Status Code Interrupt Level.

Error Code EiS register.

INTRPPC ERR

Error Description: UNDETERMINED ERROR

This error is logged for undetermined interrupt error.

DETAIL DATA

Bus Number Virtual memory handle (IOCC).

Bus Status Code Interrupt Level.

ISI PROC

Error Description: INSTRUCTION STORAGE INTERRUPT

This error is logged by the interrupt handler when Instruction Storage Interrupt occurs.

DETAIL DATA

Instruction Storage Interrupt Status Register

Defines the cause of the Instruction Storage Interrupt.

Instruction Storage Save/Restore Register 0

SRR0 is used to save machine status on interrupts and to restore machine status when a Return From Interrupt instruction is executed. In general, SRR0 contains the instruction address which caused the interrupt or the instruction address to return to after an interrupt is serviced. For this interrupt, SRR0 contains the address of the instruction that caused the Floating Point Unavailable Interrupt.

Segment Register	Segment register of the instruction that caused the exception.

KERNEL PANIC

Error Description: SOFTWARE PROGRAM ABNORMALLY TERMINATED

This error is recorded by the kernel panic routine. Throughout the kernel there are direct calls to panic or indirect calls via the assert routine. These calls are initiated when the kernel gets into a corrupted state that it cannot recover from.

DETAIL DATA

Assert String	If the panic routine was called by the assert routine, this string contains the file name and line number within the module where assert was called.
Panic String	String corresponding to the last kernel print before the panic.

MACHINECHECK

This error is logged by the machine check handler.

DETAIL DATA

Machine Check Error Status Register

Contents of MESR.

Machine Check Error Address Register

Contents of MEAR.

MEMORY

Error Description: MEMORY FAILURE

This error is logged during system configuration if the IPL ROS memory test detected a memory problem.

DETAIL DATA

Failing Module	Failing memory module.

MISC ERR

Error Description: MISCELLANEOUS INTERRUPT

This error is logged by the Miscellaneous Interrupt handler for I/O bus timeout or channel check.

DETAIL DATA

Bus Status Register

Contents of Bus Status Register.

Miscellaneous Interrupt Register

Contents of Miscellaneous Interrupt Register.

PROGRAM INT

Error Description: PROGRAM INTERRUPT

This error is logged by the interrupt handler when Program Interrupt occurs while in kernel mode.

DETAIL DATA

Segment Register	Segment register of the instruction that caused the exception.

Machine Status Save/Restore Register 0

SRR0 is used to save machine status on interrupts and to restore machine status when a Return From Interrupt instruction is executed. In general, SRR0 contains the instruction address which caused the interrupt or the instruction address to return to after an interrupt is serviced. For this interrupt, SRR0 contains the address of the instruction that caused the Floating Point Unavailable Interrupt.

Machine Status Save/Restore Register 1

SRR1 is used to save machine status on interrupts and to restore machine status when a Return From Interrupt instruction is executed. When an interrupt occurs, bits 0–15 of SRR1 are loaded with information specific to each interrupt. Bits 16–31 of the MSR are placed in bits 16–31 of SRR1.

Machine State Register

The state of the processor.

REBOOT ID

Error Description: System shut down by user.

This error is logged when the **shutdown** command is invoked.

DETAIL DATA

User ID	User ID of session that invoked the **shutdown** command.
Reboot Type	Soft, hard, or timed reboot.
Time to Reboot	Only used for type `timed reboot`; otherwise, the value is `0`.

SYS RESET

Error Description: System Reset Interrupt received.

This error is logged by the interrupt handler when the Reset button on the control panel is used to reboot the system.

DETAIL DATA

Keymode Switch Position at Boot Time

`service` or `normal`.

Keymode Switch Position Currently

`service` or `normal`.

Graphics Subsystem Error Identifiers for theError Log

GRAPHICS

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by the Graphics Subsystem which includes the inpute device drivers and graphics display device drivers.

An example of an LFT error log entry in the system error log.

```
ERROR LABEL:            GRAPHICS
ERROR ID:               <an errorid>




Date/Time:              <time and date>
Sequence Number:        <sequence number>
Machine Id:             <machine ID>
Node Id:                <node ID>
Class:                  <class>
Type:
Resource Name:          <resource name>


Error Description
SOFTWARE PROGRAM ERROR

Probable Causes
SOFTWARE PROGRAM


Failure Causes
SOFTWARE PROGRAM


Recommended Actions
IF PROBLEM CONTINUES TO OCCUR REPEATEDLY THEN DO THE FOLLOWING
CONTACT APPROPRIATE SERVICE REPRESENTATIVE REPORT DETAILED DATA


Detail Data
DETECTED        FAILED  RC       ERROR   LOCATION
```

The Detail Data consists of the following:

DETECTED	Specifies the name of the module where the error occurred.
FAILED	Specifies the called function that returned an error.
RC	Specifies the return code.
ERROR	Specifies a number that identifies the error.
LOCATION	Specifies a unique number (per module) to help locate the failure.

Table 1. List of Errors in the lftconfig Module.					
DETECTED	**FAILED**	**RC**	**ERROR**	**LOCATION**	**ERROR DESCRIPTION**
lftconfig	NULL	0	1000	1	LFT is already initialized. No re– covery – reboot
lftconfig	NULL	0	1001	2	LFT is in an unknown state. No recovery – reboot
lftconfig	xmalloc	0	1002	3	Could not allocate the lft_ptr data structure
lftconfig	NULL	0	1003	4	Invalid data transfer size User space –> Kernel space
lftconfig	xmalloc	0	1002	5	Could not allocate the dds structure
lftconfig	copyin	> 0	1004	6	Copy of the dds structure failed
lftconfig	NULL	0	1040	7	No font file in the DDS structure
lftconfig	xmalloc	0	1002	8	No space for font file names
lftconfig	copyin	> 0	1004	9	Could not copy font file names
lftconfig	lft _fonts _init	> 0	1041	10	Font initialization failed
lftconfig	NULL	0	1030	11	No software key– board map file pointer in the DDS struct
lftconfig	xmalloc	0	1002	12	No space for **swkb** map file names
lftconfig	copyin	> 0	1004	13	Could not copy **swkb** map file names
lftconfig	lft– _swkbd– _init	> 0	1031	14	**swkbd** initialization failed
lftconfig	lft_init	> 0	1005	15	lft initialization failed
lftconfig	lft_ STREAMS _init	> 0	1011	16	lft STREAMS initializa– tion failed
lftconfig	pincode	> 0	1012	17	Could not pin code
lftconfig	NULL	0	1013	18	lft termination failed
lftconfig	NULL	0	1014	19	Invalid command

Table 2. List of Errors in the lftinit Module.					
DETECTED	**FAILED**	**RC**	**ERROR**	**LOCATION**	**ERROR DESCRIPTION**
lftinit	fp_open– dev	> 0	1020	1	Could not open a display device driver

lftinit	get– _disp– _dataptr	> 0	1007	2	Unable to get the phys_display data structure pointer
lftinit	xmalloc	0	1002	3	Could not allocate the vtm structure
lftinit	vttinit	> 0	1060	4	Display device driver – initialization failed
lftinit	vttact	> 0	1061	5	Display device driver – activation failed
lftinit	NULL	0	1008	6	No displays initialized
lftinit	NULL	0	1009	7	Default display did not initialize
lftinit	devswqry	> 0	1010	8	Could not query the devsw table
lftinit	NULL	0	1007	9	The device has a bad status in devsw table or the dataptr=0
lftinit	get– _devsw– _dataptr	> 0	1007	10	Unable to get the phys_display data structure pointer

Table 3. List of Errors in the lftterm Module.

DETECTED	FAILED	RC	ERROR	LOCATION	ERROR DESCRIPTION
lftterm	vttdact	> 0	1062	1	Display device driver – deactivation failed
lftterm	vttterm	> 0	1063	2	Display device driver – termination failed

Table 4. List of Errors in the lftfonts Module.

DETECTED	FAILED	RC	ERROR	LOCATION	ERROR DESCRIPTION
lftfonts	xmalloc	0	1002	1	Could not allocate space for fonts
lftfonts	fp_open	> 0	1021	2	Could not open the font file
lftfonts	NULL	0	1041	3	lft did not load any fonts
lftfonts	create– _fkproc	> 0	1043	4	Could not create the font kernel process
lftfonts	fp_fstat	> 0	1022	5	Could not stat the font file
lftfonts	fp_fstat	0	1022	6	Incorrect font file size
lftfonts	fp_lseek	>0	1023	7	Font file seek error

lfttfonts	xmalloc	0	1002	8	Could not allocate space for font data
lfttfonts	fp_read	>0	1024	9	Font file read error
lfttfonts	NULL	0	1041	10	Unsupported variable width font
lfttfonts	NULL	0	1001	11	lft_ptr is NULL

Table 5. List of Errors in the lftswkbd Module.					
DETECTED	**FAILED**	**RC**	**ERROR**	**LOCATIO N**	**ERROR DESCRIPTION**
lftswkbd	fp_open	>0	1021	1	Could not open the software keyboard map file
lftswkbd	fp_fstat	>0	1022	2	Could not stat **swkbd** map file
lftswkbd	fp_lseek	>0	1023	3	**swkbd** file seek error
lftswkbd	NULL	0	1002	4	Could not allocate space for the **swkbd** file
lftswkbd	fp_read	>0	1024	5	Read of the **swkbd** map file failed

