Precision Architecture RISC Diagnostics Manual

HP Apollo 9000 Series 700

HEWLETT PACKARD

HP Part No. 09740-90041
Printed in U.S.A. February 1991

Edition 1
E0291

Notice

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

UNIX is a trademark of AT&T Laboratories in the USA and other countries.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Printing History

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update; the edition does not change when an update is incorporated.

First Edition ..... February 1991

List of Effective Pages

The List of Effective Pages gives the date of the current edition and of any pages changed in updates to that edition. Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. No information is incorporated into a reprinting unless it appears as a prior update.

Effective Page	Date
All	February 1991

Safety Considerations

This product and related documentation must be reviewed for familiarization with safety markings and instructions before operation.

WARNING

The WARNING sign denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in injury. Do not proceed beyond a WARNING sign until the indicated conditions are fully understood and met.

CAUTION

The CAUTION sign denotes a hazard. It calls attention to an operating procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a CAUTION sign until the indicated conditions are fully understood and met.

Preface

This manual contains information about the online diagnostics and ISL diagnostics/utilities for the HP Apollo Series 700 Precision Architecture RISC Workstation. It is intended to be used as technical support hardware documentation for Hewlett-Packard CEs, CEC Engineers, SEs, and other qualified support personnel. The procedures and software focus primarily on the hardware troubleshooting environment and require specific training for correct and safe usage.

Table of Contents

Contents

1

Online Diagnostics Overview

Introduction

The online diagnostics subsystem provides a means of testing hardware modules and devices attached to the HP Precision Architecture RISC computer system. The PA-RISC system and the diagnostic subsystem are intimately tied together for error logging, and for restricting access to other users during diagnostic testing.

The subsystem provides a common, standard user interface to all the diagnostic programs and utilities, as well as a controlling mechanism for diagnostic access to I/O devices on the system. Finally, the subsystem can control the normal I/O error logging process, allowing dynamic display of errors as they occur.

Each major hardware component or aspect of the PA-RISC system can be tested by a diagnostic. Each diagnostic is described separately in the following chapters. All of the diagnostic/utility chapters share the same format.

Introduction	Brief explanation of the purpose and nature of the diagnostic.
Defects and Enhancements	STARS Database product number for comments about the diagnostic.
Minimum Configuration	Necessary hardware and software to run the diagnostic.
Operating Instructions	Information about how to start the diagnostic.
Default Tests	Lists the tests that are automatically executed if no test sections are specified in the **RUN** command.
RUN Command	Sample **RUN** command using the typical loaded system configuration described in this chapter.
Test Execution	What happens after the diagnostic is started.
Test Section Descriptions	What is being tested by each section of the diagnostic.
Commands	Commands available during the diagnostic (if any).
Error and Warning Messages	Lists all error and warning messages displayed by the diagnostic along with a probable cause and suggested action statements.

Operating Requirements

In order to support the online diagnostics subsystem, a PA-RISC computer system must be up and running. User access to at least one functioning terminal is also required.

Online Diagnostic Subsystem Components

The online diagnostics subsystem is composed of the the Diagnostic User Interface (DUI) and the diagnostic programs which can be run using the operating system.

Diagnostic User Interface

The DUI provides the communication link between the user and the diagnostic system. The DUI provides such functions as sending messages to the user from diagnostic programs, and returning replies to the appropriate section(s) of the online diagnostic software.

Diagnostic Programs

The diagnostic programs are a comprehensive set of software to test the devices and components supported on the HP Precision Architecture RISC computer family.

Diagnostic programs are divided into three groups: diagnostics and verifiers; utilities and tools; and system exercisers. Diagnostics are programs that can determine which field replaceable units (FRUs) are malfunctioning.

Verifiers cannot isolate defective FRUs, but can verify which functions of the device are operating properly. Verifiers can determine probable cause of device failures or aid the user in making such determinations. Some diagnostics and verifiers provide thorough tests of the internal I/O modules as well as complete functional tests and system type tests for peripheral devices.

Utilities and tools provide a means for obtaining system information or performing specific I/O operations. System exercisers provide a means of using (loading) a particular part of the system. These programs provide a way of using system resources under stress conditions that equal or exceed those expected under maximum load.

External excercisers are interactive programs provided for some diagnostics to provide the user with access to the set of internal diagnostics and utilities within a particular device.

The next section describes the diagnostics available on the system.

Available Diagnostics

The following diagnostic programs and utilities, along with the Diagnostic User Interface (DUI) and any background processes, are currently supported on the system:

Diagnostics

- Memory Diagnostic (MEMDIAG)
- LAN Diagnostic (LANDAD)
- SCSI CD Diagnostic (SCSICD)
- SCSI Digital Data Storage Tape Drive Diagnostic (SCSIDDS)

Online Diagnostics Subsystem Operating Software

- Diagnostic User Interface (DUI)

DUI Modes

The diagnostic system provides three modes of operation for each diagnostic program: disruptive mode, destructive mode, and normal mode. The diagnostic system determines the mode in which each diagnostic program is allowed to run by considering such things as the device being tested, and the user mode in which the system is running. When the diagnostic program requests access to a device, either at program initiation or at some other time, it is told which mode in which to run via a device control procedure.

In general, the diagnostic is usually granted destructive mode unless the selected device is a system disk or exclusive access to the device cannot be obtained for the diagnostic. The diagnostic program must decide which tests can be run in the mode it was given.

Disruptive Mode	In disruptive mode, the program can run tests of a "disruptive" nature on the selected device. A disruptive test does not destroy any data on the device, but could cause errors for other users on the system. For example, the internal selftest on a system disk is disruptive, since the disk temporarily goes offline to perform the test, causing errors for others who try to access the disk at the same time.
Destructive Mode	In destructive mode, the program may run any test on the selected device. This mode is required for tests that have the potential for corrupting data on the device being tested. There are virtually no restrictions on tests run in this mode and, therefore, this mode is handled with extreme care by the diagnostic program. An example of a destructive test is one that reformats the media on a system disk, thus destroying all of the data on it.
Normal Mode	In normal mode, the diagnostic program cannot run any tests on the selected device that are potentially destructive or disruptive in nature.

User Modes

There are three user modes available: Single User Mode, Multi-User Mode (normal state), and Single Disk Mode.

Single User	Can be selected by a user with the required capability. The primary purpose of Single User Mode is for testing that may cause data integrity problems. Typically, it is used only in the event of a major problem with the system hardware.
Multi-User Mode	Can be selected by a user with the required capability.
Single Disk Mode	Is selectable only on system boot-up and is system specific.

Security

Four levels of security are available for users in the online diagnostics subsystem. Access to the various programs is restricted by security level. In addition, each program may restrict certain functionality to users of various security levels.

Level 0	The highest security; the user may install, remove, or update programs through a utility program and may do anything that a user at level 1 may do.
Level 1	The user may perform destructive tests, read or modify data on any device, may enter SUM or MUM modes, and do anything that a user at level 2 may do.
Level 2	The user may perform disruptive tests, but may not display or modify user data, and may do anything that a user at level 3 may do.
Level 3	The user may run non-disruptive tests only.

The following table lists the user capabilities required for each security level:

Security	
Level 0	Superuser; Configurable
Level 1	Configurable
Level 2	Configurable
Level 3	Users not in /usr/diag/security

Contents

2

Diagnostic User Interface

Introduction

The Diagnostic User Interface (DUI) enables users to run on-line diagnostic programs on PA-RISC systems through a common, consistent interface. Users need only learn one set of commands to run, process, and manipulate all on-line diagnostic programs.

Numerous commands are available to start, stop, monitor, add, and delete diagnostic programs.

Various utility functions are also provided through the DUI - the output of a diagnostic may be redirected to a file and/or echoed to a hardcopy device; the inputs to a diagnostic program may be obtained from a file rather than from the user's terminal; part of a diagnostic session may be controlled from a command file. The complete set of all such functions will be found in the "COMMANDS" section of this document.

All of the commands accessible through the DUI are invoked identically on all systems on which the online diagnostics subsystem runs. Each individual command has the identical effect on every system on which the online diagnostics subsystem runs.

Diagnostic programs which run under the DUI include:

- programs which test peripheral devices and individual circuit boards and attempt to isolate a failure to a FRU (Field Replaceable Unit)
- programs which load (or stress) a system in order to recreate the conditions under which a system might exhibit failures
- programs which can isolate a "functional" failure, such as a read or a write failure, even if they cannot pinpoint the hardware involved in the failure
- programs which perform utility functions useful to a diagnostician such as creating a map of the system configuration or displaying the contents of log files.

The diagnostic programs available through the DUI may vary from system to system, but all diagnostic programs and associated commands are invoked identically no matter what system they run on.

Note

This document discusses the commands

- DEFAULT
- FOREGROUND
- REPLY
- SET
- SHOWDEFAULT
- SHOWPARMS

and refers to a programmatic interface and the ability to run multiple diagnostics using just one **run** command. None of these features is implemented in the first release. If one of the commands listed above is invoked, nothing will be done. The **DUI** prompt will reappear immediately after the command is given.

If more than one diagnostic is named within the **run** command, an error will be printed, and the **DUI** prompt will reappear. The user may then re-enter the command.

Defects and Enhancements

Submit defect reports and enhancement requests concerning this diagnostic through the STARS database referencing product number 30600-10023.

Invoking the DUI

To enter the online diagnostics subsystem, one enters the command "SYSDIAG" at the system's prompt. Upon entering the DUI, the following banner will be displayed:

```
**********************************************************************
******                                                          ******
******              ONLINE DIAGNOSTIC SYSTEM                    ******
******                                                          ******
****** (C) Copyright Hewlett Packard Co. 1987, 1989            ******
******                All Rights Reserved                       ******
******                                                          ******
******             DUI Version: A.02.03                         ******
******        Diagnostic Monitor Version   A.02.02              ******
******                                                          ******
**********************************************************************

     Type "HELP" for assistance.

WARNING:   This Diagnostic System has been designed as a tool for
Hewlett-Packard service personnel only.  Incorrect use of this
System could result in loss of data and additional Hewlett-Packard
support charges.  Hewlett-Packard will not be responsible for
consequences resulting from unauthorized use of the Diagnostic System.

DUI >
```

The user is now free to enter diagnostic system commands. To exit the DUI, the user simply types EXIT.

Version Identification

The DUI's version number is displayed in the banner. The format of the version is:

```
VERSION.UPDATE.FIX
```

Where *version* refers to major changes in large groups of system software, each *update* indicates a major change or addition to the Diagnostic User Interface, and a *fix* indicates bug fixes within the DUI.

Entering Commands

Replies and Responses

The user may be prompted from time to time for certain data. If one of several specific responses, such as **yes** or no, is required, the valid responses will be indicated in parenthesis. The default response - the assumed response if nothing is given but a carriage return - will be indicated in square brackets ([]). For example, if the user wants to exit and there are still running processes, he might be asked the question:

```
Do you wish to abort the currently running processes (Y/N) [N] ?
```

Valid responses include **Y**, **N**, **yes**, and no. Simply hitting the <RETURN> key selects the default response of **N**.

Continuation Lines

If a command is too long to fit on one line, the user may continue the command on subsequent lines by using the continuation character (\). This character causes the DUI to delay interpretation of the command. Usually, interpretation would begin immediately after the receipt of the carriage return. When the continuation character is given at the end of a command line (followed by a carriage return), the DUI prompt will appear on the next line - the user may then continue to type the command. A command may consist of numerous continuation lines provided that the total length of the text entered does not exceed 255 characters. For example:

```
DUI > run wizbang errcount=9 \
DUI >> ldev3  erronly
```

Note that the DUI prompt is slightly different to indicate that a continuation of the previous line is expected.

Command Comments

Command lines may include comments if desired. The beginning of a comment is indicated by the # character; the end-of-line automatically terminates the comment.

```
DUI > list              # This is a comment

DUI> # This is a
DUI> #multi-line comment .
```

Installation Instructions

The DUI will be shipped to customers already installed in all releases; no user activity will be required. The set of diagnostic programs will also be installed in all releases prior to being shipped to customers. If a diagnostic program must be installed into the On-Line Diagnostic system at a customer's site the **install** command may be used at the DUI prompt.

Input and Output Files

The DUI works with *usefile* files, *infile* files, and *outfile* files. (See the USEFILE and OUTFILE commands and the INFILE and OUTFILE run command modifiers.) *Usefile* files and *infile* files are assumed to be comprised of lines 80 (ASCII) characters in length. *Outfile* files, which are made by the DUI, will also consist of 80 character length lines.

Usefiles, *infiles*, and *outfiles* may reside in any directory as long as the user has the appropriate capability to access them. The complete path name may be specified for all files and must be specified if the file resides somewhere other than in the directory the user is running the DUI from.

The form of complete path names differ among operating systems. For example, on MPE XL a complete path name for **somefile** might be **somefile.mygroup.official** or just **somefile.mygroup**, while on HP-UX **/user/mydir/somefile** might be appropriate. It is assumed that the user is familiar enough with the file system organization in the operating system being used to be able to specify the appropriate path.

If the file name provided for an *outfile* already exists, it will not be overwritten, rather an error message will be printed.

Interactive Interface

The normal mode of operation of the DUI is interactive; however, interaction may be mimicked through a *usefile* (a file containing commands to be executed along with information the user would normally give interactively during the execution of those commands). Data which would usually be sent to a diagnostic by typing the information in at the user's terminal may also be entered automatically if an *infile* is specified. Conversely, information which would usually be sent directly from the diagnostic system to the user's terminal may be redirected by using an *outfile* or by giving the **hardcopy** command or modifier.

Usefile, *infile*, *outfile*, and **hardcopy** are more fully explained in their own sections of this document.

Programmatic Interface

A simple programmatic interface to the DUI exists. A program may invoke the DUI by using whatever construct is provided in the operating system being used to execute a command interpreter or shell command from within a program. The command to be executed is SYSDIAG. The program may send a one line command, in the form of an information string, to the DUI.

The one line may be any command the DUI would normally accept, including the **usefile** command. The DUI will terminate immediately after executing the one line command.

For example, on MPE XL the intrinsic **create** or **createprocess** could be used while on HP-UX one of the **exec** system calls would be used.