Table 6. List of Errors in the cfglft Module.					
DETECTED	**FAILED**	**RC**	**ERROR**	**LOCATIO N**	**ERROR DESCRIPTION**
cfglft	odm– _get_list	–1	1102	1	Could not receive list of adapters from the PdAt
cfglft	odm– _get_list	–1	1103	2	ODM failure re– trieving list of displays from the CuDv
cfglft	NULL	0	1105	3	Found no available displays for the lft
cfglft	malloc	0	1106	4	Could not malloc memory for the dds structure
cfglft	getattr	0	1101	5	No font*_path attribute found in either CuAt or PdAt
cfglft	NULL	0	1107	6	No fonts found in the ODM belonging to the lft
cfglft	getattr	0	1101	7	No default_disp attribute found in either CuAt or PdAt

cfglft	odm– _get_list	–1	1103	8	ODM failure in retrieving list of adapters from the CuDv
cfglft	get– _devno	>0	1108	9	Failed to get the devno of a display
cfglft	putattr	–1	1101	10	Failed to update the CuAt with a new default_disp value.
cfglft	odm– _get_list	–1	1102	11	ODM failure retriev– ing list of keyboards from PdAt
cfglft	odm– _get_obj	–1	1103	12	ODM failure re– trieving keyboard information from CuDv
cfglft	get– _devno	>0	1108	13	Failed to get the devno of a keyboard
cfglft	NULL	–1	1122	14	No available system keyboard found
cfglft	odm_get_list	–1	1102	15	ODM failure re– trieving list of fkproc attributes from PdAt
cfglft	get– _attrval	0	1101	16	Found no **swkb_path** attribute in either CuAt or PdAt
cfglft	odm– _get_first	<= 0	1102	17	Found no **swkb_path** attribute in PdAt
cfglft	open	–1	1109	18	Failed to open the system keyboard file
cfglft	mknode	>0	1110	19	**mknode** failed for the lft special file /dev/lft0
cfglft	stat	>0	1110	20	**stat** failed on lft special file /dev/lft0
cfglft	unlink	>0	1110	21	**unlink** failed on lft special file /dev/lft0
cfglft	mknode	>0	1110	22	**mknode** failed for the lft special file /dev/lft0 after unlink– ing the previous special file
cfglft	NULL	0	1111	23	Illegal parameters passed into cfglft
cfglft	NULL	0	1112	24	Logical device name passed into cfglft is invalid
cfglft	odm_– initialize	–1	1100	25	ODM failed to initialize
cfglft	odm– _get_first	0	1103	26	Failed to find the logical name for the lft in the CuDv
cfglft	odm– _get_first	–1	1103	27	ODM failure re– trieving logical name of lft from the CuDv

cfglft	odm– _get_first	0	1104	28	Failed to find the PdDv class associated with the lft
cfglft	odm– _get_first	–1	1104	29	ODM failure re– trieving PdDv class for the lft
cfglft	NULL	0	1113	30	lft is listed as missing in the CuDv
cfglft	build_dds	>0	1114	31	The build_dds function had an error building the dds
cfglft	loadext	0	1115	32	Failure occurred while trying to load the lft driver
cfglft	genmajor	–1	1116	33	Failure occurred creating the major number for the lft
cfglft	genminor	0	1117	34	Failure occurred creating the minor number for the lft
cfglft	sysconfig	–1	1118	35	Failure occurred while initializing the lft driver
cfglft	make– _special– _file	>0	1110	36	Failure occurred while creating the lft special file
cfglft	lft_auto– push	>0	1120	37	Failure occurred during auto push of STREAMS modules
cfglft	odm– _change _obj	–1	1103	38	Failure occurred while updating the lft status to available
cfglft	getattr	0	1101	39	Failure getting syscons attribute
cfglft	odm– _run– _meth– ods	–1	1118	40	odm_run_method failed doing **mkitab**
cfglft	sysconfig	>0	1115	41	Error loading 1dterm module
cfglft	sysconfig	>0	1115	42	Error loading tioc module
cfglft	stream– _ioctl	>0	1115	43	Error in ioctl to SAD to auto push modules
cfglft	loadext	0	1121	44	Failure occurred while unloading the lft driver
cfglft	sysconfig	–1	1119	45	Failure occurred while terminating the lft driver
cfglft	odm_run_metho ds	–1	1118	46	odm_run_methods failed doing rmitab

Table 7. List of Errors in the startlft Module.					
DETECTED	**FAILED**	**RC**	**ERROR**	**LOCATIO N**	**ERROR DESCRIPTION**
startlft	odm_– initialize	–1	1200	1	ODM failed to initialize
startlft	odm_– get_first	0	1203	2	No lft driver infor– mation found in the PdDv class
startlft	odm_– get_first	–1	1203	3	ODM failure re– trieving PdDv class for lft
startlft	odm_– get_first	–1	1201	4	ODM failure re– trieving information from the PdAt class
startlft	odm_– get_first	–1	1202	5	ODM failure re– trieving information from the CuDv class
startlft	NULL	0	1204	6	No available displays found for the lft
startlft	odm_– run_– method	>0	1205	7	Error occurred attempting to define the lft

Table 8. List of Errors in the ucfglft Module.					
DETECTED	**FAILED**	**RC**	**ERROR**	**LOCATIO N**	**ERROR DESCRIPTION**
pop_mod	load_ext	0	1300	1	Failed to get kmid for STREAMS module
pop_mod	NULL	0	1301	2	Unknown STREAMS module
pop_mod	sysconfig	<0	1302	3	Sysconfig failed attempting to unconfigure STREAMS module
pop_mod	loadext	0	1300	4	Failed to unload a STREAMS module
ucfglft	odm_– open_– class	–1	1303	1	Open of the CuDv class in the ODM failed
ucfglft	odm_– get_first	0	1304	2	Failed to find any objects belonging to LFT in the CuDv
ucfglft	odm_– get_first	–1	1305	3	ODM error occurred while retrieving object from CuDv
ucfglft	odm_– get_first	0	1306	4	Failed to find any objects belonging to LFT in the PdDv

ucfglft	odm_– get_first	–1	1307	5	ODM error occurred while retrieving object from PdDv
ucfglft	getmajor	–1	1308	6	Error occurred attempting to get the major number for LFT
ucfglft	getminor	0	1309	7	Error occurred attempting to get the minor number for LFT
ucfglft	loadext	0	1310	8	Error occurred attempting to unload the LFT driver
ucfglft	odm_– change_ obj	–1	1311	9	Error occurred updating LFT status to defined
ucfglft	odm_– close_ class	–1	1312	10	Error occurred closing the CuDv class of the ODM

Table 9. List of Errors in the GT1X Device Driver.

DETECTED	FAILED	RC	ERROR	LOCATIO N	ERROR DESCRIPTION
config.c	xmalloc	ENOME M	1001	1	Unable to xmalloc define device struct
define.c	xmalloc	ENOME M	1001	1	Unable to xmalloc phys_display struct
define.c	xmalloc	ENOME M	1001	2	Unable to xmalloc ddf structure
define.c	dev– swadd	>0	1003	3	Unable to add driver to device switch table
define.c	NULL	ENODEV	1007	4	Phys_display structure not found for device
open.c	i_init	>0	1006	1	Unable to register SLIH
open.c	setjmpx	EXCEPT – _IO_SGA	1004	2	Unable to register parity handler
ioctl.c	NULL	EINVAL	1002	1	Invalid ioctl command
interrupt.c	setjmpx	EXCEPT – _IO_SGA	1004	1	SGA Bus I/O Exception
draw.c	setjmpx	EXCEPT – _IO_SGA	1004	1	SGA Bus I/O Exception
draw.c	setjmpx	EXCEPT – _IO	1004	2	SGA Bus I/O Exception
load_pal.c	setjmpx	EXCEPT – _IO_SGA	1004	1	SGA Bus I/O Exception

reset.c	setjmpx	EXCEPT_ _IO_SGA	1004	1	SGA Bus I/O Exception
vtt_act.c	NULL	EINVAL	1000	1	Invalid terminal mode specified
vtt_defc.c	xmalloc	ENOMEM	1001	1	Unable to xmalloc cursor bit map
vtt_defc.c	xmalloc	ENOMEM	1001	2	Unable to xmalloc cursor inverted bit map
vtt_init.c	xmalloc	ENOMEM	1001	1	Unable to xmalloc local driver data (1d)
vtt_init.c	load_ _font_ _table	EXCEPT_ _IO_SGA	1004	2	Unable to load aix character font
vtt_init.c	xmalloc	ENOMEM	1001	3	Unable to xmalloc presentation space
vtt_init.c	setjmpx	EXCEPT_ _IO_SGA	1004	4	SGA Bus I/O Exception
vtt_scr.c	setjmpx	EXCEPT_ _IO_SGA	1004	1	SGA Bus I/O Exception

Table 10. List of Errors in the vtt2drvr Module Color gda/Gray gda Device Driver.

DETECTED	FAILED	RC	ERROR	LOCATION	ERROR DESCRIPTION
vtt2drvr	setjmpx	EXCEPT_ _IO	1000	1	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT_ _IO	1000	2	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT_ _IO	1000	3	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT_ _IO	1000	4	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT_ _IO	1000	5	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT_ _IO	1000	6	Unable to set parity handler
vtt2drvr	NULL	NULL	1002	7	Neither GRAPHICS nor KSR mode specified
vtt2drvr	NULL	!KSR_ _MODE	1002	8	Not in KSR_ _MODE
vtt2drvr	setjmpx	EXCEPT_ _IO	1000	9	Unable to set parity handler
vtt2drvr	NULL	NULL	1004	10	Bad Command specified

vtt2drvr	setjmpx	EXCEPT– _IO	1000	11	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT– _IO	1000	12	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT– _IO	1000	13	Unable to set parity handler
vtt2drvr	xmalloc	NULL	1001	14	Could not alloc memory for local data
vtt2drvr	xmalloc	NULL	1001	15	Could not alloc memory for presentation space
vtt2drvr	setjmpx	EXCEPT– _IO	1000	16	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT– _IO	1000	17	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT– _IO	1000	18	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT– _IO	1000	19	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT– _IO	1000	20	Unable to set parity handler
vtt2drvr	setjmpx	EXCEPT– _IO	1000	21	Unable to set parity handler
vtt2drvr	xmalloc	NULL	1001	22	Could not alloc memory for pd
vtt2drvr	xmalloc	NULL	1001	23	Unable to set parity handler
vtt2drvr	talloc	NULL	1005	24	Could not alloc timer for device
vtt2drvr	setjmpx	EXCEPT– _IO	1000	25	Unable to set parity handler
vtt2drvr	dev– swadd	>0	1006	26	Could not add to dev switch table
vtt2drvr	devswqry	NULL	1007	27	Null physical disp list
vtt2drvr	copyin	!NULL	1008	28	Could not copy buffer
vtt2drvr	copyin	!NULL	1008	29	Could not copy buffer
vtt2drvr	copyout	!NULL	1009	30	Could not copy buffer
vtt2drvr	NULL	NULL	1004	31	Invalid query option
vtt2drvr	NULL	NULL	1004	32	Invalid command
vtt2drvr	devswqry	NULL	1007	33	Null physical display list
vtt2drvr	NULL	DMA– _FAIL	1010	34	Invalid DMA channel ID
vtt2drvr	i_init	!INTR– _SUCC	1011	35	Could not set interrupt data
vtt2drvr	devswqry	NULL	1007	36	Null physical display list
vtt2drvr	NULL	NULL	1013	37	Device is in use
vtt2drvr	xmalloc	NULL	1001	38	Could not allo– cate dds stuff

vtt2drvr	devswqry	NULL	1007	39	Null physical display list
vtt2drvr	NULL	NULL	1004	40	Invalid command
vtt2drvr	devswqry	NULL	1007	41	Null physical display list
vtt2drvr	NULL	NULL	1007	42	Do not know which pd to release
vtt2drvr	devswdel	>0	1014	43	Could not delete entry from devsw table

Table 11. List of Errors in the vtt2intr Module Color gda/Gray gda Device Driver.

DETECTED	FAILED	RC	ERROR	LOCATION	ERROR DESCRIPTION
vtt2intr	setjmpx	EXCEPT–_IO	1000	1	Unable to set parity handler

HCON Error Identifiers for the Error Log

WHP0001

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if a memory allocation error occurs.

DETAIL DATA

File Name	File name logged by HCON.
IPC Id	IPC message queue ID.

WHP0002

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC message queue receive error occurs.

Refer to the Detail Data under WHP001.

WHP0003

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC message queue send error occurs.

Refer to the Detail Data under WHP001.

WHP0004

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC message queue create error occurs.

Refer to the Detail Data under WHP001.

WHP0005

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC message queue create error occurs.

Refer to the Detail Data under WHP001.

WHP0006

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC message queue stat error occurs.

Refer to the Detail Data under WHP001.

WHP0007

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC message queue set error occurs.

Refer to the Detail Data under WHP001.

WHP0008

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC message queue remove error occurs.

Refer to the Detail Data under WHP001.

WHP0009

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC shared segment attachment error occurs.

Refer to the Detail Data under WHP001.

WHP0010

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC shared segment detachment error occurs.

Refer to the Detail Data under WHP001.

WHP0011

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC shared segment allocation error occurs.

Refer to the Detail Data under WHP001.

WHP0012

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC shared segment start error occurs.

Refer to the Detail Data under WHP001.

WHP0013

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC shared segment set error occurs.

Refer to the Detail Data under WHP001.

WHP0014

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by HCON if IPC shared segment remove error occurs.

Refer to the Detail Data under WHP001.

IDE Error Identifiers for the Error Log

ATAIDE ERR1

Error Description: IDE ADAPTER RESET FAILURE

This error is logged by the IDE adapter device driver in the event where IDE devices do not become available after a bus reset. It is unlikely that any operations to the IDE adapter can continue after one of these errors.

DETAIL DATA

IDE Bus and Command: The IDE bus state information and current command, which is analyzed by the Diagnostic Programs.

ATAIDE ERR2

Error Description: IDE DMA TRANSFER ERROR

This error is logged by the IDE adapter device driver in the event of a temporary DMA failure. This may be caused by either a system resource problem, a device error, or a hardware problem. Since these errors are not necessarily catastrophic, operations may or may not continue successfully after the error.

IDE Bus and Command: The IDE bus state information and current command, which is analyzed by the Diagnostic Programs.

ATAIDE ERR3

Error Description: IDE DEVICE ERROR

This error is logged by the IDE adapter device driver in the event of a temporary hardware error involving the IDE adapter, the system I/O bus and related hardware, or both. Either these errors cannot be retried, or all allowed retries have been exhausted. It is unlikely that any operations to the IDE adapter can continue after one of these errors.

IDE Bus and Command: The IDE bus state information and current command, which is analyzed by the Diagnostic Programs.

ATAIDE ERR4

Error Description: IDE COMMAND TIMEOUT ERROR

This error is logged by the IDE adapter device driver in the event of a command not completing successfully because the IDE device did not respond. Since these errors are not necessarily catastrophic, operations may or may not continue successfully after the error.

IDE Bus and Command: The IDE bus state information and current command, which is analyzed by the Diagnostic Programs.

Integrated Ethernet Error Identifiers for the Error Log

IENT ERR1

Error Description: ADAPTER ERROR

This error indicates that the Ethernet controller is malfunctioning.

DETAIL DATA

Return Code	The error return code as defined in the **/usr/include/sys/errno.h** file.
Status Code	For internal use only.
Program Check Code	For internal use only.

IENT ERR2

Error Description: CONFIGURATION OR CUSTOMIZATION ERROR

This error indicates that the Ethernet device could not be activated because it could not obtain the necessary resources. The resources include DMA channels, interrupt level, bus memory space, and system memory space.

DETAIL DATA

Return Code	The error return code as defined in the **/usr/include/sys/errno.h** file.
Status Code	For internal use only.

IENT ERR3

Error Description: SOFTWARE DEVICE DRIVER

This error indicates that the Ethernet device experienced a Micro Channel error during normal operation. It is likely that data has been lost.

DETAIL DATA

Return Code	For internal use only.
Status Code	For internal use only.
Program Check Code	For internal use only.

IENT ERR4

Error Description:

This error indicates that the Ethernet driver is experiencing a shortage of system–wide mbuffers (clusters).

OR

The Ethernet driver recieved an invalid packet size.

OR

The Ethernet driver failed to free memory.

OR

The Ethernet driver failed an action command.

DETAIL DATA

Return Code	
816	mbuffer/cluster shortage. Verify the mbuffer watermarks using the **no** command. A possible fix is to bump up the following watermarks until errors are no longer being reported (the tradeoff is pinned memory usage): • **lowclust** • **lowmbuf** • **thewall** • **mb_cl_hiwat**.
683	An invalid package size was received.
1407, 162	Memory could not be freed up.
1829	An action command failed.
Status Code	The system control block command or status word.
Program Check Code	The return code.

For information concerning tuning system mbuffers, refer to the *AIX Performance Tuning Guide*, order number 86 A2 72AP.

IENT ERR5

Error Description: COMMUNICATIONS SUBSYSTEM FAILURE

This error indicates that a transmission problem has been encountered.

DETAIL DATA

Return Code	The error return code, as defined in the **/usr/include/sys/entuser.h** file.
Status Code	The transmit command block status word.
Program Check Code	The system control block status word.

LION Error Identifiers for the Error Log

LION BOX DIED

Error Description: LOST COMMUNICATION WITH 64–PORT CONCENTRATOR

This error is logged by the 64–port concentrator driver if communications with the concentrator are lost.

LION BUFFERO

Error Description: BUFFER OVERRUN: 64–PORT CONCENTRATOR

This error occurs when the hardware buffer in a 64–port concentrator is overrun.

LION CHUNKNUMC

Error Description: BAD CHUNK COUNT: 64–PORT CONTROLLER

This error occurs when the value for the number of characters in a chunk does not match the actual values in the buffer.

LION HRDWRE

Error Description: CANNOT ACCESS MEMORY ON 64–PORT CONTROLLER

This error is logged by the 64–port concentrator driver if it is unable to access memory on the 64–port controller.

LION MEM ADAP

Error Description: CANNOT ALLOCATE MEMORY: ADAP STRUCTURE

This error is logged by the 64–port concentrator driver if **malloc()** for the **adap** structure fails.

LION MEM LIST

Error Description: CANNOT ALLOCATE MEMORY: LI_ADAP

This error is logged by the 64–port concentrator driver if **malloc()** for the **li_adap** structure fails.

LION UNKCHUNK

Error Description: UNKNOWN ERROR CODE FROM THE 64–PORT CONCENTRATOR.

DETAIL DATA

ERROR CODE	Number of characters in the chunk received.

LVM Error Identifiers for the Error Log

CMDLVM

Error Description: DISK OPERATION ERROR

This error is logged by the logical volume manager (LVM).

DETAIL DATA

Physical Volume ID of physical volume where error occurred.

LVM BBDIR90

Error Description: BAD BLOCK DIRECTORY OVER 90% FULL

This message is issued when the bad block directory is over 90% full during an operation to update it. When this directory is full, bad blocks cannot be relocated.

DETAIL DATA

Current Number Of Entries

Number of entries in the bad block directory.

LVM BBDIRBAD

Error Description: BAD BLOCK RELOCATION FAILURE—PV NO LONGER RELOCATING NEW BAD BLOCKS

A bad block relocation attempt by the LVM has failed. The bad block directory on the physical volume has been corrupted.

DETAIL DATA

Block Number Number of the bad block.

LVM BBDIRERR

Error Description: BAD BLOCK RELOCATION FAILURE—PV NO LONGER RELOCATING NEW BAD BLOCKS

An update operation on the bad block directory has failed due to the reason indicated by the error code. Any bad blocks on this DASD that have not already been relocated will not be relocated.

DETAIL DATA

Directory Operation

The directory operation, if any, the system was working on at the time of the error.

LVM BBDIRFUL

Error Description: BAD BLOCK RELOCATION FAILURE

A bad block relocation operation has failed because the bad block directory on this DASD is full. Any bad blocks on this DASD that have not already been relocated will not be relocated.

LVM BBEPOOL

Error Description: BAD BLOCK RELOCATION FAILURE—PV NO LONGER RELOCATING NEW BAD BLOCKS

The bad block relocation pool has no free blocks left. Therefore, any bad blocks on this DASD that have not already been relocated will not be relocated.

LVM BBFAIL

Error Description: BAD BLOCK RELOCATION FAILURE—PV NO LONGER RELOCATING NEW BAD BLOCKS

A bad block relocation attempt by the LVM failed due to a cause other than a media error. The cause is indicated by the error code.

DETAIL DATA

Block Number Number of the bad block.

LVM BBRELMAX

Error Description: BAD BLOCK RELOCATION FAILURE—PV NO LONGER RELOCATING NEW BAD BLOCKS

An attempt at hardware relocation followed by the maximum number of attempts at software relocation have all failed for the stated bad block. No more relocations will be attempted on this physical volume.

DETAIL DATA

Block Number Number of the bad block.

LVM HWFAIL

Error Description: HARDWARE DISK BLOCK RELOCATION FAILED

A hardware relocation operation initiated by LVM for a bad disk block has been unsuccessful. This could be due to hardware limitations.

DETAIL DATA

Block Number Number of the bad block.

LVM HWREL

Error Description: HARDWARE DISK BLOCK RELOCATION ACHIEVED

A hardware disk block relocation initiated by the LVM was successful.

DETAIL DATA

Block Number Number of the bad block.

LVM MISSPVADDED

Error Description: PHYSICAL VOLUME DEFINED AS MISSING

During the varyon procedure for a volume group, a physical volume was missing. The physical volume could be powered off or physically missing.

LVM MISSPVRET

Error Description: PHYSICAL VOLUME IS NOW ACTIVE

A previously missing or removed physical volume has become active.

LVM MWCWFAIL

Error Description: MIRROR CACHE WRITE FAILED

A write operation to update the mirror write consistency cache failed.

DETAIL DATA

Block Number Number of the bad block.

LVM SA FRESHPP

Error Description: PHYSICAL PARTITION MARKED ACTIVE

A formerly stale copy of a mirrored partition has been marked fresh. This could be due to its location on a formerly missing physical volume that became available and for which the **syncvg** command has been issued.

DETAIL DATA

Physical Volume Device Major/Minor

The device number of the major/minor physical volume.

Logical Volume Device Major/Minor

The device number of the major/minor logical volume.

LVM SA WRTERR

Error Description: FAILED TO WRITE VOLUME GROUP STATUS AREA

A write operation to update the volume group status area did not complete successfully. The volume group status area contains the status of mirrored partitions and of the physical volumes.

LVM SA PVMISS

Error Description: PHYSICAL VOLUME DECLARED MISSING

A physical volume is missing while the system is writing a volume group status area (VGSA). The VGSA contains the status of mirrored partitions and of physical volumes.

LVM SA STALEPP

Error Description: PHYSICAL PARTITION MARKED STALE

A copy of a mirrored partition has been marked stale. This could be due to its location on a missing physical volume at the time the partition is updated.

DETAIL DATA

Physical Volume Device Major/Minor

The device number of the major/minor physical volume.

Logical Volume Device Major/Minor

The device number of the major/minor logical volume.

LVM SA QUORCLOSE

Error Description: QUORUM LOST, VOLUME GROUP CLOSING

If a plurality of volume group status areas are not active, then the volume group is shut down. The check of the quorum count is done whenever a physical volume is missing during a volume group status area update.