Interrupts

The effects of typing a user interrupt while in the DUI depend upon what action is taking place. Interrupts will only affect diagnostic programs and processes which are running in the foreground. Background processes cannot be interrupted.

- If no program is running in the foreground and no USEFILE file is being used
  - "*** INTERRUPT ***" will be printed on the standard output device
  - The DUI prompt will reappear
  - Nothing else will be done.
- If executing a USEFILE
  - "*** INTERRUPT ***" will be printed on the standard output device
  - Reading of the *usefile* is stopped and the *usefile* is closed. All *usefiles* nested with the current *usefile* are also closed.
  - If a program is running in the foreground it will either receive a message that an interrupt has occurred or will be suspended as explained below.
- If a program has requested control of its own interrupts
  - The program will be informed of the interrupt. What then occurs is program dependent.
- If a program is running in the foreground and has not requested control of its own interrupts
  - "*** INTERRUPT ***" will be printed on the standard output device
  - The program will be SUSPENDED.
  - The DUI prompt will reappear

The control characters used to generate a user interrupt are operating system dependent. On MPE XL (CTRL-Y) works; on HP-UX (CTRL-C).

Security

Many diagnostic tests have the indirect effect of corrupting data (e.g., selftests which write into onboard registers). To ensure system and user data integrity, the online diagnostics subsystem has its own security mechanisms which augment the normal operating system security. The online diagnostics subsystem also uses its own security mechanisms to ensure diagnostic system integrity.

Commands given to the `DUI` will not be executed if the caller has too low a security level. Security level is sometimes called capability level - the two are synonymous.

The online diagnostics subsystem maps the user's operating system security into one of four diagnostic system security levels:

1. diagnostic security level 0
2. diagnostic security level 1
3. diagnostic security level 2
4. diagnostic security level 3

Diagnostic security level 0 is the highest; level 3 is the lowest.

The correspondence between the user's operating system security level and the user's diagnostic security level is operating system dependent. This is necessary because the different operating systems vary in the way different classes of security are defined and in the kinds of security granted different users. However, some general statements may be made as to the diagnostic security levels various users would be given.

An HP-UX superuser (root) or an MPE XL system manager would be assigned the highest diagnostic security level, level 0, and could perform any action the diagnostic system was capable of performing. A user who had only the lowest level of security available on an operating system would be assigned the lowest diagnostic security level, level 3, and could perform only those actions which were non-destructive and which did not require exclusive access to a device. This would include, in most cases, such actions as running diagnostic sections which copied and decoded the *identify* block from a hardware board.

Users whose operating system security fell somewhere between the lowest and highest would be assigned diagnostic security level 1 or level 2 and could perform such actions as sending loopbacks to devices. But again, the diagnostic security level assigned to a user and hence the actual tests and actions which a particular user could perform are operating system dependent.

Users on MPE XL with operating system capability `sm` are given diagnostic security level 0. Those with `di` capability are given diagnostic security level 1. Those with `op` capability are given diagnostic security level 2. All other users are assigned diagnostic security level 3.

An HP-UX user is assigned a diagnostic security level based on whether or not the user has an entry in the diagnostic SECURITY file. Each entry in this file consists of a user's login name, along with a number. Those users whose names are followed by the number "0", such as root, are given diagnostic security level 0. Those users whose names are followed by "1" are given diagnostic security level 1. Those whose names are followed by "2" or "3" are given diagnostic security level 2 or 3, respectively. Thus, a SECURITY file might look like this:

```
root:0
jdoe:2
jroe:0
tsmith:1
```

If the SECURITY file cannot be found, or the user's login name cannot be found in it, the user is assigned diagnostic security level 3.

The HP-UX diagnostic SECURITY file is maintained by the system administrator, and can only be modified by a superuser. The SECURITY file is located in the same directory as the other major diagnostic files, normally `/usr/diag/bin`. Please see the system administrator if a new entry is needed in this file.

Every action which might be performed through the online diagnostics subsystem is associated with one of four diagnostic security states.

These states are:

1. non-destructive/non-exclusive
2. non-destructive/exclusive
3. destructive/non-exclusive
4. destructive/exclusive

Non-destructive and *destructive* (the two possible test modes) indicate the possibility that an action could destroy data. *Non-exclusive* and *exclusive* (the two possible access modes) refer to whether or not a device must be accessed exclusively (locking all other processes out during the time of use) to perform an action.

Many actions do not directly involve devices so, for those actions, the access mode is moot. For example, purging a diagnostic using the PURGE command does not directly involve a device but has a drastic effect on the diagnostic system; the DUI would only allow a user whose diagnostic system security level was such that he could run destructive tests to execute the PURGE command.

The online diagnostics subsystem checks the user's diagnostic security level, whether the device being tested is being accessed exclusively or non-exclusively (access mode), and the level of potential destructiveness (test mode) of the requested tests or the given commands before allowing any particular set of tests to run or commands to execute. The state of the operating system also affects which commands and tests a user can run through the DUI.

For example, a user whose diagnostic security level was such that he would normally not be allowed to perform a *destructive* action on a *non-exclusive* device would be allowed to do so if the operating system had been booted in single-user mode.

If a user attempts to run a test or execute a command which requires a higher security level than he possesses, a message will be printed explaining why the action could not be performed.

Notation and Special Symbols

The following notation is used in the command and command modifier syntax diagrams. The notation is also used in messages printed in response to the help command.

[] An element inside brackets is *optional*. Several elements stacked inside brackets means the user may select any one or none of these elements. For example:

```
[A]
[B]
[C]  User may select A or B or C or none.
```

When brackets are nested, parameters in inner brackets may be specified only if parameters in outer brackets are specified.

For example:

[parm1 [parm2 [parm3]]]

may be entered as

parm1 parm2 parm3

or

parm1

or

parm1 parm2

Optional parameters which are not positional are shown as follows:

[parm1][parm2]

[,...] or **[...]** Means that the immediately preceding item in the syntax diagram may be repeated any number of times. The [,...] form means that each instance of the repeated item must be preceded by a comma.

{} An element inside curly braces is *required*. When several elements are stacked within braces in a syntax statement, the user must select one of those elements. For example:

```
                {A}
                {B}
                {C}  User must select either A or B or C.

| |           Vertical parallel lines indicate that any or
              none of the options may be used in any sequence.
              However, the elements may not be repeated. For
              example:

                |A|
                |B|
                |C|  The user may choose A, B, and C; or C and A,
                     or B alone, etc.

UPPERCASE     Represent literals which are to be entered
              exactly as shown except that they may be
              entered in lower case. Also, if an abbreviation
              or alternate token is listed for the item, a
              substitution is allowed. Lastly, only the first
              n characters which establish uniquely what
              the literal is need be entered.

Special Character Literals

              The special characters  +   -   /   (   )   "   =
              are literals to be entered exactly as shown in the
              syntax diagrams.

italics       Items printed in italics are to be
              replaced with user supplied information.
```

Delimiters and Abbreviations

All keywords and options may be abbreviated to the shortest number of characters which make the token unique. Delimiters for keywords and options include spaces and/or semicolons (;). Other delimiters may be specified in the descriptions and syntax diagrams for individual commands and command modifiers.

UPPER and lower case command lines are equivalent.

User Set Default Values

The user may specify default values which differ from the normal diagnostic system default values for command modifiers using the set command.

The user may set a "universal" default value for a modifier - that is, a value which will have effect at all times - or a default value which will take effect only when a particular diagnostic is run. In either case, if another value for the modifier is given in a run command, that value will have effect only until the diagnostic named in the run command completes execution.

Modifier default values which have been changed, either universally or for individual diagnostics, using the **set** command can be reset to their diagnostic system default values using the **default** command.

For more information, please see the **set** command and the **default** command descriptions in this document.

Designating Devices to be Tested

Usually, when a diagnostic program is invoked the device to be tested must be named. This can be done in one of two ways: either by using the logical name of the device or giving the physical path address of the device. That is, one of the two command modifiers **ldev** or **pdev** must be given with the **run** command, or must be set using the **set** command.

The form these strings take varies among the operating systems and even from one HPPA machine to another.

For example, under the MPE XL operating system the logical name (**ldev**) of a device is a number (e.g., 3) while in the HP-UX operating system the logical name of a device is that of a special device file (e.g., **dsk/c0d0**).

The physical path address (**pdev**) of a device is a series of numbers separated by various punctuation marks. Each number in the series corresponds to a physical connection along the electrical pathway to the device. For example, an HPIB device adapter might have an address of 4.2 on an HP3000 series 930, but an address of 2/8.0 on an HP9000 series 850.

It is assumed that the user of the diagnostic system knows enough about the operating system and machine being worked with to determine the correct **ldev** or **pdev** for the device being tested. A utility program, **SYSMAP**, is provided in some diagnostic installations and may be run through the **DUI** (**SYSMAP** may NOT be available in all installations) to help determine the desired **ldev** or **pdev**.

Running Multiple Diagnostics

Several diagnostic programs may be invoked simultaneously using the **run** command. To do this, the user simply names all of the diagnostics to be run along with the command modifiers for each in a **run** command.

Examples:

```
DUI> run xdiag ldev 0 loop 5 erronly ydiag pdev 4.3.2 \
DUI>> errpause background outfile yout

DUI> qdiag ldev 3 infile qinf & rdiag ldev 0 \
DUI>> sc 4/5(20,23)
```

All but one of the programs must be run in the background; only one program may run in the foreground at any one time.

It is suggested that the output of programs run in the background be redirected to an *outfile*. The **DUI** will not impose any order on the messages received from multiple diagnostic programs but will output each message as it is received.

When multiple diagnostic programs are invoked, the **DUI** will do all the internal checks and initial set up it would usually do for each of the diagnostics. THEN it will launch the diagnostics one immediately after the other using the mechanisms provided by the operating

system. The DUI will not wait for one of the diagnostics to complete before launching the next.

The number of diagnostics which may be run simultaneously is dependent on the number of processes the operating system will allow any one user to run simultaneously - the diagnostic system itself does not impose a limit.

Command Summary

The following is a list and brief description of each of the commands available in the DUI.

Command Name	Description
ABORT	Terminates active diagnostic program or utility.
CI	Provides access to the operating system command interpreter.
CODETEST	For HP internal use only.
DEFAULT	Resets command modifiers to their original diagnostic system default values.
DIAGSYSTEM	Used to access and manipulate internal diagnostic system processes which are usually not accessible to the user.
DO	Allows the user to re-execute any command in the command line history stack. It also permits the user to edit the command before re-executing it.
EXIT	Terminates the DUI and returns control to the operating system.
FOREGROUND	Moves a diagnostic which has been running in the background into the foreground.
HARDCOPY	Causes all terminal input/output to be echoed to the system printer.
HELP	Accesses HELP facility for information about the DUI and its commands or for information about any of the diagnostic programs.
INSTALL	Allows the user to install diagnostic programs.
LIST	Provides information about any or all of the programs in the diagnostic system.
LISTREDO	Displays the user's command history stack.
MODE	Displays and/or alters the current operating system mode.
MODIFY	Allows a user to change information about a diagnostic program without needing to reinstall the diagnostic.
OUTFILE	Causes all diagnostic system input and output to be written into the specified file.
PURGE	Removes programs from the diagnostic library.
REDO	Allows user to display (for command editing) and re-execute any command in the command line history stack.
REDOLOAD	Replaces the user's command history stack with the history stack which was saved by the command REDOSAVE.
REDOSAVE	Causes all or part of the user's command history stack to be saved into a file.

REDOSIZE	Allows the user to set the maximum number of commands which will be saved in the user's command history stack.
REPLY	Used to send a reply to the prompt of a diagnostic program which is running in the background.
RESUME	Resumes processing of a suspended program.
RUN	Loads and executes a specified program.
SETVAR	Allows the user to explicitly set the values of environmental variables which control various features of the user interface.
SET	Allows the user to explicitly reset the system default values for modifiers which may be given on a run command line.
SHOWACTIVE	Lists programs which are currently active for the user.
SHOWDEFAULT	Causes a list of command modifiers and their default values to be displayed.
SHOWPARMS	Causes a list of user settable modifiers and their current values to be displayed.
SHOWSTATE	Will cause the current system mode (single-user or multi-user) and the user's security capability to be displayed.
SUSPEND	Suspends processing of a specified program.
UNLOCK	Releases specified device from lock status.
USEFILE	Causes input to the diagnostic system to be input from the specified file rather than from a user's terminal.

The following pages provide a detailed description of each of the available DUI commands. The description includes information about syntax, options available for each command, limitations of each command, and examples.

ABORT

The abort command terminates a program.

The chosen diagnostic program will be aborted regardless of what it is doing. The diagnostic system may perform clean up actions on any devices allocated to the program but this is not guaranteed. If more than one program is running or in a suspended state, the *program-id* modifier or ALL modifier should be given. If only one program is running or suspended when the ABORT command is given no modifier is necessary.

Syntax:

```
Abort [ALL                    ] [UNCONDITIONAL]
      [program-id [[,]... ]] [UNCONDITIONAL]
```

Modifiers:

program-id	The process identifier number of the diagnostic program to be aborted. If there is more than one active program and the *program-id* is not specified, a list of *program-ids* from which to choose will appear.
ALL	Abort all programs running under the diagnostic system. This will cause all programs running under the DUI through which the abort command was received to terminate.
UNCONDITIONAL	Has no effect in online diagnostics subsystem; all aborts are unconditional.

Command Examples:

```
DUI> abort 23

DUI> abort

DUI> abort 4 32
```

CI

CI (command interpreter) invokes the system's command interpreter or shell so that one or more operating system commands may be executed.

If the CI command is given without a modifier, the command interpreter prompt, which is system specific, will appear. Command interpreter commands can then be given until the user specifically exits back to the DUI. The command used to exit back to the DUI is system specific - **exit** on MPE XL and on HP-UX systems.

If the CI command is given with a *command* that one command will be executed and the DUI prompt will reappear.

Syntax:

```
CI [command]
```

Abbreviations and Alternative Tokens:

```
!
:
```

Modifiers:

command The command (and its arguments) which is to be executed by the system's command interpreter or shell.