DETAIL DATA

Quorum Count	The number of the status areas that reside in the volume group.
Active Count	The number of active status areas in the volume group.

LVM SWREL

Error Description: SOFTWARE DISK BLOCK RELOCATION ACHIEVED

A software disk block relocation initiated by the LVM was successful.

DETAIL DATA

Block Number	Number of the bad block.

MACHINE CHECK Error Identifiers for the Error Log

MACHINE CHECK 604

Error Description: MACHINE CHECK on a Power PC 604 type processor.

This error indicates either a multiple memory parity error, a data or address bus parity error, or an internal cache parity error.

DETAIL DATA

TIME STAMP Time when the error occurred.

MACHINE STATUS SAVE / RESTORE REGISTER 0

Register containing the machine status saved.

MACHINE STATUS SAVE / RESTORE REGISTER 1

Register containing the machine status saved.

MEM Error Identifiers for the Error Log

MEM1

Error Description: MEMORY FAILURE

This error indicates the absence of a memory card out of a memory card pair.

DETAIL DATA

Card Slot	The card slots of the affected memory cards.

MEM2

Error Description: MEMORY FAILURE

This error indicates the failure of up to two SIMMs on a memory card.

DETAIL DATA

Card Slot	The card slot of the affected memory card.
Memory SIMM Number	The identification number of the failing SIMM or SIMMs.

MEM3

Error Description: MEMORY FAILURE

This error indicates the failure of a memory card out of a memory card pair.

DETAIL DATA

Card Slot	The card slot of the affected memory card.

MPQP Error Identifiers for the Error Log

MPQP ADPERR

Error Description: MPQP ADAPTER ERROR

This error is logged by the MPQP device driver when invalid conditions are returned by the adapter code or the adapter is not functioning.

DETAIL DATA

File Name	Module and line number where the error occurred.
Port Number	Port number generating the condition.
Command	Command generating the condition.
Receive Error	Indicates the type of error that caused the receive error threshold to be exceeded. Refer to **sys/mpqp.h** for the codes that correspond to each error type.

MPQP ASWCHK

Error Description: MPQP ADAPTER SOFTWARE CHECKSUM ERROR.

This error is logged by the MPQP device driver when checksum on adapter software did not compare.

Refer to Detail Data under MPQP ADPERR.

MPQP BFR

Error Description: MPQP BUFFER ALLOCATION

This error is logged by the MPQP device driver when the system is unable to allocate a needed memory resource to the driver.

Refer to Detail Data under MPQP ADPERR.

MPQP CTSDRP

Error Description: MPQP CTS DROPPED DURING TRANSMIT

This error is logged by the MPQP device driver when CTS (clear to send) is dropped while a transmission is in progress.

Refer to Detail Data under MPQP ADPERR.

MPQP CTSTO

Error Description: COMMUNICATION PROTOCOL ERROR

This error is logged by the MPQP device driver when CTS (clear to send) fails to come on during transmission.

Refer to Detail Data under MPQP ADPERR.

MPQP DSRDRP

Error Description: MPQP DSR DROPPED

This error is logged by the MPQP device driver when DSR (data set ready) is dropped while DTR (data terminal ready) is still on.

Refer to Detail Data under MPQP ADPERR.

MPQP DSROFFTO

Error Description: MPQP DSR OFF TIMEOUT

This error is caused by starting a call while the DSR (data set ready) is on. If the DSR is on before the adapter drives DTR, the adapter assumes the previous call is still in progress. This is a security measure to ensure the previous session was terminated before the next call starts.

Refer to Detail Data under MPQP ADPERR.

MPQP DSRTO

Error Description: MPQP DSR TIMEOUT

This error is logged by the MPQP device driver when DSR (data set ready) fails to come on. It is a result of the modem failing to signal a ready condition.

Refer to Detail Data under MPQP ADPERR.

MPQP IPLTO

Error Description: MPQP INITIAL PROGRAM LOAD TIMEOUT

This error is logged by the MPQP device driver.

Refer to Detail Data under MPQP ADPERR.

MPQP QUE

Error Description: MPQP UNABLE TO ACCESS QUEUE

This error is logged by the MPQP device driver when it has been unable to access one of the command or response queues maintained by the adapter or driver code.

Refer to Detail Data under MPQP ADPERR.

MPQP RCVERR

Error Description: MPQP RECEIVE ERROR THRESHOLD EXCEEDED

If data is not received correctly, the driver logs an error. The application may recover by retransmitting. A certain amount of receive errors may be normal and is not cause for concern unless the errors become excessive and significantly degrade throughput.

Refer to Detail Data under MPQP ADPERR.

MPQP RCVOVR

Error Description: MPQP RECEIVE OVERRUN

This error is logged by the MPQP device driver when the driver is unable to provide adapter code with transfer buffers fast enough to keep up with incoming data, possibly resulting in data loss

Refer to Detail Data under MPQP ADPERR.

MPQP X21CECLR

Error Description: MPQP X21 CALL ESTABLISHMENT CLEAR

This error is logged by the MPQP device driver if the network has cleared a call, usually due to an error. If CPS data is returned, consult the CPS number to find out the reason for the clear.

Refer to Detail Data under MPQP ADPERR.

MPQP X21CPS

Error Description: MPQP X21 CALL PROGRESS SIGNAL

This error is logged by the MPQP device driver when an X.21 call progress signal is received.

Refer to Detail Data under MPQP ADPERR.

MPQP X21DTCLR

Error Description: MPQP X21 DATA TRANSFER CLEAR

This error is logged by the MPQP device driver when an unexpected X.21 clear is received during data transfer state.

Refer to Detail Data under MPQP ADPERR.

MPQP X21TO

Error Description: MPQP X21 TIMEOUT

This error is logged by the MPQP device driver if the X.21 subroutine does not complete because the timer expired. The timer is indicated in the detail data.

Refer to Detail Data under MPQP ADPERR.

MPQP XFTO

Error Description: MPQP TRANSMIT FAILSAFE TIMEOUT

This error is logged by the MPQP device driver if transmission does not complete, usually due to the following reasons: CTS not returned by modem, or clocks not detected by port.

Refer to Detail Data under MPQP ADPERR.

MPQP XMTUND

Error Description: MPQP TRANSMIT UNDERRUN

This error is logged by the MPQP device driver when the driver or adapter code is unable to provide communications hardware with data fast enough to maintain protocol timing. SYN characters will be inserted to maintain timing.

Refer to Detail Data under MPQP ADPERR.

MSLA Error Identifiers for the Error Log

MSLA ADAPTER

Error Description: ADAPTER ERROR

An error has occurred that represents an error condition interrupt from the multisubchannel line access (MSLA) adapter.

DETAIL DATA

File Name	File name associated with the location of the error.
Return Code	The return code from the failing module (see **/sys/errno.h**).
Line Address	Memory address associated with the data transmission.

MSLA CLOSE

Error Description: SOFTWARE PROGRAM ABNORMALLY TERMINATED

An error has occurred during the processing of a device driver **close** command or a driver **halt** command.

Refer to the Detail Data under MSLA ADAPTER.

MSLA INTR

Error Description: COMMUNICATION PROTOCOL ERROR

An error has occurred during the processing of interrupts from the MSLA.

Refer to the Detail Data under MSLA ADAPTER.

MSLA PROTOCOL

Error Description: COMMUNICATION PROTOCOL ERROR

The system has detected a handshaking error between the adapter and either the device driver or the internal device driver communications.

Refer to the Detail Data under MSLA ADAPTER.

MSLA START

Error Description: OUT OF RESOURCES

An error has occurred that prevents the start of a session. This error may signify a permanent program input/output error.

Refer to the Detail Data under MSLA ADAPTER.

MSLA WRITE

Error Description: ADAPTER ERROR

A write error has occurred during processing on the device driver.

Refer to the Detail Data under MSLA ADAPTER.

NETBIOS Error Identifiers for the Error Log

NB1

Error Description: SOFTWARE PROGRAM ERROR

This error indicates local area network (LAN) basic input/output (I/O) system (NETBIOS) is out of program status block (PSB) in the interrupt handler.

NB2

Error Description: SOFTWARE PROGRAM ERROR

This error indicates that NETBIOS has encountered an unknown **router** command option.

DETAIL DATA

Router Command Options

The **router** command options containing the unknown option.

NB3

Error Description: SOFTWARE PROGRAM ERROR

This error indicates that NETBIOS is out of input/output buffer.

NB4

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if the DISABLE SAP input/output control (IOCtl) fails.

DETAIL DATA

Return Code System error code (see /**sys**/**errno.h**).

NB5

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if the Service Access Point (SAP) is already closed.

NB6

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if a buffer overflow occurs.

NB7

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS in the event of inactivity without termination.

NB8

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS when activity resumes.

NB9

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS in the event of an undefined result indicator.

DETAIL DATA

Result Code The undefined result.

NB10

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS when it is out of input/output buffer in an OPEN LINK STATION.

NB11

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS in the event of an out of input/output buffer in OPEN LINK STATION completion.

NB12

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS when the OPEN LINK STATION IOCtl failed.

DETAIL DATA

Result Code System error code (see /**sys**/**errno.h**).

NB13

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS when it is out of input/output buffer in CONNECT LINK STATION.

NB14

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if CONTACT IOCtl failed.

DETAIL DATA

Return Code System error code (/**sys**/**errno.h**).

NB15

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if CONNECT LINK STATION is out of retries.

NB16

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if CONNECT LINK STATION failed.

NB17

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if HALT LINK STATION IOCtl failed.

DETAIL DATA

Return Code System error code (/**sys**/**errno.h**).

NB18

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if out of input/output buffer in an ENABLE SAP IOCtl.

NB19

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if ENABLE SAP IOCtl failed.

DETAIL DATA

Return Code	System error code (/**sys**/**errno.h**).

NB20

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if WRITE IOCtl failed.

DETAIL DATA

Return Code	System error code (/**sys**/**errno.h**).

NB21

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS in the event of an Error Operation Result from logical link control.

DETAIL DATA

Result Code	Result.

NB22

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if an unknown user–SAP correlator is received in the interrupt handler.

DETAIL DATA

SAP Correlator	The unknown SAP correlator.

NB23

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if an unknown user–SAP correlator is received.

DETAIL DATA

SAP Correlator	The unknown SAP Correlator.

NB24

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS in the event of an invalid correlator from open SAP/LS (Service Access Point/Local Service) response.

NB25

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if a call to **devswadd** subroutine fails.

DETAIL DATA

Return Code	System error code (/**sys**/**errno.h**).

NB26

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if call to **devswdel** subroutine fails.

DETAIL DATA

Return Code System error code (/**sys**/**errno.h**).

NB27

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if open still pending at the time of termination.

NB28

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if unknown configuration options are detected.

DETAIL DATA

Configuration Option

The unknown configuration option.

NB29

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if a call to **malloc** subroutine fails.

NB30

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by NETBIOS if a call to **palloc** subroutine fails.