Command Examples:

```
DUI> ci
%                    {the command interpreter's prompt}

DUI> : listf

DUI> !ls
```

Limitations:

Warning

If the operating system command to exit a session or to begin a new session is given (e.g., logout on HP-UX or bye on MPE XL), the DUI will be terminated. No clean up will be done. This is especially dangerous if the system was placed into single-user mode through the DUI (see MODE command). The system may need to be rebooted before anyone, including the console operator, will be able to log on.

CODETEST

This command is provided for Hewlett-Packard internal use only.

DEFAULT

DEFAULT resets command modifiers to their original diagnostic system default values. (The modifiers could have had their values reset by the **set** command.)

DEFAULT will have no effect if it is used with a modifier which still has its original diagnostic system default value. No error message or warning will be printed.

If the command is given without an argument the effect is the same as if the user typed **DEFAULT ALL**.

Syntax:

```
DEFAULT  [command modifier [[,]...]]
         [ALL                      ]
```

Modifiers:

command modifier One of the following

```
                        BACKGROUND
                        DEBUG
                        ERRCOUNT
                        ERRONLY
                        ERRPAUSE
                        ERRPRINT
                        HARDCOPY
                        INFILE
                        LDEV
                        LOOP
                        OUTFILE
                        PDEV
                        SECTIONS
                        TRACE
                        Diagnostic Specific Parameters

                Please note that this list includes all
                run command modifiers except SECURITY.

                The "Diagnostic Specific Parameters" can be
                reset to null - the diagnostic system default -
                by giving a pair of double quotes ("") as a
                modifier to DEFAULT.

ALL             Resets all modifiers, including the "Diagnostic
                Specific Parameters" modifier to their diagnostic
                system default values.
```

Command Examples:

```
DUI> default loop,"",trace,hardcopy

DUI> default all
```

Related Commands:

```
SET
SHOWDEFAULT
SHOWPARMS
```

DIAGSYSTEM

DIAGSYSTEM is used to access and manipulate internal diagnostic system processes which are usually not accessible to the user. **DIAGSYSTEM** may be used to get a listing of running diagnostic system processes along with their program identifiers; abort diagnostic system processes; and launch diagnostic system processes. These processes run outside of the user interface and their execution cannot be altered by the user. For example, some diagnostic system software is sold by HP as an add-on product - **DIAGSYSTEM** allows this software to be added to a customer's system without needing to do an **xinstall** and **reboot**. This command also obviates the need to reboot the system to launch diagnostic system processes. Examples of such processes would be diagnostic logging processes and diagnostic statistical analysis processes.

Syntax:

```
DIAGSYSTEM {SHOWACTIVE    }
           {ABORT prog_id}
           {RUN prog_name}
```

Abbreviatons and Alternate Tokens:

```
DS
```

Modifiers:

SHOWACTIVE	Display a list of running diagnostic system processes along with their program identifiers.
ABORT *prog_id*	Terminate the diagnostic system process specified by the given program identifier.
RUN *prog_name*	Launch the diagnostic system process specified by *prog_name*.

Command Examples:

```
DUI> diagsystem showactive

DUI> ds run memlogp

DUI> diagsystem abort 33
```

Limitations:

Only a user with the highest diagnostic security level (diagnostic security level 0) will be allowed to abort a diagnostic system process.

DO

DO allows the user to re-execute any command in the command line history stack. It also permits the user to edit the command before re-executing it. The (edited) command is executed immediately after the carriage-return; no interactive editing may occur. (The REDO command must be used if interactive editing is desired.)

Syntax:

```
DO [[CMD=]command-id][, edit-directives      ]
                     [;EDIT= edit-directives]
```

Modifiers:

command-id Specifies the command to re-execute. The command may be specified by its relative or absolute order in the command line history stack, or by name (as a string) in whole or in part. The default *command-id* is -1, the most recent command. An error is detected if the *command-id* does not exist in the command line history stack.

COMMAND-ID	EXECUTES
(omitted)	The most recent command (same as DO -1).
-n	nth command before the most recent one. N is a number in the command line stack relative to the most recent command, which is -1.
m	Command number m in the command line stack. The number m is absolute (not relative).
string	The most recent command beginning with this string.

edit-directives String specifying the changes to be made in the command represented by the *command-id* before its re-execution. If no *edit-directives* are given, the command is re-executed immediately - no editing is performed.

Note

Both *command-id* and *edit-directives* must be surrounded by quote marks ("or ') if they contain any delimiters such as: , ; " ' [ ] = or a space.

When an editing command such as replace, insert, delete, or change is to take effect anywhere other than in column 1 of the original command, the position (in the original command string) where the edit should begin can be designated by preceding the edit command with spaces.

Example:

```
: diuspend    mistyped command
: do ,drs     delete the 'd' in col. 1 and replace
                the 'i' with an 's'
: suspend     result of edit


: suspllnd          mistyped command
: do ,"     ddie"   delete the fifth and sixth characters
                      and insert an 'e'
: suspend           result of edit
```

The editing directives which may be used as *edit-directives* are:

DIRECTIVE	EFFECT	
i	INSERT.	If text follows the i, the text following i is inserted in the current line at the position after the i.
r	REPLACE.	If text follows the r, the text following r replaces the same number of characters in the current line, beginning at the position of r.
d	DELETE.	Deletes a character from the current line for each d specified in the edit line. Note that "d d" does not specify a range, but simply deletes one character from the position above each d. Multiple d's may be followed by an insert or replace operation.
d>	DELETE.	Deletes to the end of the current line from the position specified by d>. It may be followed by an INSERT or REPLACE operation.
>	APPEND.	> followed by text appends the text to the end of the current line. If > is positioned beyond the end of the current line, then a replacement is performed instead.
>d	DELETE.	Deletes from the end of the current line, right-to-left. Multiple d's may be specified after >, as well as INSERT and REPLACE strings.
>r	REPLACE.	Replaces characters at the end of the command line. The replacement is done so that the last (rightmost) character of the replacement string is at the end of the line.

c CHANGE. Changes all occurrences of one string into another in the current line when the searched for string and the replacement string are properly delimited. A proper delimiter is a non-alphabetic character: ', ", /, etc.

The substitution is specified as:

c<delim>*search-string*<delim>
[*replace-string*[<delim>]]

Omitting the *replace-string* causes occurrences of the *search-string* to be deleted.

other Simple replacement. Any other character (not i, r, d, d>, >, >d, c, or u) causes that character to be replaced in the current line at the position indicated by the character. In fact, simple replacement also occurs for the editing characters i, r, c, or > if they are not followed by text; or if > appears at or beyond the current end of line.

Command Examples:

EDITING SAMPLES

Practical uses of the editing commands listed above are shown here:

EDIT	ACTION
rxyz	Replaces the current text with xyz starting at the position of r.
xyz	Replaces the current text with xyz starting at the position of x.
ixyz	Inserts xyz into the current line, starting at the position immediately before the i.
ddd	Deletes three characters, one above each d.
"d xyz"	Deletes a single character above the d, skips one space, then replaces the current text with xyz starting at the position of x.
ddixyz	Deletes two characters, then inserts xyz in the current line in the position before the i.

'd d'	Deletes one character above the first d, skips two spaces and deletes a second character above the second d. It does not delete a range of characters.
'd d>xyz'	Deletes a single character above the first d, skips two spaces and deletes to the end of the line beginning at the second d, and then appends xyz to the end of line.
>xyz	Appends xyz to the end of the current line.
>ddxyz	Deletes the last two characters from the end of the current line and then appends xyz to the end of the line.
>rxyz	Replaces the last three characters in the current line with xyz.
>ixyz	Appends xyz to the end of the line. In this case, the i command is superfluous, because > accomplishes the same result. Using >xyz would be sufficient.
c/ab/def	Changes all occurrences of ab to def, starting at c.
c"ab"	Deletes all occurrences of "ab" starting at c.
cxyz	Replace the current text with cxyz, starting at c. Because delimiters have not been specified (as they were in the previous two examples), this is a simple replacement.

EXAMPLES

DO pas	Re-executes the the most recent command beginning with the string pas.
DO 10	Re-executes command number 10 (absolute) on the command history stack.
DO -2	Re-executes the second-to-last command on the stack (one command before the most recent).

`DO -2, c/5a/5b`	Change all occurrences of 5a to 5b in the command preceding the most recent one before re-executing it. The default is -1.
`do ,c/5a/5b`	Change all occurrences of "5a" to "5b" in the most recent command before re-executing it.
`DO run, ">;debug"`	Append ;debug to the the most recent :RUN command and then re-execute it.
`DO 'RUN MYP', '>;LIB=G'`	Find the most recent command beginning with RUN MYP and append ;LIB=G before re-executing it.

Related Commands:

REDO
LISTREDO
REDOLOAD
REDOSAVE
REDOSIZE

Limitations:

- DO is based on an MPE XL command.

EXIT

EXIT causes the DUI to terminate. The system prompt will then appear. If any diagnostic programs are running or suspended when the command is received the user will be asked if he wishes to abort them - if the answer is no the EXIT will not be processed; the user will remain in the DUI.

An exception to this will occur if either the QUIET or the UNCONDITIONAL modifiers are used. If either or both of these modifiers are used all diagnostic programs existing under the DUI (suspended, running in background, etc.) will be aborted automatically - the user will not be queried.

Syntax:

```
EXIT |QUIET         |
     |UNCONDITIONAL|
```

Abbreviations and Alternate Tokens:

E

Modifiers:

QUIET	Causes all diagnostics running under the DUI to be aborted before the DUI is exited. The user is not queried to confirm that a process should be aborted.
UNCONDITIONAL	Has the same effect as the QUIET modifier except that all programs running under the DUI will be aborted even if it means leaving the system in an unusual state; normal clean up will not necessarily be done. For example: devices under test might remain in a locked state; tables within the diagnostic system might not be updated to reflect the actual state of the system.

Command Examples:

```
DUI> exit

DUI> exit unconditional

DUI> exit quiet

DUI> exit unconditional quiet
```

naming both modifiers is not an error although only the "unconditional" will really have an effect

Limitations:

- The UNCONDITIONAL modifier may leave the diagnostic system and the I/O system in an unknown state: there will not necessarily be any graceful way to recover from this. The effects of this modifier may not be consistent across different operating systems nor even consistent from one session to another on the same machine.

FOREGROUND

The **FOREGROUND** command moves a diagnostic which has been running in the background to the foreground. If a program identifier (a process id number) is not specified and only one diagnostic program is running in the background, that program will automatically be placed in the foreground.

If more than one process is running in the background, but the user does not specify which is to be brought to the foreground, the user will be given a list of the processes running in the background from which to choose.

Syntax:

FOREGROUND [*program-id*]

Abbreviations and Alternate Tokens:

FG

Modifiers:

program-id The process identifier of the diagnostic program which is to be brought into the foreground.

Command Examples:

```
DUI> fg

DUI> foreground 17
```

Related Commands:

RUN (with BACKGROUND modifier)

Limitations:

- Only one diagnostic program may run in the foreground at any time.
- Once a program is running in the foreground it cannot be placed in the background again: it must run to completion in the foreground.

HARDCOPY

The HARDCOPY command causes all further input to and output from the diagnostic system to be printed by a hardcopy device. Input and output will also continue to appear on the standard output device.

This will continue until HARDCOPY is turned off or the diagnostic system is exited.

Syntax:

```
HARDCOPY [LDEV [=] logical device name [ENV [=] environment]]
         [PDEV [=] physical path       [ENV [=] environment]]
         [OFF                                               ]
         [ON                                                ]
```

Abbreviations and Alternate Tokens:

HC

Modifiers:

LDEV *logical device name*
: The logical name of the hardcopy device which is to be used.

PDEV *physical path*
: The physical path address of the hardcopy device to be used

ENV *environment*
: Information which may be used by the hardcopy device to control printing in some way. What this information is and how it must be stated varies from hardcopy device to hardcopy device and from operating system to to operating system. This argument is provided as a convenience for users who wish to control their printing environment and who are knowledgeable about the "environments" used by the chosen hardcopy device.

  The diagnostic system will use some printing environment by default if the user does not specify one by using the ENV argument.

OFF
: Cease to echo input and output to the hardcopy device.

Command Examples:

```
DUI> hc

DUI> hardcopy ldev 23
```

Related Commands:

OUTFILE
RUN (with OUTFILE modifier)

Limitations:

- Output may not go to the printer immediately; it may be spooled first.

HELP

The HELP command is used to get information about the diagnostic system. Using the HELP command without stating a topic will cause general information about the on-Line diagnostic subsystem to appear. If a topic is specified, information about that topic will appear. HELP messages are available for every diagnostic program and every section and step in every diagnostic. HELP messages also exist for every command listed in this document.

Syntax:

```
HELP [HELP                              ]
     [command [SYNTAX]                  ]
     [run command modifier  [SYNTAX]    ]
     [MNEMONICS                         ]
     [program name                      ]
     [program name SECTIONS             ]
     [program name SC                   ]
     [program name SECTION n            ]
     [program name SC n                 ]
     [program name (n)                  ]
     [LDEV [=] logical device           ]
     [LDEV [=] logical device ACCESS    ]
     [LDEV [=] logical device ID        ]
     [PDEV [=] physical path            ]
     [PDEV [=] physical path ACCESS     ]
     [PDEV [=] physical path ID         ]
```

Abbreviations and Alternate Tokens:

```
H
?
```

Modifiers:

omitted — General information about the online diagnostics subsystem will be printed.

HELP — HELP used as a modifier to the HELP command will cause a complete list of the topics about which HELP can give information to be printed.

command — Causes a brief description of the purpose of the *command* to appear along with a syntax diagram. A help message exits for every command in this document.

command SYNTAX — Causes only the command syntax to appear, not a description of the command.

run command modifier
Causes a brief description of the purpose of the *run command modifier* to appear along with a syntax diagram. A help message exits for every run command modifier in this document.

run command modifier **SYNTAX**
Causes only the command syntax to appear, not a description of the run command modifier.