PSLA Error Identifiers for the Error Log

PSLA001

Error Description: DEVICE ERROR

This error is logged by the primary system/serial link adapter (PSLA) device driver for hardware failures such as a PSLA graphic unit error or a cable error.

DETAIL DATA

DETECTING MODULE	The name of the module detecting the error.
FAILING MODULE	The name of the failing module.
MAJOR/MINOR DEVICE NUMBER	Major and minor device numbers.
ERROR CODE	System error code (see **sys/errno.h**).
REASON CODE	Describes type of error.
LOCATION	Unique error location information for PSLA.
SENSE DATA	The sense data consists of failure data which is analyzed by the Diagnostic Programs.

PSLA002

Error Description: DEVICE ERROR

This error is logged by the PSLA device driver if a device driver or microcode error occurs.

Refer to the Detail Data under PSLA001.

PSLA003

Error Description: LINK ERROR

This error is logged by the PSLA device driver when a host link error occurs.

Refer to the Detail Data under PSLA001.

RECOV ECC Error Identifiers for the ErrorLog

RECOV ECC ERR

Error Description: MEMORY FAILURE

This error indicates either the failure of a SIMM in a memory card, or the failure of a memory card.

DETAIL DATA

STATUS CODE	Indicates the type of the parity error:	
	0500	Single bit parity error
	0501	Multiple bit parity error
	0502	No memory parity error
	0503	Incoherent ASIC status
	0504	Incoherent interleaving table
CARD NUMBER	The card number (from **0** to **3**) of the affected card.	
SIMM LOCATIONS	The locations of the failing SIMMs. (Up to 4 SIMMs may be defective at one time.) The leftmost bit indicates SIMM number 1, while the rightmost bit indicates SIMM number 16.	
PHYSICAL MEMORY ADDRESS	Address of the physical memory where the failure was detected.	

Notes:

1. T he CARD NUMBER and the SIMM LOCATIONS data are valid only if the STATUS CODE is equal to **0** or **1** .
2. The **RECOV ECC ERR** error identifier apply to systems with either SIMM or DIMM memory modules.

RS Error Identifiers for the Error Log

RS 8 16 ARB

Error Description: INVALID 8/16 PORT ARBITRATION REGISTER

This error is logged by the TTY driver when invalid 8/16 port arbitration register is detected.

DETAIL DATA

Adapter Return Code

Adapter return code.

RS MEM EDGE

Error Description: CANNOT ALLOCATE MEMORY: EDGE STRUCTURE

This error is logged by the TTY driver when **malloc()** for TTY edge structure fails.

RS MEM EDGEV

Error Description: CANNOT ALLOCATE MEMORY: EDGE VECTOR

This error is logged by the TTY driver when **malloc()** for edge vector fails.

RS MEM IOCC

Error Description: CANNOT ALLOCATE MEMORY: IOCC STRUCTURE

This error is logged by the TTY driver when **malloc()** for TTY IOCC structure fails.

RS MEM PVT

Error Description: CANNOT ALLOCATE MEMORY: PRIV. STRUCTURE

This error is logged by the TTY driver when **malloc()** for TTY private structure fails.

RS PROG IOCC

Error Description: SOFTWARE ERROR: IOCC NOT CONFIGURED

This error is logged by the TTY driver during configuration when it detects that the IOCC is not in a configured state.

RS PROG SLIH

Error Description: SOFTWARE ERROR: CANNOT FIND SLIH

This error is logged by the TTY driver during configuration when it cannot find its **slih** structure.

SCSI Error Identifiers for the Error Log

SCSI ERR1

Error Description: ADAPTER ERROR

This error is logged by the SCSI adapter device driver in the event of a permanent hardware error involving the SCSI adapter, the system I/O bus and related hardware, or both. Either these errors cannot be retried, or all allowed retries have been exhausted. It is unlikely that any operations to the SCSI adapter can continue after one of these errors.

DETAIL DATA

Adapter Return Code: The sense data consists of failure data, which is analyzed by the Diagnostic Programs.

SCSI ERR2

Error Description: ADAPTER ERROR

This error is logged by the SCSI adapter device driver in the event of a potentially catastrophic hardware error involving the SCSI adapter, the system input/output bus and its related hardware, or both. Either these errors cannot be retried, or all allowed retries have been exhausted. Since these errors are not necessarily catastrophic, operations may or may not continue successfully after the error.

Refer to the Detail Data under SCSI ERR1.

SCSI ERR3

Error Description: MICROCODE PROGRAM ERROR

This error is logged by the SCSI adapter device driver in the event of a permanent unknown adapter microcode error. Either these errors cannot be retried, or all allowed retries have been exhausted. It is unlikely that any operations to the SCSI adapter can continue after one of these errors.

Refer to the Detail Data under SCSI ERR1.

SCSI ERR4

Error Description: MICROCODE PROGRAM ERROR

This error is logged by the SCSI adapter device driver in the event of a potentially catastrophic unknown adapter microcode error. Either these errors cannot be retried, or all allowed retries have been exhausted. Since these errors are not necessarily catastrophic, operations may or may not continue successfully after the error.

Refer to the Detail Data under SCSI ERR1.

SCSI ERR5

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by the SCSI adapter device driver in the event of a permanent catastrophic device driver logic error. Either these errors cannot be retried, or all allowed retries have been exhausted. It is unlikely that any operations to the SCSI driver can continue after one of these errors.

Refer to the Detail Data under SCSI ERR1.

SCSI ERR6

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by the SCSI adapter device driver in the event of a potentially catastrophic device driver logic error. Either these errors cannot be retried, or all allowed retries have been exhausted. Since these errors are not necessarily catastrophic, operations may or may not continue successfully after the error.

Refer to the Detail Data under SCSI ERR1.

SCSI ERR7

Error Description: UNDETERMINED ERROR

This error is logged by the SCSI adapter device driver in the event of a potentially permanent kernel service routine failure. Either these errors cannot be retried, or all allowed retries have been exhausted. It is unlikely that any operations to the SCSI adapter device driver can continue after one of these errors.

Refer to the Detail Data under SCSI ERR1.

SCSI ERR8

Error Description: UNDETERMINED ERROR

This error is logged by the SCSI adapter device driver in the event of a potentially permanent kernel service routine failure. Either these errors cannot be retried, or all allowed retries have been exhausted. Since these errors are not necessarily catastrophic, operations may or may not continue successfully after the error.

Refer to the Detail Data under SCSI ERR1.

SCSI ERR9

Error Description: POTENTIAL DATA LOSS CONDITION

This error indicates that there may be downlevel SCSI equipment installed. A potential data loss condition is possible.

SCSI ERR10

Error Description: TEMPORARY SCSI BUS ERROR

This error is logged by the SCSI device driver in the event of a potentially catastrophic SCSI bus error including cabling, device adapter, or related hardware. Since these errors are not necessarily catastrophic, operations may or may not continue successfully after the error.

SD Error Identifiers for the Error Log

SDA ERR1

Error Description: STORAGE SUBSYSTEM FAILURE

This entry indicates that an unrecoverable adapter hardware error has occurred.

DETAIL DATA

SENSE DATA The sense data consists of failure data which is analyzed by the Diagnostic Programs.

SDA ERR2

Error Description: STORAGE SUBSYSTEM FAILURE

This entry indicates that the system has encountered an unrecoverable adapter hardware error.

Refer to Detail Data under SDA ERR1.

SDA ERR3

Error Description: UNDETERMINED ERROR

This entry indicates that an unrecoverable system–detected error has occurred.

Refer to Detail Data under SDA ERR1.

SDA ERR4

Error Description: UNDETERMINED ERROR

This entry indicates that the system has encountered a recoverable system–detected error.

Refer to Detail Data under SDA ERR1.

SDC ERR1

Error Description: LINK ERROR

This entry indicates that a serial link between the controllers and DASD failed.

Refer to Detail Data under SDA ERR1.

SDC ERR2

Error Description: STORAGE SUBSYSTEM FAILURE

This entry indicates that an unrecoverable controller hardware error has occurred.

Refer to Detail Data under SDA ERR1.

SDC ERR3

Error Description: STORAGE SUBSYSTEM FAILURE

This entry indicates that the system has encountered a unrecoverable controller hardware error.

Refer to Detail Data under SDA ERR1.

SDM ERR1

Error Description: MICROCODE PROGRAM ERROR

This entry indicates that a microcode error has been detected.

Refer to Detail Data under SDA ERR1.

SLA Error Identifiers for the Error Log

SLA CRC ERR

Error Description: SLA LINK CHECK CRC ERROR

This error indicates that the CRC checksum for a frame is not correct.

DETAIL DATA

Sense Data	The sense data consists of failure data that is analyzed by the Diagnostic Programs. The `Status 1` and `Status 2` fields are valid.

SLA DRIVER ERR

Error Description: SLA LINK CHECK FAULT IN LASER DRIVE

This error indicates that the optics card has detected a fault in the transmit driver.

DETAIL DATA

Sense Data	The sense data consists of failure data that is analyzed by the Diagnostic Programs. The `Status 1` and `Status 2` fields are valid.

SLA EXCEPT ERR

Error Description: INTERNAL SERIAL LINK ADAPTER EXCEPTION

This error indicates that a PIO exception has been detected.

DETAIL DATA

Sense Data	The sense data consists of failure data that is analyzed by the Diagnostic Programs. Neither the `Status 1` nor the `Status 2` field is valid.

SLA FRAME ERR

Error Description: SLA LINK CHECK POSSIBLE LOST FRAME

This error is logged whenever a frame is lost. The detected error can be either a response time out or a send count error.

DETAIL DATA

Sense Data	The sense data consists of failure data that is analyzed by the Diagnostic Programs. If the `Status 1` field is `0xFFFFFFFF` and the `Status 2` field is `0`, then a response time out occurred. If the Status 1 field is `0` and the `Status 2` field is `0xFFFFFFFF`, then a send count error was detected.

SLA PARITY ERR

Error Description: SLA BUFFER PARITY ERROR

This error is logged when the SLA detects either a parity error in the TAG register or a parity error in the data buffer.

DETAIL DATA

Sense Data The sense data consists of failure data that is analyzed by the Diagnostic Programs. The `Status 1` field and the `Status 2` field are valid.

SLA PROG ERR

Error Description: SLA PROGRAMMING CHECK

This error is logged when the SLA detects a program check. This error can be caused if the receiver did not map enough space for the message or if there is an incorrect tag as the first TOD clock work area.

DETAIL DATA

Sense Data The sense data consists of failure data which is analyzed by the Diagnostic Programs. The `Status 1` and `Status 2` fields are valid.

SLA SIG ERR

Error Description: SLA LINK CHECK SIGNAL FAILURE

This error is logged when the SLA detects a signal error in excess of 1 second.

DETAIL DATA

Sense Data The sense data consists of failure data which is analyzed by the Diagnostic Programs. The `Status 1` and `Status 2` fields are valid.

SNA Error Identifiers for the Error Log

SNA CLDF

Error Description: CHILD PROCESS FAILURE

This error is logged by SNA when one of its child processes terminates unexpectedly or will not start.

DETAIL DATA

Component ID	SNA component name.
Detecting Module	Module that detected the error.
Module line number	Line in the detecting module.
Error Code	System error number (see /**sys**/**errno.h**).
Failing Module	Name of the process that terminated.

SNA CSA1

Error Description: INVALID XID RECEIVED

This error is logged by SNA when a module detects that the received XID is invalid.

DETAIL DATA

Sense Data	The sense data consists of failure data captured for the error.
Detecting Module	Module that detected the error.
Additional Subvector	
	Alert subvector.

SNA CSA2

Error Description: XID PROTOCOL ERROR

This error is logged by SNA when a module detects that the received XID generates a protocol error.

Refer to the Detail Data under SNA CSA1.

SNA DDCF

Error Description: UNABLE TO CONFIGURE DEVICE DRIVER

This error is logged by SNA when a SNA device driver encounters a problem during device configuration.

Refer to the Detail Data under SNA CLDF.

SNA IPCF

Error Description: SNA IPC FAILURE

This error is logged by SNA when a module encounters an error with IPC processing.

Refer to the Detail Data under SNA CLDF.

SNA OMDDF

Error Description: OPEN TO SNA MDD FAILED

This error is logged by SNA when a process is unsuccessful in its attempt to open the SNA manager device driver (MDD).

Refer to the Detail Data under SNA CLDF.

SNA PRF

Error Description: SNA PROFILE ERROR

This error is logged by SNA when a module encounters an error with the SNA profile database.

Refer to the Detail Data under SNA CLDF.

SNA SECF

Error Description: SNA SECURITY FAILURE

This error is logged by SNA when a module encounters an error with SNA security processing.

Refer to the Detail Data under SNA CLDF.

SNA SEMF

Error Description: SNA SEMAPHORE FAILURE

This error is logged by SNA when a module encounters an error with semaphore processing.

Refer to the Detail Data under SNA CLDF.

SNA SHMF

Error Description: SNA SHARED SEGMENT FAILURE

This error is logged by SNA when a problem is encountered with the SNA shared segment.

Refer to the Detail Data under SNA CLDF.

SNA SRCF

Error Description: CALL TO SRC FAILED

This error is logged by SNA when a module encounters an error with a call to the system resource controller (SRC).

Refer to the Detail Data under SNA CLDF.

SNA SRF

Error Description: SNA SYSTEM RESOURCE FAILURE

This error is logged by SNA when a module encounters a system resource failure.

Refer to the Detail Data under SNA CLDF.

TAPE Error Identifiers for the Error Log

TAPE ERR1

Error Description: TAPE OPERATION ERROR

This error is logged by the tape device driver in the event of a tape media error. Either these errors cannot be retried, or all allowed retries have been exhausted. This is normally an indication of defective media or dirty read and write heads on the tape device. It is unlikely that I/O operations to the tape device can continue after one of these errors.

DETAIL DATA

Sense Data	The sense data consists of failure data that is analyzed by the Diagnostic Programs.

TAPE ERR2

Error Description: TAPE DRIVE FAILURE

This error is logged by the tape device driver in the event of a permanent hardware error involving the tape device. Either these errors cannot be retried, or all allowed retries have been exhausted. It is unlikely that I/O operations to the tape device can continue after one of these errors.

Refer to the Detail Data under TAPE ERR1.

TAPE ERR3

Error Description: TAPE DRIVE FAILURE

This error is logged by the tape device driver in the event of a recoverable error involving either the tape device or media. Since these errors are recoverable, operations to the device can continue.

Refer to the Detail Data under TAPE ERR1.

TAPE ERR4

Error Description: TAPE DRIVE FAILURE

This error is logged by the tape device driver in the event of an adapter detected error possibly involving the tape device, the SCSI adapter, and the system I/O bus and related hardware. Either these errors cannot be retried, or all allowed retries have been exhausted. It is unlikely that I/O operations to the tape device can continue after one of these errors.

Refer to the Detail Data under TAPE ERR1.

TAPE ERR5

Error Description: UNDETERMINED ERROR

This error is logged by the tape device driver in the event of an unknown error. This error may be caused by one or more of the following: the tape device, the SCSI adapter, or the system I/O bus and related hardware and/or kernel services. Either these errors cannot be retried, or all allowed retries have been exhausted. It is unlikely that I/O operations to the tape device can continue after one of these errors.

Refer to the Detail Data under TAPE ERR1.

TAPE ERR6

Error Description: TAPE OPERATION ERROR

This error is logged by the tape device driver when the tape drive has had 30 hours of use without being cleaned. The drive continues to operate normally, but a cleaning tape should

be inserted at the first opportunity. If the drive is not cleaned, it will experience more soft errors, eventually resulting in failure of device commands.

DETAIL DATA

Sense Data The sense data consists of failure data that should be made available to the service organization is if this problem requires a service call.

TMSCSI Error Identifiers for the Error Log

TMSCSI CMD ERR

Error Description: ATTACHED SCSI TARGET DEVICE ERROR

This error indicates a hardware error that cannot be retried has occurred while the system was sending a command.

DETAIL DATA

SENSE DATA The sense data consists of failure data that is analyzed by the Diagnostic Programs.

TMSCSI READ ERR

Error Description: ATTACHED SCSI INITIATOR DEVICE

This error indicates a hardware error that cannot be retried has occurred while the system was receiving data.

Refer to the Detail Data under TMSCSI CMD ERR.

TMSCSI RECVRD ERR

Error Description: ATTACHED SCSI TARGET DEVICE ERROR

This error indicates that a recovered error has occurred while the system was sending a command.

Refer to the Detail Data under TMSCSI CMD ERR.

TMSCSI UNKN SFW ERR

Error Description: SOFTWARE PROGRAM ERROR

This error indicates that a device driver has detected a software error.

Refer to the Detail Data under TMSCSI CMD ERR.

TMSCSI UNRECVRD ERR

Error Description: ATTACHED SCSI TARGET DEVICE ERROR

This error indicates that an unrecovered error has occurred while the system was sending a command.

Refer to the Detail Data under TMSCSI CMD ERR.

TOK Error Identifiers for the Error Log

TOK ADAP CHK

Error Description: ADAPTER ERROR

This error indicates that a Token–Ring adapter check has occurred.

DETAIL DATA

Sense Data	The sense data consists of failure data which is analyzed by the Diagnostic Programs.
Additional Subvectors	Additional alert subvectors.

TOK ADAP ERR

Error Description: POTENTIAL DATA LOSS CONDITION

This error is logged when there may be downlevel Token–Ring equipment installed. A potential data loss condition is possible.

Refer to the Detail Data under TOK ADAP CHK.

TOK AUTO RMV

Error Description: AUTO REMOVAL

This error indicates that the adapter was a member of the beacon fault domain and removed itself to perform a self–test, which was unsuccessful.

Refer to the Detail Data under TOK ADAP CHK.

TOK BAD ASW

Error Description: MICROCODE PROGRAM ERROR

This error indicates an incompatible microcode and adapter.

Refer to the Detail Data under TOK ADAP CHK.

TOK BEACON1

Error Description: OPEN FAILURE

This error indicates that the adapter detected a beaconing condition on the ring during the insertion process.

Refer to the Detail Data under TOK ADAP CHK.

TOK BEACON2

Error Description: TOKEN–RING INOPERATIVE

This error indicates that the ring was in a beaconing condition for longer than the time allowed by the hard–error detection timer.

Refer to the Detail Data under TOK ADAP CHK.

TOK BEACON3

Error Description: TOKEN–RING TEMPORARY ERROR

This error indicates that ring was in a beaconing condition for less than 52 seconds and was then recovered.

Refer to the Detail Data under TOK ADAP CHK.

TOK CONGEST

Error Description: COMMUNICATIONS OVERRUN

This error indicates that the ring error monitor (REM) detected an adapter on the ring that is experiencing excessive congestion and is discarding a significant number of frames.

Refer to the Detail Data under TOK ADAP CHK.

TOK DOWNLOAD

Error Description: MICROCODE PROGRAM ABNORMALLY TERMINATED

This error indicates that the microcode download failed.

TOK DUP ADDR

Error Description: OPEN FAILURE

This error indicates that the adapter detected the presence of a station with its individual address on the ring during the insertion process.

Refer to the Detail Data under TOK ADAP CHK.

TOK ERR5

Error Description: OPEN FAILURE

This error indicates that an unknown adapter hardware error occurred during the insertion process.

Refer to the Detail Data under TOK ADAP CHK.

TOK ERR10

Error Description: AUTO REMOVAL

Refer to the Detail Data under TOK ADAP CHK.

TOK ERR15

Error Description: ADAPTER ERROR

This error indicates that the Token–Ring device handler caught an unknown system error.

Refer to the Detail Data under TOK ADAP CHK.

TOK ESERR

Error Description: EXCESSIVE TOKEN–RING ERRORS

This error indicates that the ring error monitor (REM) detected excessive soft errors for the ring.

Refer to the Detail Data under TOK ADAP CHK.

TOK MC ERR

Error Description: ADAPTER ERROR

This error indicates that the Token–Ring device handler has detected a Micro Channel error.

Refer to the Detail Data under TOK ADAP CHK.

TOK NOMBUFS

Error Description: RESOURCE UNAVAILABLE

This error indicates that a Token–Ring device handler request for an **mbuf** structure was denied.

Refer to the Detail Data under TOK ADAP CHK.

TOK PIO ERR

Error Description: ADAPTER ERROR

This error indicates that the Token–Ring device handler has detected a programmable input/output error.

Refer to the Detail Data under TOK ADAP CHK.

TOK RCVRY ENTER

Error Description: ADAPTER ERROR

This error indicates that the Token–Ring device handler has entered Network Recovery Mode.

TOK RCVRY EXIT

Error Description:

This error indicates that the Token–Ring device handler has exited Network Recovery mode successfully.

Refer to the Detail Data under TOK ADAP CHK.

TOK RCVRY TERM

Error Description: ADAPTER ERROR

This error indicates that the Token–Ring device handler has terminated Network Recovery mode.

Refer to the Detail Data under TOK ADAP CHK.

TOK RMV ADAP1

Error Description: OPEN FAILURE

This error indicates that the adapter received a Remove Ring Station MAC frame during the insertion process.

Refer to the Detail Data under TOK ADAP CHK.

TOK RMV ADAP2

Error Description: REMOVE ADAPTER COMMAND RECEIVED

This error indicates that the adapter received a Remove Adapter command from a LAN manager and left the LAN.

Refer to the Detail Data under TOK ADAP CHK.

TOK TX ERR

Error Description: ADAPTER ERROR

This error indicates that the Token–Ring device handler has detected a transmission error.

Refer to the Detail Data under TOK ADAP CHK.

TOK WIRE FAULT

Error Description: WIRE FAULT

This error indicates a wire–fault condition on the ring.

Refer to the Detail Data under TOK ADAP CHK.