MNEMONICS
Causes a list of all currently recognized mnemonics to be output.

program-name
Causes a general message about the diagnostic to appear. Although the specific contents of this message will vary from diagnostic to diagnostic, typically, the message will state what the diagnostic tests and what the diagnostic is capable of doing.

program-name SECTIONS
program-name SC
Causes a list of all the sections available in the named diagnostic to appear along with a brief description of what each does. A list of mnemonics which may be used in place of section numbers will also appear. Each mnemonic will be annotated with a list of the section numbers it replaces.

program-name SECTIONS *n*
program-name SC *n*
Causes a detailed description of section *n* of the named diagnostic to appear. If the section is divided into steps, a list of these will appear. If any of the steps may be replaced with mnemonics, a list of those mnemonics annotated with the steps they may replace will also appear.

program-name (*n*)
Causes a detailed description of step *n* of the named diagnostic to appear.

LDEV *logical device*
PDEV *physical path*

Results in a message stating the access mode (exclusive or non-exclusive) and the test mode (destructive or non-destructive) of the device. The user will be told what must be done to get the diagnostic security level necessary to access the device. The product name of the device (e.g., HP1234B) will also be given.

LDEV *logical device* ACCESS
PDEV *physical path* ACCESS

The access mode (exclusive or non-exclusive) and the test mode (destructive or non-destructive) of the device will appear. A message will also be output stating what the user must do to get the diagnostic security level needed to access the device.

LDEV *logical device* ID
PDEV *physical path* ID

The product name of the device will appear. This will be some number such as HP1234A.

Command Examples:

```
DUI> h

DUI> ? foodiag

DUI> help foodiag sections

DUI> h foodiag sc 5

DUI> h foodiag (155)

DUI> help pdev 4/2.3

DUI> ? mnemonics

DUI> ? h

DUI> ? resume

DUI> h h syntax
```

INSTALL

The **INSTALL** command is used to add programs to the diagnostic system.

The **INSTALL** command causes the **DUI** to update diagnostic system files with information about the program to be installed. It also places the program's message catalog and the executable program file in their correct places in the diagnostic system. Special files (such as downloadable code files) needed by the program are also taken into the diagnostic system using the **INSTALL** command.

The user is expected to have ready an executable copy of the program and a GENCAT formatted copy of the message catalog. The user is also expected to have ready any special files peculiar to the program. These files may be formatted in any way needed by the program.

Information about the program, which is used each time the program is run, is gathered from the program's message catalog when the **INSTALL** command is given.

The **INSTALL** command is used only to add new programs to the diagnostic system; the user will not be allowed to install (add) a program which already exists in the diagnostic system. Instead, the user may

1. revise information about an already installed program by using the **MODIFY** command, or
2. reinstall a program by first removing it from the diagnostic system using the PURGE command, or
3. install the program under a different name.

Information about the diagnostic is gathered from messages in the installation set of the diagnostic's message catalog. The content of these messages is as follows:

```
1 - Program version number in the v.uu.ff format
    (e.g., A.01.09).

2 - Catalog version number in the v.uu.ff format
    (e.g., A.01.09).

3 - Type of program being installed. Options are:

            DIAGNOSTIC

            UTILITY

            EXERCISER

            VERIFIER
```

```
4 - Devices diagnosed by the program.

    ID          [AUTO] [DECODE]

     :            :       :        %

     :            :       :        %

     :            :       :        %

{Where:                                                    }
{ ID      = Product number for diagnosed device            }
{           (e.g., HP7978B)                                }
{ AUTO    = If specified, this indicates that the          }
{           program serves as the designated auto-         }
{           diagnostic for this device.                    }
{ DECODE  = If specified, this indicates that the          }
{           program serves as the designated hardware      }
{           status decoder for this device.                }
```

Please note the percent signs (%) in the above example. These tell the message formatter, GENCAT, that the following line is part of the message. The percent signs **must** appear at the end of every line except the last in message 4.

If any of the above messages contains an invalid value, an error message descriptive of the exact problem will be issued and the installation will be aborted. It will then be up to the user to fix the message catalog and reinstall.

The user will be prompted for four additional pieces of information which are needed to complete the installation process. These are:

- The location (file path) of the diagnostic system into which the installation is to be done. This allows experimental/working diagnostic subsystems to be built and maintained for development and integration purposes. The default will always be the location of the diagnostic system originally issued with the operating system.
- The name of the file containing the executable program. This name must be specified as fully as is necessary to locate it in the file system.
- The names of the files containing the formatted message catalogs. These names must be specified as fully as is necessary to find them in the file system. The message catalog files must be the formatted output of the Native Language Support catalog formatter, GENCAT.

  The user will also be asked to specify the language of each message catalog. The language of the message catalog must be specified in the same language in which the DUI is currently printing messages. For example, if the DUI is currently using English and a German language catalog is being installed, the language of the catalog would be specified as being "German." If the DUI is currently using German, the language of a German language catalog would be specified as "Deutsche."

Caution

MOST IMPORTANTLY, THE LANGUAGE OF THE FIRST MESSAGE CATALOG INSTALLED FOR EACH DIAGNOSTIC PROGRAM MUST BE THE SAME LANGUAGE THE DUI IS CURRENTLY USING TO PRINT MESSAGES.

- The DUI reads the installation information for a diagnostic program from the installation set of the first message catalog installed for that program. If this information is in a different language than that of the DUI, the DUI will not be able to process it and the installation will fail.
- The names of any additional files used by the program. Such files might, for example, contain downloadable code or environmental scripts used by the program.

The information gathered from the message catalog will be displayed (in the same format as for the LIST command). This assumes that no error occurred while getting the information from the message catalog.

If the installation is successful, a message indicating so will be generated and the user will be returned to the DUI prompt. If the installation fails, messages explaining the exact nature of the problem will be displayed before the DUI prompt reappears.

Syntax:

```
INSTALL program-name
```

Abbreviations and Alternate Tokens:

none

Modifiers:

program-name	The name by which the program will be referred when issuing the RUN command.

Data Prompts:

```
Specify the file path of the diagnostic system into which the
program will be installed:
FILE PATH [<cr>= ccccccccccccc] >
```

{Where ccccccccccccc will be the file path to the group or directory where the "normal" or "default" diagnostic system resides}

```
Specify name (qualify as necessary) of executable program file:
FILE NAME [<cr>= cancel install]  >
```

```
Specify name (qualify as necessary) and language of message
 catalog file:
FILE NAME (current language) [<cr>= cancel install]  >
FILE NAME [<cr>= no more message catalogs]  >
   LANGUAGE [<cr>= cancel this file]  >
FILE NAME [<cr>= no more message catalogs]  >
```


```
LANGUAGE [<cr>= cancel this file]  >
  :
  :

Specify name (qualify as necessary) of another associated file:
FILE NAME [<cr>= no more files]  >
FILE NAME [<cr>= no more files]  >
   :          :
   :          :
```

Output:

The following program is being added to *name of diagnostic subsystem*

Name	Program Version	Program Type	Catalog Languages	Devices	Associated Files
cccccccc	*c.cc.cc*	*cccc*	*ccccccccc*	*cccccc* ad*	*cccccccc*
				cccccc	*cccccccc*
			:	:	:
			:	:	:
			:	:	:

```
* a  program is the auto-diagnostic for this device
  d  program is the hardware status decoder for this device
```

Command Examples:

```
DUI> install foodiag
```

Related Commands:

```
MODIFY
PURGE
```

Limitations:

- The installation will be rejected if the named program already exists in the diagnostic system.

LIST

The LIST command causes a list of programs installed in the diagnostic system to appear. The modifiers used determine which programs are listed and what additional information is given for each.

Syntax:

```
LIST [program-name [[,]... ]] [LONG ]
                              [SHORT]
     [PRODUCT product-name    ] [LONG ]
                              [SHORT]
     [TYPE {DIAGNOSTIC}       ] [LONG ]
           {EXERCISER}          [SHORT]
           {VERIFIER}
           {UTILITY}
```

Abbreviations and Alternate Tokens:

L

Modifiers:

omitted — A list of all programs installed in the diagnostic system will be printed.

LONG — The *program-name* information for all installed diagnostic programs is printed. (See *program-name* LONG modifier, below).

SHORT — A list of all programs installed in the diagnostic system will be printed. (Same effect as giving the LIST command without any modifiers).

program-name
program-name SHORT
The program name is printed.

program-name LONG
The program name is printed along with its

* executable program (code) version number
* message catalog languages
* type (diagnostic, exerciser, verifier, or utility)
* auto-diagnostic responsibilities
  (i.e., is it an auto-diagnostic)
* decoding responsibilities
  (i.e., may it be used to decode hardware status)
* products which the program tests
* associated files (such as downloadable code files used by the program)

```
PRODUCT product-name
PRODUCT product-name SHORT
              The programs which run with the specified product are
              listed.

PRODUCT product-name LONG
              The programs which run with the specified product are
              listed along with complete information about each
              program.
              (See program-name LONG modifier, above).

TYPE DIAGNOSTIC
TYPE DIAGNOSTIC SHORT
              Lists the 'diagnostic' programs.

TYPE DIAGNOSTIC LONG
              Lists the 'diagnostic' programs with complete program
              information.
              (See program-name LONG modifier, above).

TYPE EXERCISER
TYPE EXERCISER SHORT
               Lists the 'exerciser' programs.

TYPE EXERCISER LONG
              Lists the 'exerciser' programs with complete program
              information.
              (See program-name LONG modifier, above).

TYPE VERIFIER
TYPE VERIFIER SHORT
               Lists the 'verifier' programs.

TYPE VERIFIER LONG
              Lists the 'verifier' programs with complete program
              information.
              (See program-name LONG modifier, above).

TYPE UTILITY
TYPE UTILITY SHORT
               Lists the 'utility' programs.

TYPE UTILITY LONG
              Lists the 'utility' programs with complete program
              information.
              (See program-name LONG modifier, above).
```

Command Examples:

```
DUI> list

DUI> list short

DUI> list type utility

DUI> list xdiag long

DUI> list product HP1234

DUI> list product HP1234 short

DUI> list long
```

Output:

```
{For default or SHORT modifier:}

program name    program name    program name    program name
        :               :               :               :
        :               :               :               :
        :               :               :               :


{For complete program name:}

          Program   Program  Catalog                    Associated
  Name    Version   Type     Languages    Devices       Files
========  ========  =======  ===========  ============  ==========
cccccccc  c.cc.cc   cccc     ccccccccccc  ccccccc ad*   cccccccc
                                          ccccccc       cccccccc
                                    :           :           :
                                    :           :           :
                                    :           :           :

* a  program is the auto-diagnostic for this device
  d  program is the hardware status decoder for this device
```

LISTREDO

The LISTREDO command displays the user's command history stack. Each command in the display will be numbered - these numbers may be used with the DO and REDO commands. If a number *n* is given as an argument to the command, only the *n* most recent commands in the history stack will be displayed. Otherwise, the entire history stack will be displayed.

Syntax:

LISTREDO [*n*]

Abbreviations and Alternate Tokens:

LR

Modifiers:

n The number of commands in the history stack to be displayed. The *n* most recent commands will appear. If *n* is greater than the current size of the user's history stack the entire history stack will be displayed

Command Examples:

```
DUI> listredo

DUI> lr 3
```

Related Commands:

DO
REDO

MODE

The **MODE** command, used without a modifier, will tell the user the current mode of the operating system; either *single user* or *multi-user*. If one of the arguments is used, the operating system will be placed in the mode specified by the argument - assuming the user has the security capability necessary to place the system into that mode.

Syntax:

```
MODE [[=] SINGLE]
     [[=] MULTI ]
```

Abbreviations and Alternate Tokens:

none

Modifiers:

omitted	The current operating system mode will be displayed.
SINGLE	Place the operating system into single user mode. If the system is already in single user mode this modifier will have no effect. When an operating system is placed in single-user mode ALL other users are logged off.
MULTI	Place the operating system into multi-user mode. If the operating system is already in multi-user mode this modifier will have no effect.

Command Examples:

```
DUI> mode multi

DUI> mode

DUI> mode single
```

Related Commands:

SHOWSTATE

Limitations:

- The operating system mode will only be changed if the command is given by a user with diagnostic system security level 0.
- Placing an operating system in single-user mode causes all other users to be kicked out of the system - this could have serious, adverse affects especially on the popularity of the kicker.

MODIFY

The MODIFY command allows a user to change information about a diagnostic program without needing to reinstall the diagnostic. The changes specified will be permanent, but the message catalog, which contains the original information, will not be updated. That is, the original installation information in the message catalog will not be overwritten.

The user will be prompted for the location (file path) of the diagnostic system to be modified. This allows experimental/working subsystems to be built and maintained for development and integration purposes. The default will always be the location of the diagnostic system issued with the operating system.

The user will then be presented with a menu of modification tasks. Any or all of the tasks may be selected in any order. Any task may be selected more than once to correct the previous correction (except, of course, that CANCEL will not undo a previous cancel). Corrections will not actually be made final until the user selects DONE, at which point a list of the program information (same format as the LIST command) will be displayed.

If the changes are successfully finalized, a message indicating so will be generated and the user will be returned to the DUI prompt. If the changes could not be finalized, messages explaining the exact nature of the problem will be displayed before the DUI prompt reappears. The CANCEL task cancels **all** pending corrections that have not already been finalized with the DONE task. If DONE is selected and no corrections are pending, the MODIFY command simply terminates without modifying the program information.

The menu is only displayed once. However, the user may recall the menu by typing REFRESH at the MODIFY TASK prompt.

Default values for MODIFY prompts correspond to the pre-existing values for the program being modified.

The CATALOG task allows the user to add to and delete from the list of message catalogs for the specified program. A list of the languages of all the message catalogs associated with the diagnostic program will be displayed. The user will then be prompted to delete from this list. Next the user will be prompted for the locations and names of any message catalog files to be added to the list. The user will also be prompted for the name of the language the message catalog is written in.

The language of the message catalog must be specified in the language that the DUI is currently printing messages in. For example, if the DUI is currently using English and a German language catalog is being installed, the language of the catalog would be specified as being "German." If the DUI is currently using German, the language of a German language catalog would be specified as "Deutsche."

Installation data in the catalogs is **not** examined (as it would be if the INSTALL command were used); therefore, the current information for the program may not correspond to that in the catalogs. All catalog files must be the formatted output of the Native Language Support catalog formatter, GENCAT.

The **CODE** task allows replacement of the code for the specified program. The user will be prompted for the location of the file containing the executable code for the program as well as for the new version number. Note that a code change strictly requires a version change. Likewise, to modify the version number, the user must supply a new program file.

The **DEVICES** task allows the user to add to, delete from, and correct the list of devices which the program diagnoses (if applicable). A list of currently diagnosed devices will be displayed. Then the user will be prompted to add to, delete from, and correct the list. In addition to supplying the device name, the user will be able to specify if the program is the designated auto-diagnostic and/or hardware status decoder for that device. If any other program already serves as the designated auto-diagnostic/status decoder for the device, the user will be prompted to confirm an override of the previous designation. Prompts for additional devices will continue to appear until a simple carriage return (<cr>) is entered to terminate the prompt. When this happens, the user will be prompted for the names of any devices to be removed from the list. The user will be warned if deleting a device from the list of devices diagnosed by the program would result in that device no longer having a designated auto-diagnostic and/or hardware status decoder in the system. As with new devices, the user will continue to be prompted for additional devices until <cr> is entered.

The **TYPE** task allows the user to change the function type of the program. Valid options include **DIAGNOSTIC**, **EXERCISER**, **UTILITY**, and **VERIFIER**. These options may be abbreviated to a minimum number of characters.

The **FILES** task allows the user to add and delete extraneous files needed by the program such as files of downloadable code. A list of the files associated with the program will be displayed. Then the user will be prompted to delete from and add to the list.

The **CANCEL** task effectively destroys all changes made since entering **MODIFY** command.

The **DONE** task exits the **MODIFY** command. If any changes are pending, they are finalized.

The **SHOW** task displays the values for the data items as modified by any pending modifications. This command will produce results identical to the **LIST** command for the program being modified.

Syntax:

MODIFY *program-name*

Abbreviations and Alternate Tokens:

MOD

Modifiers:

program-name	The name of the diagnostic program for which information is to be changed.

Command Examples:

```
DUI> modify xdiag

DUI> mod ydiag
```

Data Prompts:

```
Specify diagnostic system in which program will be modified:
[<cr>= supported system] >

   Modification Options:
          CATALOG  - Add or delete program message catalogs
          CODE     - Replace program code
          DEVICES  - Change list of diagnosed devices
          TYPE     - Change type of program
          FILES    - Add or delete special files

          CANCEL   - Cancel all pending changes and return
                     to modify menu

          DONE     - Implement changes and return to DUI
                     prompt

          EXIT     - Cancel all pending changes and return
                     to DUI prompt

          LIST     - Display pending changes

   MODIFY TASK >


   {for CANCEL task: }

        OK to cancel pending corrections? (YES/NO) [NO]  >


   {for CATALOG task: }

         Catalogs installed for this program are:
              - American
              - English
              - French
              - Canadian-French
              - Kanada
              - Katakana
              - German
```

```
Enter languages of catalogs which should be deleted:
   SYNTAX: language-name (as listed above)

    LANGUAGE [<cr>= no more languages] >
    LANGUAGE [<cr>= no more languages] >
    LANGUAGE [<cr>= no more languages] >
        :          :
        :          :


Need file containing program MESSAGE CATALOG and
  the language the file is written in ...
FILE NAME [<cr>= no more files] >
 LANGUAGE [<cr>= cancel this file] >
FILE NAME [<cr>= no more files] >
 LANGUAGE [<cr>= cancel this file] >
    :
    :


{for CODE task: }

    Need file containing executable PROGRAM code....
    FILE NAME [<cr>= cancel] >
    PROGRAM VERSION (v.uu.ff) >


{for DEVICES task: }
    Current devices installed for this program:
        - device v  AUTO
        - device x
        - device y  AUTO  DECODE
        - device z        DECODE

    Enter devices NO LONGER SERVICED by this program:
                           SYNTAX : device-name
            DEVICE  [<cr>= no more devices] >
            DEVICE  [<cr>= no more devices] >
            DEVICE  [<cr>= no more devices] >
               :          :
               :          :

    Enter NEW devices now serviced by this program
       OR CORRECTIONS to current devices
                           SYNTAX : device-name [Auto] [Decode]
            DEVICE  [<cr>= no more devices] >
            DEVICE  [<cr>= no more devices] >
            DEVICE  [<cr>= no more devices] >
               :          :
               :          :
```

```
{for TYPE task: }

     Program Type Options:
               DIAGNOSTIC
               EXERCISER
               UTILITY
               VERIFIER

     Specify program TYPE [ccccccccc] >


{for FILES task: }

     Current files:
               fileA
               fileB
                 :
                 :
                 :

     Enter files to be deleted
        FILE NAME [<cr>= no more files] >
        FILE NAME [<cr>= no more files] >
        FILE NAME [<cr>= no more files] >
              :          :
              :          :

     Enter files to be added
        FILE NAME [<cr>= no more files] >
        FILE NAME [<cr>= no more files] >
        FILE NAME [<cr>= no more files] >
              :          :
              :          :
```

Output:

```
        {for DONE task: }

    The following MODIFICATION is about to be made to the
    On-line Diagnostic Subsystem:


             Program   Program  Catalog                          Associated
      Name   Version   Type     Languages   Devices              Files
    ========  ========  =======  ==========  ===========  ==========
    cccccccc  c.cc.cc   cccc     ccccccccc    cccccc ad*   cccccccc
                                             cccccc       cccccccc
                                      :           :            :
                                      :           :            :
                                      :           :            :

  * a  program is the auto-diagnostic for this device
    d  program is the hardware status decoder for this device


        {for SHOW task: }


             Program   Program  Catalog                          Associated
      Name   Version   Type     Languages   Devices              Files
    ========  ========  =======  ==========  ===========  ==========
    cccccccc  c.cc.cc   cccc     ccccccccc    cccccc ad*   cccccccc
                                             cccccc       cccccccc
                                      :           :            :
                                      :           :            :
                                      :           :            :

  * a  program is the auto-diagnostic for this device
    d  program is the hardware status decoder for this device
```

Related Commands:

```
INSTALL
LIST
PURGE
```

Limitations:

- The installation information in the existing message catalog will not be updated.
- The installation information contained in a new message catalog will be ignored.

OUTFILE

The OUTFILE command causes all diagnostic system input and output to be written into the specified file. This is in addition to having the I/O appear on the user's terminal.

Outfiles are opened and processed so that each "line" in the file will consist of 80 characters. That is, each "line" has some combination of 80 characters (including spaces), followed by a carriage return, or by a line feed, or by a carriage return and a line feed (the line termination character(s) is operating system dependent). This file organization would be thought of as "80 byte fixed length record ASCII" on MPE XL.

A user needn't do anything to cause an *outfile* to be organized in this manner. Conversely, a user cannot do anything to force the diagnostic system to write to *outfiles* in any other way.

Syntax:

```
OUTFILE [=] {filename}
            {OFF     }
```

Abbreviations and Alternate Tokens:

```
OF
```

Modifiers:

filename	The name of the file into which all I/O should be placed.
OFF	Stop copying I/O into the file.

Command Examples:

```
DUI> outfile yfoo

DUI> of xfoo

DUI> of off
```

Limitations:

- I/O will not be copied into a file if the name given is that of a pre-existing file.

PURGE

The PURGE command causes the named program to be removed from the diagnostic system. The catalog and program files for the named program are deleted from the location of the selected diagnostic system and all references to the named program are removed from diagnostic system files. All special files (such as downloadable code files) associated with the program will also be deleted.

The user will be prompted for the location of the diagnostic system from which the program is to be purged. Information about the program to be purged will be displayed (LIST command format) and the user will be prompted for a verification of the purge. If the diagnostic cannot be purged, a detailed explanation will be provided before returning to the DUI prompt. Otherwise, the user will be immediately returned to the DUI prompt.

Syntax:

PURGE *program name*

Abbreviations and Alternate Tokens:

none

Modifiers:

program name	The name of the diagnostic program which is to be removed from the specified diagnostic system.

Data Prompts:

```
Specify diagnostic system from which program will be purged:
FILE PATH [<cr>= ccccccccccccc] >
```

{Where ccccccccccccc will be the file path to the group or directory where the "normal" or "default" diagnostic system resides}

```
The following program, along with all of its associated files and
message catalogs, will be removed from the diagnostic system:

          Program   Program  Catalog                          Associated
  Name    Version   Type     Languages    Devices             Files
========  ========  =======  ===========  ==========          ==========
cccccccc   c.cc.cc    cccc     ccccccccc    cccccc ad*          cccccccc
                                           cccccc              cccccccc
                                    :           :                  :
                                    :           :                  :
                                    :           :                  :

* a  program is the auto-diagnostic for this device
  d  program is the hardware status decoder for this device

  Do you still wish to remove this program? [yes/no]
```

Command Examples:

```
DUI> purge xdiag
```

Related Commands:

```
INSTALL
MODIFY
```

REDO

REDO allows the user to edit and re-execute any command in the command line history stack. REDO is interactive. The DUI will display the command so that the user may edit it. After the user edits the command the DUI will display the modified command. This version of the command may also be edited by the user. This will continue until the user enters a carriage return without having done any more editing on the command line displayed by the DUI - the command will then be executed.

The first edit of the command may be specified with *edit-directives* attached to the REDO command. Subsequent edits of the command are done by typing editing directives under the displayed command (see examples).

Syntax:

REDO [[CMD=]*command-id*][, *edit-directives*]
[;EDIT= *edit-directives*]

Abbreviations and Alternate Tokens:

none

Modifiers:

command-id — Specifies the command to re-execute. The command may be specified by its relative or absolute order in the command line history stack, or by name (as a string) in whole or in part. The default *command-id* is -1, the most recent command. An error is detected if the *command-id* does not exist in the command line history stack.

The command represented by *command-id* will be displayed to the user (in a modified form if the command was given with *edit-directives*). The user may then modify the command again.

Each time the user modifies the command it will be re-displayed in its modified form. Command execution will not occur until the user hits a carriage-return without first having re-edited the command.

COMMAND-ID EXECUTES

(omitted)	Previous command (same as REDO -1)
-n	The nth command before the most recent one. N is a number in the command line stack relative to the most recent command. The most recent command is -1.
m	Command number m in the command line stack. The number m is absolute.
string	The most recent command beginning with this string.

edit-directives

A string specifying changes to be made in the command represented by the *command-id* before it is displayed to the user. When the (edited) command line is displayed, the user may hit carriage-return to execute the command or may edit the command further by using the editing directives. Editing directives are placed under that part of the command string where the user wishes them to take effect.

If no *edit-directives* are given, the command represented by the *command-id* is displayed so the user may edit it as described above.

The *edit-directives* must be surrounded by quotation marks (" ") if they contain any scanner/parser delimiters such as: , ; " ' [ ] = or a space.

The editing directives which may be used as *edit-directives* are:

DIRECTIVE	EFFECT
i	INSERT. If text follows the i, the text following i is inserted in the current line at the position after the i.
r	REPLACE. If text follows the r, the text following r replaces the same number of characters in the current line, beginning at the position of r.

d	DELETE. Deletes a character from the current line for each d specified in the edit line. Note that "d d" does not specify a range but simply deletes one character from the position above each d. Multiple d's may be followed by an insert or replace operation.
d>	DELETE. Deletes to the end of the current line from the position specified by d>. It may be followed by an INSERT or REPLACE operation.
>	APPEND. > followed by text appends the text to the end of the current line. If > is positioned beyond the end of the current line, then a replacement is performed instead.
>d	DELETE. Deletes from the end of the current line, right-to-left. Multiple d's may be specified after >, as well as INSERT and REPLACE strings.
>r	REPLACE. Replaces characters at the end of the command line. The replacement is done so that the last (rightmost) character of the replacement string is at the end of the line.
c	CHANGE. Changes all occurrences of one string to another in the current line when the search string and replace string are properly delimited. A proper delimiter is a non-alphabetic character: ', ", /, etc. The substitution is specified as: c<delim> search-string <delim> [replace-string [<delim>]] Omitting the replace-string causes occurrences of search-string to be deleted, with no substitution.
u	UNDO. A single u in column one cancels the most recent edit of the current line. Using the UNDO command twice in a row cancels all edits for the current line and re-establishes the original, unedited line. If u is placed anywhere other than column one of the current line, then a simple replacement is performed. UNDO makes sense only if you have a line on which you have performed some editing that can be "undone."

other	Simple replacement. Any other character (not i, r, d, d>, >, >d, c, or u) causes that character to be replaced in the current line at t position indicated by the character. In fact, simple replacement also occurs for the editing characters i, r, c, or > if they are not followed by text; or if > appears at or beyond the current end of line.

Command Examples:

EDITING SAMPLES

Practical uses of the editing commands listed above are shown here:

EDIT	ACTION
u	First occurrence undoes the previous edits. The u must be in column one.
u	Second occurrence undoes all edits on the current line. The u must be in column one.
rxyz	Replaces the current text with xyz starting at the position of r.
xyz	Replaces the current text with xyz starting at the position of x.
ixyz	Inserts xyz into the current line, starting at the position immediately before the i.
ddd	Deletes three characters, one above each d.
'd xyz'	Deletes a single character above the d, skips one space, then replaces the current text with xyz starting at the position of x.
ddixyz	Deletes two characters, then inserts xyz in the current line in the position before the i.
"d d"	Deletes one character above the first d, skips two spaces and deletes a second character above the second d. It does not delete a range of characters.
'd d>xyz'	Deletes a single character above the first d, skips two spaces and deletes to the end of the line beginning at the second d, and then appends xyz to the end of line.
>xyz	Appends xyz to the end of the current line.
>ddxyz	Deletes the last two characters from the end of the current line and then appends xyz to the end of the line.

```
>rxyz              Replaces the last three characters in the current
                   line with xyz.

>ixyz              Appends xyz to the end of the line. In this case,
                   the i command is superfluous, because >
                   accomplishes the same result. Using >xyz would
                   be sufficient.

c/ab/def           Changes all occurrences of ab to def, starting
                   at c.


c"ab"              Deletes all occurrences of "ab" starting at c.

cxyz               Replaces the current text with cxyz, starting at c.
                   Because no delimiters have been specified (as they
                   were in the previous two examples), this is a
                   simple replacement.
```

```
EXAMPLES


REDO pas           Edits the the most recent command beginning with
                   the string pas.

REDO 10            Edits command number 10 (absolute) on the command
                   history stack.

REDO -2            Edits the second-to-last command on the stack (one
                   command before the most recent).

REDO , "c/$null/$STDLIST"  Change all occurrences of $null to $STDLIST
                           in the most recent command before editing
                           it.

REDO run, ">;debug" Append ;debug to the the most recent :RUN command
                    and then edit it.
```

Related Commands:

```
DO
LISTREDO
REDOLOAD
REDOSAVE
REDOSIZE
```

Limitations:

- REDO is based on an MPE XL command.