TOK WRAP TST

Error Description: OPEN FAILURE

This error indicates that the adapter detected a problem on its lobe during the wrap–test portion of the insertion process.

Refer to the Detail Data under TOK ADAP CHK.

TTY Error Identifiers for the Error Log

This category of identifiers contains the following error labels:

TTY BADINPUT

Error Description: BAD TTYINPUT RETURN

This error is logged by the TTY driver if the tty driver buffer is full. Input data can no longer be loaded.

DETAIL DATA

Error Code	System error code (see **sys/errno.h**).

TTY OVERRUN

Errror Description: RECEIVER OVERRUN ON INPUT

The sending device is ignoring flow control and overloading the hardware buffer on the adapter. This occurs prior to the driver accessing the hardware FIFO.

Refer to Detail Data under TTY BADINPUT.

TTY PARERR

Error Description: PARITY/FRAMING ERROR ON INPUT

Parity errors are occurring to asynchronous ports on a character–by–character basis.

Refer to Detail Data under TTY BADINPUT.

TTY PROG PTR

Error Description: SOFTWARE ERROR: T_HPTR FIELD INVALID

This error is logged by the TTY driver if `t_hptr` pointer is null.

Refer to Detail Data under TTY BADINPUT.

TTY TTYHOG

Error Description: TTYHOG OVERRUN

The sending device is ignoring flow control. This error occurs after the hardware FIFO has been accessed and has written to the software buffer.

Refer to Detail Data under TTY BADINPUT.

VCA Error Identifiers for the Error Log

VCA INITZ

Error Description: HOST INDEPENDENT INITIALIZATION FAILED

This error indicates that a hardware problem was detected during the voice control adapter (VCA) card's configuration.

DETAIL DATA

Line Number	The line number in the source code where the error was logged.
Detecting Module	The name of the source code function that logged the error.
Failing Module	The name of the source code function where the error occurred.
Return Code	The return code sent to the user from the **perror** subroutine. The error definition may be found in the **/usr/include/sys/perror.h** file.

VCA INTR1

Error Description: INTERRUPT HANDLER REGISTRATION CODE FAILED

This error is caused by a failure in the **intr_enable** and **i_init** kernel services. This failure prevents the interrupt handler from being properly installed, but the voice control adapter device driver still functions.

Refer to Detail Data under VCA INITZ.

VCA INTR2

Error Description: UNEXPECTED INTERRUPT

This error is caused by a hardware failure. The card is creating unexpected interrupts.

Refer to Detail Data under VCA INITZ.

VCA INTR3

Error Description: INVALID INTERRUPT

This error is caused by hardware sending interrupts to the wrong track.

Refer to Detail Data under VCA INITZ.

VCA INTR4

Error Description: INTERRUPT TIMED OUT

This error is caused by hardware not sending an interrupt when an interrupt was expected.

Refer to Detail Data under VCA INITZ.

VCA IOCTL1

Error Description: INVALID IOCTL REQUEST

This error indicates that the user tried to issue an IOCtl request that is not supported by the device driver.

Refer to Detail Data under VCA INITZ.

VCA IOCTL2

Error Description: INVALID IOCTL RESPONSE

This error indicates that the user tried to issue an IOCtl request that did not have the required permissions.

Refer to Detail Data under VCA INITZ.

VCA MEM

Error Description: FAILED PINNING MEMORY

This error indicates that the VCA device driver could not pin either the **current_vca** structure or the bottom half of the driver during installation.

Refer to Detail Data under VCA INITZ.

X25 Error Identifiers for the Error Log

This category of identifiers contains the following error labels:

X25 ADAPT

Error Description: ADAPTER ERROR

This error is logged by the X.25 device driver if the adapter card could not be found on the bus.

DETAIL DATA

Sense Data	The sense data consists of failure data which is analyzed by the Diagnostic Programs.

X25 ALERT5

Error Description: X–5, MODEM FAILURE: DCD, DSR, CABLE

This error is logged by the X.25 device driver if the physical layer cannot be established to the modem. The signal lines DCD and DSR are not active.

Refer to Detail Data under X25 ADAPT.

X25 ALERT7

Error Description: X–7, MODEM FAILURE: ACU NOT RESPONDING

This error is logged by the X.25 device driver if the attached auto–call unit failed to establish a call.

Refer to Detail Data under X25 ADAPT.

X25 ALERT8

Error Description: X–8, X.21 NOT CONNECTED

This error is logged by the X.25 device driver when it fails to connect to an X.21 network.

Refer to Detail Data under X25 ADAPT.

X25 ALERT9

Error Description: X–9, FRAME TYPE W RECEIVED

This error is logged by the X.25 device driver if a W type FRMR frame is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT10

Error Description: X–10, FRAME TYPE X RECEIVED

This error is logged by the X.25 device driver if an X type FRMR frame is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT11

Error Description: X–11, FRAME TYPE Y RECEIVED

This error is logged by the X.25 device driver when a Y type FRMR frame is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT12

Error Description: X–12, FRAME TYPE Z RECEIVED

This error is logged by the X.25 device driver when a Z type FRMR frame is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT13

Error Description: X–13, FRAME TYPE W TRANSMITTED

This error is logged by the X.25 device driver when a W type FRMR frame is transmitted.

Refer to Detail Data under X25 ADAPT.

X25 ALERT14

Error Description: X–14, FRAME TYPE X TRANSMITTED

This error is logged by the X.25 device driver when an X type FRMR frame is transmitted.

Refer to Detail Data under X25 ADAPT.

X25 ALERT15

Error Description: X–15, FRAME TYPE Y TRANSMITTED

This error is logged by the X.25 device driver when a Y type FRMR frame is transmitted.

Refer to Detail Data under X25 ADAPT.

X25 ALERT16

Error Description: X–16, FRAME TYPE Z TRANSMITTED

This error is logged by the X.25 device driver when a Z type FRMR frame is transmitted.

Refer to Detail Data under X25 ADAPT.

X25 ALERT17

Error Description: X–17, FRAME RETRY N2 REACHED

This error is logged by the X.25 device driver when the frame level retry count (N2) is reached.

Refer to Detail Data under X25 ADAPT.

X25 ALERT18

Error Description: X–18, UNEXPECTED DISC RECEIVED

This error is logged by the X.25 device driver if a disconnect frame is received unexpectedly.

Refer to Detail Data under X25 ADAPT.

X25 ALERT19

Error Description: X–19, DM RXD DURING LINK ACTIVATION

This error is logged by the X.25 device driver when a DM frame was received after sending a SABM.

Refer to Detail Data under X25 ADAPT.

X25 ALERT21

Error Description: X–21, CLEAR INDICATION RECEIVED

This error is logged by the X.25 device driver if a clear indication packet with DCE–originated cause and diagnostics is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT22

Error Description: X–22, RESTART INDICATION RECEIVED

This error is logged by the X.25 device driver after a restart indication packet containing a DCE–originated cause and diagnostics was received by the DCE.

Refer to Detail Data under X25 ADAPT.

X25 ALERT23

Error Description: X–23, RESET REQUEST BY X.25 ADAPTER

This error is logged by the X.25 device driver if a reset request packet is sent automatically by the adapter.

Refer to Detail Data under X25 ADAPT.

X25 ALERT24

Error Description: X–24, CLEAR REQUEST BY X.25 ADAPTER

This error is logged by the X.25 device driver when the adapter code sent out a clear request with DCE–originated cause and diagnostics. The clear was not one requested by a higher layer.

Refer to Detail Data under X25 ADAPT.

X25 ALERT25

Error Description: X–25, RESTART REQUEST BY X.25 ADAPTER

This error is logged by the X.25 device driver when a restart request containing DCE–originated cause and diagnostics is transmitted.

Refer to Detail Data under X25 ADAPT.

X25 ALERT26

Error Description: X–26, TIMEOUT ON RESTART REQUEST, T20

This error is logged by the X.25 device driver if timer T20 expires before receipt of a restart confirmation packet.

Refer to Detail Data under X25 ADAPT.

X25 ALERT27

Error Description: X–27, TIMEOUT ON RESET REQUEST, T22

This error is logged by the X.25 device driver if timer T22 expires before receipt of a reset confirmation packet.

Refer to Detail Data under X25 ADAPT.

X25 ALERT28

Error Description: X–28, TIMEOUT ON CALL REQUEST, T21

This error is logged by the X.25 device driver if timer T21 expired before receipt of either a call connected or clear indication packet.

Refer to Detail Data under X25 ADAPT.

X25 ALERT29

Error Description: X–29, TIMEOUT ON CLEAR REQUEST, T23

This error is logged by the X.25 device driver if timer T23 expires before receipt of a clear confirmation packet.

Refer to Detail Data under X25 ADAPT.

X25 ALERT30

Error Description: X–30, DIAGNOSTIC PACKET RECEIVED

This error is logged by the X.25 device driver if a diagnostic packet is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT31

Error Description: X–31, RESET INDICATION PACKET RECEIVED

This error is logged by the X.25 device driver when a reset indication packet with DCE–originated cause and diagnostics is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT32

Error Description: X–32, (DCE) CLEAR INDICATION BY X.25 ADAPTER

This error is logged by the X.25 device driver when a clear indication containing DCE–generated cause and diagnostics is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT33

Error Description: X–33, (DCE) RESET INDICATION BY X.25 ADAPTER

This error is logged by the X.25 device driver if a reset indication packet containing a DCE–originated cause and diagnostics is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT34

Error Description: X–34, (DCE) RESTART INDICATION BY X.25 ADAPTER

This error is logged by the X.25 device driver after a restart indication packet containing a DCE–originated cause and diagnostics is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT35

Error Description: X–35, (DCE) RESTART REQUEST RECEIVED

This error is logged by the X.25 device driver when a restart request indication packet with DCE–originated cause and diagnostic codes is received.

Refer to Detail Data under X25 ADAPT.

X25 ALERT36

Error Description: X–36, (DCE) TIMEOUT ON RESTART INDICATION, T10

This error is logged by the X.25 device driver when timer T10 expired at the DCE before receipt of a restart confirmation packet.

Refer to Detail Data under X25 ADAPT.

X25 ALERT37

Error Description: X–37, (DCE) TIMEOUT ON RESET INDICATION, T12

This error is logged by the X.25 device driver if timer T12 expired before receipt of a reset confirmation packet.

Refer to Detail Data under X25 ADAPT.

X25 ALERT38

Error Description: X–38, (DCE) TIMEOUT ON CALL INDICATION, T11

This error is logged by the X.25 device driver if timer T11 expired before receipt of a call accept or clear request packet.

Refer to Detail Data under X25 ADAPT.

X25 ALERT39

Error Description: X–39, (DCE) TIMEOUT ON CLEAR INDICATION, T13

This error is logged by the X.25 device driver if timer T13 expired before receipt of a clear confirmation packet.

Refer to Detail Data under X25 ADAPT.

X25 CONFIG

Error Description: X.25 CONFIGURATION ERROR

This error is logged by the X.25 device driver when the adapter fails its internal configuration.

Refer to Detail Data under X25 ADAPT.