REDOLOAD

REDOLOAD replaces the user's command history stack with the history stack which was saved by the command REDOSAVE.

Syntax:

REDOLOAD [*filename*]

Abbreviations and Alternate Tokens:

RDLD

Modifiers:

filename	The name of the history stack which was saved using the REDOSAVE command. If no file is specified, the default saved history file, cmdhist, will be restored.

Command Examples:

```
DUI> redoload

DUI> rdld histfile
```

Related Commands:

DO
REDO
REDOSAVE
REDOSIZE

Limitations:

- Attempting to restore a file which does not contain a history stack will cause unpredictable results. cmdhist, the default saved history file, will always contain the stack most recently saved - this might have been saved by another user.

REDOSAVE

REDOSAVE causes all or part of the user's command history stack to be saved into a file. This file may later be restored as the current history stack by using the REDOLOAD command. The command history stack file created by this command is a permanent file and so will continue to exist between operating system sessions.

Syntax:

REDOSAVE [FILE [=] *filename*] [*n*]

Abbreviations and Alternate Tokens:

RDSV

Modifiers:

FILE *filename*	The name of the file in which to save the command history stack. If specified, this must be a new file. If not specified, the command history stack will be saved into the default file, **cmdhist**, whether or not this file already contains a command history stack.
n	The number of commands in the current history stack to be saved by being written to a file. The *n* most recent commands will be saved. If *n* is greater than the number of commands actually in the current history stack or if *n* is not specified the entire history stack will be saved.

Command Examples:

```
DUI> redosave

DUI> rdsv 3

DUI> rdsv file foohist

DUI> rdsv 18 file fiehist
```

Related Commands:

DO
REDO
REDOLOAD
REDOSIZE

Limitations:

- If a file is named, it cannot be a pre-existing file. If no file is specified, the history stack will be saved into the default file **cmdhist**, overwriting the contents of this file.

REDOSIZE

REDOSIZE allows the user to set the maximum number of commands which will be saved in the user's command history stack. If this command is not used, the history stack will contain a maximum of 25 commands.

Syntax:

```
REDOSIZE [=] n
```

Abbreviations and Alternate Tokens:

```
RDSZ
```

Modifiers:

n	The number of commands to be held in the user's command history stack. The *n* most recent commands will be retained.

Command Examples:

```
DUI> rdsz 15

DUI> redosize 12
```

Related Commands:

```
DO
REDO
REDOLOAD
REDOSAVE
```

Limitations:

- Setting the command history stack to an arbitrarily large size using this command could cause unpredictable results on some machines. Although no limit is placed on the number of commands which the history stack may hold, it is assumed that the user will limit the size to some "reasonable" number. The maximum "reasonable" size is operating system and machine dependent.

REPLY

REPLY is used to send a reply to the prompt of a diagnostic program which is running in the background.

If there is only one program running in the background, the *program-id* (a unique process identifier) need not be specified. If more than one program is running in the background and no *program-id* is specified, a list of the background programs waiting for replies will be given. Both the program-id and the programs prompt of every background program waiting for a reply will appear. The user may then repeat the **REPLY** command.

Not giving a *reply message* has the effect of sending a null string to the program as the reply.

Syntax:

REPLY [*program-id*] "*reply message*"

Abbreviations and Alternate Tokens:

REP

Modifiers:

program-id	The unique identifying number assigned to a process by the operating system.
reply message	The message to be sent to the background diagnostic. Could be a carriage return.

Command Examples:

```
DUI> reply yes

DUI> rep 12 no
```

Related Commands:

BACKGROUND

Limitations:

- The DUI does not check the reply message for validity - the message is sent to the diagnostic program exactly as it is typed.
- If a valid *program-id* is given without a *reply message*, the effect is to send a null string as the reply. A null string is also sent to the program if no *program-id* is given but only one program with a reply pending is running in the background.

RESUME

RESUME causes the execution of a previously suspended diagnostic program to continue. If the program which is to be resumed is not specified and the user has only one suspended program that one program will be resumed. A list of programs will be printed if more than one program is suspended and the user does not specify a *program-id*. The user should then repeat the command giving one or more of the *program identifiers* from this list.

Syntax:

```
RESUME [program identifier [[,]...]] [BACKGROUND]
       [ALL                         ] [BACKGROUND]
```

Abbreviations and Alternate Tokens:

```
RES
```

Modifiers:

program identifier	The diagnostic's identifier - the unique number assigned to the diagnostic program by the operating system.
ALL	Resume running all suspended programs.
BACKGROUND	Place the specified programs in the background - if they are not there already - and then resume them.

Command Examples:

```
DUI> resume

DUI> res

DUI> res 24 26 background

DUI> res all background
```

Limitations:

- The user may only resume programs which he "owns."
- "Resumed" programs which had been running in the foreground before being suspended will run in the foreground after being resumed. "Resumed" programs which were running in the background when suspended will run in the background when resumed.
- If *i*) more than one *program identifier* is given as an argument to the RESUME command or *ii*) the ALL argument is given but the BACKGROUND argument is not given and *iii*) more than one of the suspended programs to be resumed was running in the foreground when suspended then only the designated foreground process most recently suspended will resume. All other suspended foreground processes will remain suspended.

  All designated background suspended processes will be resumed.

RUN

The **RUN** command is used to start one or more diagnostic programs. It may be used explicitly or implicitly.

When more than one diagnostic program is named within a run command, all internal setup for each one of the programs will be performed before the DUI calls upon the operating system to launch the programs one after the other. The DUI will not wait for one program to complete before launching the next.

Syntax:

[RUN] {*program name* [*command modifier* [...]]} [,...]

Abbreviations and Alternate Tokens:

none

Modifiers:

program name The diagnostic program which is to be run.

command modifier The possible modifiers are

BACKGROUND
DEBUG
ERRCOUNT
ERRONLY
ERRPAUSE
ERRPRINT
HARDCOPY
INFILE
LDEV
LOOP
OUTFILE
PDEV
SECTIONS
TRACE
Diagnostic Specific Parameters

Please see the following pages for descriptions of each of these modifiers.

Command Examples:

```
DUI> run xdiag

DUI> xdiag

DUI> run xdiag &, ydiag pdev=4.0 sc 3, zdiag ldev 0 bg
```

Limitations:

- The number of diagnostic programs which can be launched with one *run* command is limited by the number of processes the operating system will allow a user.
- Only one diagnostic program may run in the foreground at any one time; when running multiple programs all but one must be run in the background.

RUN COMMAND MODIFIERS

The following pages present detailed functional specifications for the RUN command modifiers.

BACKGROUND

The BACKGROUND modifier causes the diagnostic program just invoked to be run in the background. Output from a program running in the background will appear on the user's terminal unless it is redirected by the use of the OUTFILE modifier.

Syntax:

BACKGROUND

Abbreviations and Alternate Tokens:

BG

Default Value:

off

Examples:

```
DUI> run xdiag &, ydiag bg
DUI> xdiag background outfile xout
```

Limitations:

- If the output of the program placed in the background is not redirected, it will appear on the user's terminal. This could cause confusion if more than one program is running in the background or if a program is also running in the foreground.
- When the output of a program running in the background is not redirected, prompts from the program will appear on the user's screen. These prompts CANNOT be replied to directly since there is no interaction possible with a program running in the background. The user must use the *reply* command to send responses to prompts from programs running in the background.

DEBUG

The DEBUG modifier causes the program to run within a debugger. If the user does not specify which debugger to use, the diagnostic system will invoke a default debugger such as the *nmdebug* debugger on MPE XL or the *xdb* debugger on HP-UX.

Arguments specific to the debugger will not be accepted. This is an exception to the *Support User Interface Standard*.

If the DEBUG modifier has been associated with a diagnostic program by the **set** command, the user may state "debug off" on the run command line in order to run the diagnostic outside of the debugger that one time.

Syntax:

```
DEBUG [[=] OFF      ]
      [[=] debugger ]
```

Abbreviations and Alternate Tokens:

```
DB
```

Default Value:

```
off
```

Examples:

```
DUI> xdiag pdev 2/4.3 debug
DUI> run ydiag ldev ydg/0 debug
DUI> run zdiag debug off
```

Limitations:

- The DEBUG modifier will, normally, be used only for purposes of diagnostic program development and maintenance. Anyone using this modifier must ensure that the diagnostic program being invoked to run under the debugger has been compiled and linked with whatever options are necessary to get the executable file to run under the chosen debugger. Further, the user must see that any additional files needed to run under the debugger are on the system and in the appropriate places. For example, if the *xdb* debugger is used the source and header files for the program must be on the system.

ERRCOUNT

ERRCOUNT sets the number of errors to tolerate before aborting the diagnostic program.

Syntax:

```
ERRCOUNT n
```

Abbreviations and Alternate Tokens:

```
EC
```

Default Value:

infinite - an unlimited number of errors may occur

Examples:

```
DUI> xdiag ldev 4 errcount 12
DUI> run ydiag ldev 2 ec 5
```

Limitations:

- The program will be aborted automatically when the specified number of errors have occurred - the user will not have a chance to continue running the diagnostic, even if the modifier ERRPAUSE has also been set.
- If the number of errors to tolerate before terminating is specified (by using the *n* argument), the number MUST be equal to or less than maxint.

ERRONLY

When ERRONLY is on, only error messages will be printed; most informational messages generated by a diagnostic program will not be displayed.

ERRONLY may be placed "on" by typing ERRONLY or ERRONLY ON.

When ERRONLY is off, all messages will be printed - informational and error.

Syntax:

```
ERRONLY [[=] ON ]
        [[=] OFF]
```

Abbreviations and Alternate Tokens:

```
EO
```

Default Value:

```
off
If ERRONLY is on, the long message form is the default
```

Examples:

```
DUI> run xdiag erronly
DUI> run xdiag erronly on
DUI> set ydiag erronly
DUI> run ydiag ldev 3 erronly off
```

ERRPAUSE

When ERRPAUSE is on and an error occurs, the user will be queried as to whether to continue executing the program. If the user responds yes the program will continue. If the user responds no the program will be aborted.

The modifier ERRPAUSE is equivalent to ERRPAUSE ON.

Syntax:

```
ERRPAUSE [[=] ON ]
         [[=] OFF]
```

Abbreviations and Alternate Tokens:

```
EPS
```

Default Value:

```
off
```

Examples:

```
DUI> run xdiag pdev 2/4.2 errpause
DUI> ydiag ldev 4 errpause off
```

Limitations:

- If ERRCOUNT has also been set and the maximum number of errors reached, the program will abort without querying the user.

ERRPRINT

ERRPRINT may be used to control the number of messages printed when an error occurs.

When the LONG modifier is in effect, all error messages associated with an error will be printed - there may be duplicate messages and some of the messages may be obscure. By default, all error messages will be printed when an error occurs - LONG is the online diagnostics subsystem default.

When the SHORT modifier is in effect, only the error message stating what the diagnostic was trying to do, or what the diagnostic believes the problem to be will be printed.

ERRPRINT will always be ON in the online diagnostics subsystem; using the OFF modifier will have no effect.

Syntax:

```
ERRPRINT [ON   ]
         [OFF  ]
         [LONG ]
         [SHORT]
```

Abbreviations and Alternate Tokens:

```
EPR
```

Default Value:

```
on, long
In the online diagnostics subsystem, ERRPRINT cannot be turned off.
```

Examples:

```
DUI> run xdiag errprint          has no effect
DUI> run xdiag errprint on       has no effect
DUI> set ydiag epr short
DUI> run ydiag ldev=3 epr off    has no effect
DUI> xdiag errprint long
```

HARDCOPY

The HARDCOPY modifier causes all of the diagnostic program's input messages and output messages to be printed on a hardcopy device such as a line printer or laser printer.

This modifier does not redirect I/O - it just causes a hardcopy of it to be created. All I/O will also appear on the user's terminal unless a modifier such as INFILE or OUTFILE has also been used.

The user may specify which hardcopy device to use by **ldev** or **pdev** but, normally, the diagnostic system will recognize a particular hardcopy device to be used by default.

The user may also specify an "environment" to be used by the hardcopy device. The environment is information which is used by the hardcopy device to control printing in some way. What this information is and how it must be stated varies from hardcopy device to hardcopy device and from operating system to to operating system. This argument is provided as a convenience for users who wish to control their printing environment and who are knowledgeable about the "environments" used by the chosen hardcopy device.

Syntax:

```
HARDCOPY [LDEV [=] logical device name [ENV [=] environment]]
         [PDEV [=] physical path       [ENV [=] environment]]
         [OFF                                               ]
         [ON                                                ]
```

Abbreviations and Alternate Tokens:

HC

Default Value:

off

Examples:

```
DUI> ydiag hardcopy
DUI> run xdiag hc ldev 7
DUI> run zdiag hc off
```

INFILE

The **INFILE** modifier causes all input data expected by a diagnostic program to be read from the specified *infile* rather than from the standard input device.

Infiles are opened and processed under the assumption that each "line" in the file will consist of 80 or less characters, followed by a termination character or set of characters. The termination character is operating system dependent. . That is, each "line" has some combination of 80 characters or less (including spaces), followed by a carriage return, or by a line feed, or by a carriage return and a line feed, depending on the operating system. This file organization would be thought of as "80 byte fixed length record ASCII" on MPE XL.

An *infile* with any "line" longer than 80 characters will not be read or processed correctly.

The OFF argument may be used if an *infile* was bound to the diagnostic using the *set* command, but the user does not wish to get input to the program from the file during the current run.

The modifier *INFILE* differs from the command *USEFILE*. A *USEFILE* contains a series of commands (and, possibly, data) and controls the diagnostic session until the end of file is reached. An *INFILE* contains data which a particular diagnostic program would expect the user to give it during the course of its (interactive) execution.

Syntax:

```
INFILE [=] {filename}
           {OFF     }
```

Abbreviations and Alternate Tokens:

```
IN
```

Default Value:

data is received via the user's terminal rather than a file

Examples:

```
DUI> run wdiag ldev 3 infile winput


The infile winput might contain the following:

       yes
       3
       continue
       yes
       exit

Assuming this is reasonable data for wdiag to receive during
the course of its execution.
```

LDEV

The LDEV represents the logical name of a device to be tested or used by a diagnostic. This name differs among the various operating systems. For example, on MPE XL systems the LDEV is a number while on HP-UX systems the LDEV is the name of a special device file.

Syntax:

LDEV [=] *logical device name*

Abbreviations and Alternate Tokens:

none

Default Value:

There is no LDEV default value. If a device is needed for a diagnostic program to run, that device MUST be specified using either its PDEV or its LDEV.

Examples:

```
DUI> run xdiag ldev 6
DUI> ydiag sc 5 ldev dsk/c0d0
```

LOOP

The LOOP modifier specifies the number of times the sections and steps are to be repeated before the diagnostic program terminates. If the modifier is given without a number the sections and steps will be repeated until an interrupt is given.

The OFF argument has the same effect as setting LOOP to 1.

If the ERRPAUSE or ERRCOUNT modifiers are also set, their effect will take precedence over LOOP.

Syntax:

```
LOOP [[=] OFF]
     [[=] n  ]
```

Abbreviations and Alternate Tokens:

none

Default Value:

If LOOP is not set only one iteration of the sections and steps will be performed before the diagnostic program terminates.

Examples:

```
DUI> run xdiag ldev 5 sc 4/6 steps 32,46,120/125 loop 6
DUI> xdiag ldev 5 loop
DUI> ydiag pdev 8.4.3 sc 7 loop off
```

Limitations:

- If LOOP is given without an argument, the sections and steps will be repeated infinitely. The only way to get out of the infinite loop is to send an interrupt and abort the diagnostic program. The program will terminate abnormally and with unpredictable results.
- If the number of loops is specified (by using the *n* argument), the number MUST be equal to or less than maxint.

OUTFILE

The OUTFILE modifier causes all output from a diagnostic program to be placed into the named file rather than be displayed on the user's terminal.

Outfiles are opened and processed so that each "line" in the file will consist of 80 characters. That is, each "line" has some combination of 80 characters (including spaces), followed by a carriage return, or by a line feed, or by a carriage return and a line feed (the line termination character(s) is operating system dependent). This file organization would be thought of as "80 byte fixed length record ASCII" on MPE XL.

A user needn't do anything to cause an *outfile* to be organized in this manner. Conversely, a user cannot do anything to force the diagnostic system to write to *outfiles* in any other way.

The OFF argument may be used if an *outfile* was bound to a diagnostic using the *set* command but the user does not wish to have output placed in the file during the current run.

Syntax:

```
OUTFILE [=] {filename}
            {OFF     }
```

Abbreviations and Alternate Tokens:

```
OUT
```

Default Value:

```
Redirection of output does not occur; all output is displayed
on the standard output device which is usually the user's
terminal.
```

Examples:

```
DUI> run xdiag outfile fooout
DUI> ydiag out yout
```

Limitations:

- The file named cannot already exist.

PDEV

A PDEV represents the physical path to a device and is composed of numbers corresponding to hardware slot numbers with various sorts of punctuation separating the numbers. The punctuation used may be machine dependent.

Syntax:

PDEV [=] *physical path*

Abbreviations and Alternate Tokens:

none

Default Value:

There is no PDEV default value. If a device is needed for a diagnostic program to run, that device MUST be specified using either its PDEV or its LDEV.

Examples:

```
DUI> run xdiag pdev 4/2.3 sc 4
DUI> run ydiag pdev 8.1
```

SECTIONS

A SECTION is a major operation or set of related operations within a diagnostic which can be explicitly invoked by a user. Some or all of a diagnostic's SECTIONS may be designated by the diagnostic developer to be default SECTIONS which are to be run if the user does not explicitly invoke one or more SECTIONS. A SECTION which performs more than one operation will have a subset of those operations, the steps, designated as the defaults to be run when the section is named but none of its steps are specified. When the SECTION is invoked without explicitly stating which steps are to be run the "default" steps will be run automatically.

SECTIONS and steps are numbered although the mnemonics listed below may be used instead of explicit numbers. When SECTIONS are invoked they are run in numerical order.

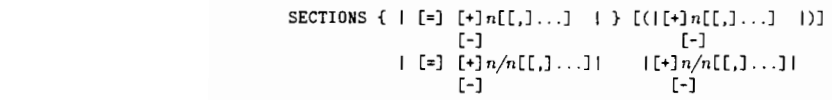
Steps are also run in numerical order. Steps may be explicitly named, using numbers or mnemonics, only if their associated SECTIONS are also named.

The security capability of the user determines whether or not an operation represented by a section or step is actually performed. For example, a user would need a high security capability before the diagnostic system would allow him to perform an operation which could cause the loss of user data on a device.

If a SECTION includes more than one separately callable operation (step) the needed capability is determined and checked for each individual operation. If a user tries to run a section or step for which he lacks the appropriate security capability a message will be output stating which security capability is needed and what security capability the user has.

Any sections and steps which the user invoked for which he does have the appropriate security capability will be run. Any which require a higher user security capability will not be.

Syntax:



Please note that steps are designated using either parenthesis () or square brackets []. The square brackets are not shown in the above syntax diagram.

The "+" and "-" before section and step numbers are used when the user has already set up a list of sections and steps to run using the **set** *command and when the user wishes to run "default" sections and or steps with certain exceptions. The "+" and "-" refer to additions to and subtractions from a list of default sections and steps. Using the "+" will cause the immediately following sections or steps to be executed IN ADDITION TO the previously set sections or steps. Using the "-" will cause previously set sections and steps to be executed with the EXCEPTION of those whose numbers are preceded by the "-".*

In any case, any number or number range preceded by the "+" will be run - security permitting - while any number or number range preceded by the "-" will not be run. The "+" and "-" may also be used with the **sections** *modifier in the* **set** *command to change the default sections and steps to be run when a diagnostic is invoked without needing to re-specify the entire list of wanted sections and steps.*

Abbreviations and Alternate Tokens:

SC

Default Value:

The sections and steps which the diagnostic writer named as the defaults for the invoked diagnostic program. This differs for every diagnostic program.

Examples:

```
DUI> run xdiag pdev 2/4 sections 5/6,7 (-9,20/50)
```

Note: in this example, step 9 will not be executed but steps 20 through 50 will be executed.

```
DUI> ydiag ldev 3 sc 4,6,+9 [10,+12]
```

Mnemonics:

The following may be used instead of section and/or step numbers. However, the section or step numbers which will be executed when one of these mnemonics is used varies from diagnostic to diagnostic. Not all diagnostics will have sections and steps associated with these mnemonics.

Each diagnostic writer determines which sections and steps will be run when one of these mnemonics is given in conjunction with a particular diagnostic.

For a list of the sections and/or steps which will be run when one of these is used type

help *program name* **sections**.

These may be substituted for section numbers and step numbers

```
all
auto
clear
default
errlog
fast
hwstatus
id
loopback
non-destructive
selftest
interactive
```

Mnemonic Descriptions:

Please note that the actual affect of any of these mnemonics may be altered if the capability of the user is too low to allow some operation to be performed. Although the descriptions speak about sections the mnemonics may also be used to substitute for steps.

ALL:	Run all sections, default and non-default alike.
AUTO:	Run all sections which are designated as autodiagnosable. That is, run all sections which might be run automatically by the operating system when it detects a possible hardware defect.
CLEAR:	Run whatever section or sections "clear" or "reset" the device being diagnosed.
DEFAULT:	Run the default sections. This mnemonic will be useful when non-default sections have been specified at a more global scope and the user wishes to only run defaults locally.
ERRLOG:	Run whatever sections read and decode error logs.
FAST:	Run the sections which the diagnostic developer has designated as "fast." That is, run those sections which will quickly test a large part of the device.
HWSTATUS:	Run the sections which read and decode the hardware status of a device.
ID:	Run the sections which "identify" the device.
LOOPBACK:	Run whichever sections perform a loopback to the device(s). The type of loopback(s) performed will vary from diagnostic to diagnostic - the diagnostic developer determines which loopback(s) to perform when this mnemonic is given.
NON-DESTRUCTIVE:	Run only non-destructive sections or steps. This will be useful to users possessing the highest security level who wish to avoid inadvertently running any potentially destructive sections.

SELFTEST:	Run the sections which perform selftests on the device(s).
INTERACTIVE:	Run the diagnostic's interactive sections.

Examples of Mnemonic Use:

Please note that a user is not ezpected to know what section or step numbers a mnemonic replaces - the user may name a section or step number and the mnemonic that replaces it on the same line. This is not an error.

If there is a conflict among statements on the same line the rightmost statement will take precedence. Nested statements are not allowed.

ALL:

```
sc all

sc all(all)

sc all(+all)
```

Please note that the "+" and "-" operators may be used with any mnemonic even if, as in this case, they have no effect.

AUTO:

```
sc auto

sc +auto
```

Run previously named sections AND autodiagnosable sections.

```
sc 5/7(auto)
```

Only run those steps of sections 5 through 7 which are autodiagnosable.

```
sc 8(-auto)
```

Do not run the autodiagnosable steps of section 8. Do run the default steps of section 8 which are not autodiagnosable.

CLEAR:

```
sc clear
```

```
sc clear(4, 10/12)
```

Run steps 4 and 10 through 12 of whatever sections clear the device. If none of the sections which clear the device have any steps 4, 10, 11, or 12 nothing will be run and a message will be output to the user stating this.

DEFAULT:

```
sc 9/24(1/300)
```

Run steps 1 through 300 of sections 9 through 24.

```
sc 14(default)
```

Only run the default steps of section 14. These will be either the diagnostic system default steps or the steps which the user has designated to be defaults for section 14 using the `set` *command. The same effect would be achieved by using* `sc 14`.

```
sc 10(+default)
```

Run the default steps of section 10. The "+" has no effect here but is not an error.

```
sc default
```

Run all default steps of all default sections. The same effect can be achieved by invoking the diagnostic on a run command line without mentioning sections.

ERRLOG:

```
sc errlog
```

```
sc auto,errlog
```

Run the default steps of both the autodiagnosable and errlog sections.

FAST:

```
sc fast
```

```
sc 2/3,fast,6,10(-auto)
```

Run the default steps of sections 2, 3, and 6 and the default steps of all "fast" sections. Run the default steps of section 10 EXCEPT for those default steps which are also autodiagnosable. If section 10 is a "fast" section still only run its non-autodiagnosable default steps.

HWSTATUS:

```
sc hwstatus
```

```
sc auto, hwstatus(2/6)
```

Run the default steps of all autodiagnosible sections AND steps 2 through 6 of any hardware status sections. If any hardware status sections are autodiagnosible their default steps will be run.

ID:

```
sc id
```

```
sc id(-auto)
```

Run the non-autodiagnosable steps only of the "identify" sections. If the "identify" sections do not have any autodiagnosable steps the directive is ignored. If the "identify" sections are composed completely of autodiagnosable steps or are themselves autodiagnosable sections no operations will be performed.

LOOPBACK:

```
sc loopback
```

```
sc 3(-auto,+default), loopback(4/3,+auto)
```

If section 3 is a loopback section its autodiagnosable steps will be run since the "+auto" directive appears to the right of the "-auto."

NON-DESTRUCTIVE:

```
sc id
```

```
sc id(non-destructive)
```

Run the non-destructive steps only of the "identify" sections. If the "identify" sections are not divided into steps only run the sections if they are non-destructive.

```
sc non-destructive,5/8,3(+auto)
```

Although sections and steps will be run in numerical order the diagnostic user interface will accept them in any order. When section 3 is run in this example any steps previously set for it at a global level will be run along with all autodiagnostic steps. If no special steps have previously been set for it, its default and autodiagnostic steps will be run.

SELFTEST:

```
sc selftest
```

```
sc selftest(non-destructive), fast
```

INTERACTIVE:

```
sc interactive
```

```
sc 4/6(-auto), interactive,fast
```

The user is not expected to know which numbers correspond to each mnemonic so any of sections 4, 5, or 6 could be "interactive" and/or "fast" sections. Duplication is acceptable. If any of sections 4, 5, or 6 is either "interactive" or "fast" its autodiagnosable steps will be run if they are also default steps since the "interactive" and "fast" directives appear to the right of "-auto."

TRACE

The TRACE modifier enables/disables software tracing messages. The entering and exiting of procedures and functions are noted by the display of messages such as

```
Entering fee
  Entering fie
    Entering foe
      Entering fum
      Exiting fum
    Exiting foe
  Exiting fie
Exiting fee
```

Selective tracing may be done by using arguments corresponding to various types of software modules: diagnostic programs (PROG), diagnostic library routines (LIB), device access routines (DAR), and I/O system modules such as device drivers (SYS). Tracing of error conditions will occur if the ERROR modifier is used. Any of these may be used in combination.

Please note that the SYS argument will not result in trace messages on most systems. The "hooks" needed to make this work are only being placed in the appropriate code on a few systems.

Syntax:

```
TRACE [=] { OFF   }
          { ALL   }
          { PROG  }
          { LIB   }
          { DAR   }
          { SYS   }
          { ERROR }
```

Abbreviations and Alternate Tokens:

TR

Default Value:

off

Examples:

```
DUI> xdiag ldev 4 sc auto trace prog
```

```
DUI> ydiag ldev 2 tr off
```

Trace was probably "bound" to ydiag using the set *command. This turns tracing off during this one run of* ydiag.

```
DUI> run wdiag tr prog dar
```

Limitations:

- TRACE can only be effective when the code modules being traced are properly instrumented. The TRACE modifier cannot display progress through code which has not had the proper calls inserted during development.