X25 IPL

Error Description: ADAPTER ERROR

This error is logged by the X.25 device driver if an adapter timed out during the reboot sequence. This is an installation–related problem and may indicate that the communication equipment was not installed correctly or that the hardware and microcode are incompatible.

Refer to Detail Data under X25 ADAPT.

X25 UCODE

Error Description: X.25 MICROCODE ERROR

This error is logged by the X.25 device driver if the adapter microcode terminates abnormally.

Refer to Detail Data under X25 ADAPT.

Miscellaneous Error Identifiers for the Error Log

CONSOLE

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by the console device driver if an unexpected error occurs when calling another module.

DETAIL DATA

Detecting Module	Name of the detecting module.
Failing Module	Name of the failing module.
Return Code	Return code from the failing module (see **sys/errno.h**).
Reason Code	Unique error indication for the console device driver.

LOST_EVENTS

Error Description: ERROR LOGGING BUFFER OVERFLOW

Probable Causes: EXCESSIVE LOGGING BY SOFTWARE PROGRAM

Install Causes: BUFFER SIZE SET TOO SMALL

Recommended Actions: INCREASE BUFFER SIZE

Failure Causes: EXCESSIVE LOGGING BY SOFTWARE PROGRAM

Recommended Actions: Identify offending software component, correct, then retry offending software component

NLS BADMAP

Error Description: SOFTWARE ERROR: NLS MAP CORRUPTER

This error occurs when the required National Language Support (NLS) map is resident but does not match data from the input stream.

DETAIL DATA

Error Code	System error code.
Rule Index	Number of the rule in the NLS rule map that failed.
Number Of Rules	Number of NLS rules that were not followed.
Map Name	NLS map name associated with a particular character set.

NLS MAP

Error Description: SOFTWARE PROGRAM ERROR

This error occurs when the required National Language Support map is not resident.

DETAIL DATA

Error Code	System error code.

OPMSG

Error Description: OPERATOR NOTIFICATION

This error is logged when the **errlogger** command is run.

DETAIL DATA

Message From errlogger Command

The text entered using the **errlogger** command.

PGSP KILL

Error Description: SOFTWARE PROGRAM ABNORMALLY TERMINATED

DETAIL DATA

Program's Paging Space

The program's paging space in 1KB blocks.

PPRINTER ERR1

Error Description: PRINTER ERROR

This error is logged by the printer device driver. The failures recorded by this error log entry are not usually fatal and should occur infrequently.

DETAIL DATA

Sense Data	The sense data consists of failure data that is analyzed by the Diagnostic Programs.
Adapter At Address	Printer control register address.
Additional Information	Information about the data being transferred and the operation being performed.
Device Address	Device address.
Generation Parameter	Pointer to internal printer structure.

RCMERR

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by the rendering context manager when an unrecoverable error is detected. These errors are not necessarily catastrophic, and operations may continue; however, the response occurring immediately after the failure may be unpredictable.

DETAIL DATA

Detecting Module	Name of the detecting module.
Failing Module	Name of the failing module.
Return Code	Return code from the failing module (see the **sys/errno.h** file).
Reason Code	Unique error indication for the console device driver.

REPLACED FRU

Error Description: REPAIR ACTION

This error is an informational message logged by the diagnostic programs that indicates that a piece of hardware has been replaced.

SRC

Error Description: SOFTWARE PROGRAM ERROR

This error is logged by the System Resource Controller daemon in the event of some abnormal condition. Abnormal conditions are divided in three areas: failing subsystems, communication failures, and other failures.

DETAIL DATA

Symptom Code	The status returned by the failed subsystem. This is the child process termination status returned in a parameter by the wait subroutine (see **sys/wait.h**).
Software Error Code	The System Resource Controller error number (see **srcerrno.h**).
Error Code	System error number (errno). If the System Resource Controller error number is SRC_ODMERR(–9090), the error code is set to an ODM error (see **odmi.h**).
Detecting Module	Name of the module detecting the error.
Failing Module	Name of the failed subsystem or other failing System Resource Controller object.

SYSLOG

Error Description: MESSAGE REDIRECTED FROM syslog

This error may be logged in the system error log by specifying *errlog* as a destination in **/etc/syslog.conf**. The **syslog** daemon redirects the specified messages to the error log.

DETAIL DATA

Syslog Message	Text of the syslog message.

Appendix D. Recovering Volume Groups

When the ODM entries for a volume group are corrupted, you may notice one of the following symptoms:

- **lsvg**, **lslv** and/or **lspv** commands fail.
- You cannot change the size of file systems.
- The system cannot find volume groups or device IDs.
- The output of `lsvg -l VGname` contains `?'s` in some fields or some fields are blank.

If the Object Data Manager (ODM) entries for a volume group are corrupted, use the shell script , on page D-2 to reinitialize the ODM entries for the rootvg volume group. By modifying the shell script, you can also reinitialize a user–defined volume group. Note that it is not necessary to reboot after running this script.

See the following to recover volume groups:

- Reinitializing the rootvg Volume Group , on page D-2
- Reinitializing a User–Defined Volume Group , on page D-3
- Restoring the Original Version of the Object Data Manager , on page D-4

Note: Note that the $$ in the lines of the shell script , on page D-2, for example:

```
cp /etc/objrepos/CuAt /etc/objrepos/CuAt.$$
```

Indicate the process ID (PID) of the shell script, creating a unique file name. If after running the script you can successfully perform the commands that had previously failed, remove the Cu*.$$ files in /etc/objrepos (where $$ is a PID).

Reinitializing the rootvg Volume Group

To reinitialize the rootvg volume group, copy the shell script to /**bin**/**rvgrecover** and run the following to make that file executable:

```
chmod +x /bin/rvgrecover
```

Then run:

```
/bin/rvgrecover
```

Use the following shell script to reinitialize the ODM entries for the rootvg volume group:

```
PV=/dev/ipldevice
VG=rootvg
    cp /etc/objrepos/CuAt /etc/objrepos/CuAt.$$
    cp /etc/objrepos/CuDep /etc/objrepos/CuDep.$$
    cp /etc/objrepos/CuDv /etc/objrepos/CuDv.$$
    cp /etc/objrepos/CuDvDr /etc/objrepos/CuDvDr.$$
    lqueryvg -Lp $PV | awk '{ print $2 }' | while read LVname; do
        odmdelete -q 'name = $LVname' -o CuAt
        odmdelete -q 'name = $LVname' -o CuDv
        odmdelete -q 'value3 = $LVname' -o CuDvDr
    done
    odmdelete -q 'name = $VG' -o CuAt
    odmdelete -q 'parent = $VG' -o CuDv
    odmdelete -q 'name = $VG' -o CuDv
    odmdelete -q 'name = $VG' -o CuDep
    odmdelete -q 'dependency = $VG' -o CuDep
    odmdelete -q 'value1 = 10' -o CuDvDr
    odmdelete -q 'value3 = $VG' -o CuDvDr
    importvg -y $VG $PV      # ignore lvaryoffvg errors
    varyonvg $VG
```

Reinitializing a User–Defined Volume Group

To reinitialize a user–defined volume group, copy the shell script , on page D-2 to a new file **/bin/uvgrecover**, edit the file as follows, and then run **chmod +x /bin/uvgrecover**.

After copying the shell script to **/bin/uvgrecover**, change the line:

```
odmdelete –q ”value1 = 10” –o CuDvDr
```

To:

```
odmdelete –q ”value1 = $VG” –o CuDvDr
```

The physical volume (PV) and volume group (VG) variables on the first two lines of the script are defaulted for the rootvg volume group. The PV is set to **/dev/ipldevice**, which is a synonym for one of the physical volumes that is a member of the rootvg volume group. For a user–defined volume group:

- Change the PV= line to be any of the disks that is a member of the volume group in question. To determine to which volume group a physical volume belongs, enter:

  ```
  lsvg ‘lqueryvg –vp hdisk# ‘ | grep GROUP
  ```

  Note: The ‘ is a grave accent, not a single quote mark. This key is usually found above the Tab key.

- Change the VG= line to be the volume group in question.

For example:

```
PV=/dev/hdisk1
VG=uservg
```

Restoring the Original Version of the Object Data Manager

The /etc/objrepos/Cu*.$$ files enable you to return to the original version of the ODM. To return to the original version of the ODM, enter:

```
cd /etc/objrepos



ls -l Cu*
```

Look for the Cu* files that have a PID at the end of their names, such as CuAt.31267. Copy the Cu*.31267 files to Cu*. For example:

```
cp CuAt.31267 CuAt
cp CuDep.31267 CuDep
cp CuDv.31267 CuDv
cp CuDvDr.31267 CuDvDr
```

Index

Symbols

Numbers

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

V

W

X

Y

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull AIX 4.3 Problem Solving Guide and Reference

Nº Reférence / Reference Nº : 86 A2 32JX 02

Daté / Dated : November 1999

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement.
Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : _____ Date : _____

SOCIETE / COMPANY : _____

ADRESSE / ADDRESS : _____

_____

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

Technical Publications Ordering Form
Bon de Commande de Documents Techniques

To order additional publications, please fill up a copy of this form and send it via mail to:
Pour commander des documents techniques, remplissez une copie de ce formulaire et envoyez-la à :

BULL ELECTRONICS ANGERS
CEDOC
ATTN / MME DUMOULIN
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

Managers / Gestionnaires :
Mrs. / Mme : **C. DUMOULIN** +33 (0) 2 41 73 76 65
Mr. / M : **L. CHERUBIN** +33 (0) 2 41 73 63 96

FAX : +33 (0) 2 41 73 60 19
E–Mail / Courrier Electronique : srv.Cedoc@franp.bull.fr

Or visit our web site at: / Ou visitez notre site web à:
`http://www-frec.bull.com` (Press Room, Technical Literature, Ordering Publications)

CEDOC Reference # Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
[_ _] : **no revision number means latest revision** / pas de numéro de révision signifie révision la plus récente					

NOM / NAME : ____________________ Date : __________

SOCIETE / COMPANY : ____________________

ADRESSE / ADDRESS : ____________________

____

PHONE / TELEPHONE : ____________________ FAX : ____________________

E–MAIL : ____________________

For Bull Subsidiaries / Pour les Filiales Bull :

Identification: ____________________

For Bull Affiliated Customers / Pour les Clients Affiliés Bull :

Customer Code / Code Client : ____________________

For Bull Internal Customers / Pour les Clients Internes Bull :

Budgetary Section / Section Budgétaire : ____________________

For Others / Pour les Autres :

Please ask your Bull representative. / Merci de demander à votre contact Bull.

PLACE BAR CODE IN LOWER LEFT CORNER

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 32JX 02

Bull

Utiliser les marques de découpe pour obtenir les étiquettes.
Use the cut marks to get the labels.

AIX
AIX 4.3 Problem Solving Guide and Reference

86 A2 32JX 02

AIX
AIX 4.3 Problem Solving Guide and Reference

86 A2 32JX 02

AIX
AIX 4.3 Problem Solving Guide and Reference

86 A2 32JX 02