Diagnostic Specific Parameters

Parameters, modifiers, or other information unique to a particular diagnostic program may, in some cases, be specified within the run command. Such information is placed within double quotes ("").

The information within the double quotes is passed directly to the invoked program - no checking is done by the DUI to determine the correctness or validity of the information being passed. Many diagnostic programs prompt for any specific information they need. This mechanism for passing information to a diagnostic is provided as a convenience to the diagnostics but is not used by all of them.

The **set** command may also be used to bind program specific parameters to a particular diagnostic. Every time that diagnostic is invoked the "set" information will be passed directly to the diagnostic.

Please see the diagnostic's manual regarding what information a particular diagnostic expects to receive in this manner.

Syntax:

"*information*"

Abbreviations and Alternate Tokens:

none

Default Value:

null

Examples:

```
DUI> run xdiag pdev 4.3.2 sc 7 (2/6,10) "some information"
DUI> set ydiag "some information"
```

Limitations:

- No checking is done before the information is passed to the diagnostic; the information is passed exactly as given.
- The information must be eighty (80) characters or less in length.

SETVAR

The SETVAR command allows the user to explicitly set the values of environmental variables which control various features of the user interface. The SETVAR command may also be used to examine the current values of these variables. The only environmental variable currently associated with the SETVAR command within the DUI is TRACE.

If SETVAR is given without a variable being specified, a list of all DUI environmental variables along with their current values will be displayed.

The variables modified by the SETVAR command affect the Diagnostic User Interface itself - NOT the diagnostic programs run from it.

Syntax:

SETVAR [*variable* [=] *value* [[,]...]]

Abbreviations and Alternative Tokens:

none

Modifiers:

variable	*value*
TRACE	OFF
	ALL
	PROG
	LIB
	SYS
	ERROR

The TRACE variable, used with the PROG modifier, results in the display of software tracing messages. The entering and exiting of the DUI's procedures and functions are noted by the display of messages such as

```
Entering fee
  Entering fie
    Entering foe
      Entering fum
      Exiting fum
    Exiting foe
  Exiting fie
Exiting fee
```

Tracing of error conditions will occur if the ERROR modifier is used. The errors which will be reported are diagnostic system internal errors. Some of these "errors" are expected conditions which are encountered during normal processing. The error messages which will be displayed as a result of TRACE ERROR are written for factory personnel troubleshooting the diagnostic system itself - they are not written for the end user and may be meaningless to many users.

Errors which would normally be reported to a user will continue to be reported whether TRACE ERROR is set or not.

PROG and ERROR may be used in combination.

Tracing will continue until the SETVAR TRACE OFF command is given or the diagnostic session is ended.

Command Examples:

```
DUI> setvar

DUI> setvar trace error

DUI> setvar trace prog

DUI> setvar trace error prog

DUI> setvar trace off
```

Limitations:

- The ALL, LIB, and SYS values of the SETVAR variable TRACE have no effect within the online diagnostics subsystem.

SET

The SET command allows the user to explicitly reset the system default values for *modifiers* which may be given on a **run** command line. Once set, these values are used for every run of every diagnostic during a diagnostic session unless the modifier values for an individual diagnostic are changed. These *modifier* values become the global defaults for the diagnostic session.

The SET command also lets the user bind particular command modifier values to individual diagnostic programs. Modifier values associated with an individual diagnostic through the use of the SET command become the default values for that diagnostic until the end of the diagnostic session.

Individual modifier values may be changed, temporarily, by naming those modifiers and their new values on the **run** command line of a diagnostic. At the conclusion of that one diagnostic run the modifier values will revert to their defaults.

If the SET command is given with a *program name* but no modifiers, a list of the current values of all possible modifiers to that *program* will be displayed.

If the SET command is given with neither *modifiers* nor a *program name*, the current global default value of each *modifier* will be displayed.

Syntax:

SET {[*program name*] *command modifier* [[,]...]}

Modifiers:

program name — The diagnostic program whose modifier default values are to be set or displayed.

command modifier — The modifier whose value is to be set at the global level. Any of the run command modifiers may be set in this way except the SECURITY modifier. The following is a complete list of the command modifiers which may be used with this command.

BACKGROUND
DEBUG
ERRCOUNT
ERRONLY
ERRPAUSE
HARDCOPY
INFILE
LDEV
LOOP
OUTFILE
PDEV
SECTIONS
TRACE
Diagnostic Specific Parameters

Please see the descriptions of each of these modifiers for an indication of values which may legitimately be assigned.

Command Examples:

```
DUI> set xdiag loop 5 errpause

DUI> set loop 20 errcount 4

DUI> set loop

DUI> set xdiag

DUI> set
```

Related Commands:

SHOWPARMS

SHOWACTIVE

The **SHOWACTIVE** command causes a list of all current diagnostic processes to be displayed along with their process identifier numbers. Each displayed diagnostic process is also noted as being "running," "suspended," or "aborting" and either "foreground" or "background."

A "running" program is one that is executing normally.

A "suspended" program is one which was "suspended" by use of the **SUSPEND** command and is waiting for a **RESUME** command to return to a running state.

An "aborting" program is one which is in the transient state occurring between the receipt of an **ABORT** command and the actual termination of the program.

"Foreground" and "background" show where the process is. Although every program will be designated as one or the other, the designation is most important for suspended programs. The behavior of the RESUME command may be affected by where a program was running when it was suspended.

Syntax:

```
SHOWACTIVE
```

Abbreviations and Alternate Tokens:

```
SA
```

Command Examples:

```
DUI> showactive

DUI> sa
```

SHOWDEFAULT

SHOWDEFAULT will cause a list of command modifiers and their default values to be displayed. The values displayed will be those the modifiers held when the diagnostic system was invoked; i.e., the initial diagnostic system default values for user settable modifiers will be displayed.

If SHOWDEFAULT is given with a list of command modifiers, the diagnostic system default values for those modifiers will be displayed. If SHOWDEFAULT is given without a list of modifiers or with the ALL argument, all user settable modifiers will be displayed along with their default values.

Syntax:

```
SHOWDEFAULT [ALL                          ]
            [command modifier [[,]...]]
```

Abbreviations and Alternate Tokens:

SD

Modifiers:

ALL	List all user settable modifiers and their default values.
command modifier	A user settable diagnostic modifier. The following is a complete list of these modifiers. Any valid run command modifier except SECURITY and the "Diagnostic Specific Parameters" modifier may be used with the SHOWDEFAULT command.

BACKGROUND
DEBUG
ERRCOUNT
ERRONLY
ERRPAUSE
HARDCOPY
INFILE
LDEV
LOOP
OUTFILE
PDEV
SECTIONS
TRACE

Command Examples:

```
DUI> showdefault

DUI> sd all

DUI> sd loop errpause errcount
```

Related Commands:

SET
SHOWPARMS

SHOWPARMS

SHOWPARMS will cause a list of user settable modifiers and their current values to be displayed.

The arguments the user gives to this command will determine which set of values for the modifiers will be displayed. If no argument is given the effect is the same as using the ALL argument.

If no argument is given or if the ALL argument is given the global default modifier values will be displayed along with the values the modifiers have for individual diagnostic programs (the values set with the SET command).

Syntax:

```
SHOWPARMS [GLOBAL                ]
          [LOCAL                 ]
          [ALL                   ]
          [program name [[,]... ]]
```

Abbreviations and Alternate Tokens:

SP

Modifiers:

GLOBAL	Display all modifiers with the values they hold globally. Each of these values will be either the diagnostic system default value for the modifier or the value for the modifier which has been explicitly named using the SET command.
LOCAL	Display only those modifiers and values explicitly set for individual diagnostic programs.
ALL	Display all modifiers with the values they hold globally. Also display those modifiers and values which have been reset for individual diagnostic programs using the SET command.
program name	Display all modifiers with the values they hold for the named program(s). These values may be the original diagnostic system default values, the global values set with the SET command, or values set explicitly for the named program using the SET command. In any case, the values displayed will be those which will be used when the named program is invoked unless the values are temporarily reset on the run command line.

Command Examples:

```
DUI> showparms

DUI> sp xdiag ydiag

DUI> sp global

DUI> showparms local

DUI> sp all
```

Related Commands:

```
SET
SHOWDEFAULT
```

SHOWSTATE

The **SHOWSTATE** command will cause the current system mode (single-user or multi-user) and the user's security capability to be displayed.

Syntax:

```
SHOWSTATE
```

Abbreviations and Alternate Tokens:

```
SS
```

Command Examples:

```
DUI> showstate

DUI> ss
```

Related Commands:

```
MODE
```

SUSPEND

The SUSPEND command causes the execution of a diagnostic program to stop. The program will remain "frozen" until a RESUME command is received at which time program execution will resume.

If the SUSPEND command is given without an argument and only one program is running, that one program will be suspended. If more than one program is running and the SUSPEND command is given without an argument, a list of programs and their program identifier numbers (*program-id*) will be displayed. The user may then repeat the SUSPEND command giving one or more of these *program-ids* as arguments.

Syntax:

```
SUSPEND [ALL                    ]
        [program id [[,]... ]]
```

Abbreviations and Alternate Tokens:

SUS

Modifiers:

ALL	Suspend every running program which was invoked by the user.
program id	The unique number identifying a particular run of a particular program. Used to tell the diagnostic system which program(s) to suspend.

Command Examples:

```
DUI> suspend

DUI> sus all

DUI> sus 23 17
```

Related Commands:

RESUME

UNLOCK

The **UNLOCK** command is used to explicitly unlock a device. This will release the device back to general access. **UNLOCK** may be used to release a device back to the system for general usage after the device has been fixed (or replaced) if it had previously been locked because it was defective.

If no **LDEV** or **PDEV** is given with the **UNLOCK** command a list of locked devices will be displayed. The devices listed will be malfunction locked.

Syntax:

```
UNLOCK [LDEV [=] logical device name ]
       [PDEV [=] physical path       ]
```

Modifiers:

omitted	A list of malfunction locked devices will be displayed.
LDEV *logical device name*	The logical name of the device.
PDEV *physical path*	The physical path to the device.

Command Examples:

```
DUI> unlock pdev 4/2.3

DUI> unlock ldev 12
```

USEFILE

The USEFILE command causes input to the diagnostic system to be gotten from the specified file rather than from a user's terminal. Reading from the file begins immediately; the file controls the diagnostic session until end of file is reached or the *usefile* is prematurely closed because an interrupt was received.

Usefiles are opened and processed under the assumption that each "line" in the file will consist of 80 or less characters, followed by a termination character or set of characters. The termination character is operating system dependent. . That is, each "line" has some combination of 80 characters or less (including spaces), followed by a carriage return, or by a line feed, or by a carriage return and a line feed, depending on the operating system. This file organization would be thought of as "80 byte fixed length record ASCII" on MPE XL.

An *usefile* with any "line" longer than 80 characters will not be read or processed correctly.

Usefiles may be nested; that is, a *usefile* may contain the USEFILE command. All open *usefiles* will be closed when an interrupt is received.

Syntax:

USEFILE [=] *filename*

Abbreviations and Alternate Tokens:

USE

Modifiers:

filename	The file from which the diagnostic system should get its input.

Command Examples:

```
DUI> usefile foocmds

DUI> use cmdfile
```

The usefile *foocmds* might contain the following:

```
run xdiag pdev 4.0.12 sc 3/10
run ydiag ldev 3 infile foo outfile fum
list type utility
mode single
run wdiag ldev 0 sc 5 outfile fee
```

Limitations:

- The USEFILE command can only be run in the foreground.
- It is assumed that the file contains commands and input sensible to the diagnostic system. If not, errors may occur.

Error Messages

The following are the error messages generated by the DUI.

305	***** COULD NOT READ MESSAGE AT DUI's PORT (DUIERR 305)**
CAUSE	Input, such as a command, was given to the DUI at a user's terminal. The input was queued to the DUI's message port. The attempt to pull the message off of the port so that it could be processed failed.
ACTION	Try again. If the failure occurs a second time get out of the diagnostic system any way possible. It may be necessary to log into another terminal and abort the DUI process - the EXIT command to the DUI is unlikely to work in the present case. Submit an SR against the DUI; give as much information as possible about the circumstances surrounding the error (was the command the first given during the diagnostic session or had previous input been processed correctly, had any unusual messages been printed when the DUI was invoked, etc.).

308	***** RECEIVED UNEXPECTED MESSAGE AT DUI'S PORT (DUIERR 308)**
CAUSE	Either a program sent an unrecognized request (something other than enable_intr_notify, disable_intr_notify, suspend_prog, or req_user_info) or the message type of a message pulled off of the DUI's port was not one the DUI could recognize.
ACTION	Use the command SETVAR TRACE ERROR. Repeat the command which resulted in the error the first time and note the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors. If the command used was RUN, it is possible that the error is in the diagnostic being run rather than in the DUI. However, if in doubt, submit the SR against the DUI rather than against the diagnostic.

309	***** COULD NOT READ FROM SET# ! MSG# ! (DUIERR 309)**
CAUSE	The attempt to retrieve a message from a diagnostic's message catalog failed. The most likely reason is that the message was never put into the catalog.
ACTION	Submit an SR against the diagnostic being run.

311 *** UNRECOGNIZED PROCESS - NOT IN LIST OF CURRENT PROCESSES (DUIERR 311)

CAUSE The DUI was either processing an output_data request, a program request, or a user reply to a program but could not find the process identifier (pid) in its (the DUI's) table of known processes. The pid is needed to process the request because, without it, the DUI has no way of knowing which catalog to pull the message to be printed from. (The DUI uses the pid as an "index" into a table which contains information about the process - including the file descriptor of the open message catalog belonging to the running program).

It is also possible that the pid was in the DUI's table, but the corresponding message catalog file descriptor was 0.

The pid is also needed when a "program reply" is processed. The reply received was a handshake to an interrupt received message which the DUI sent the program. If the pid associated with the handshaking cannot be found in the DUI's table the DUI cannot reset the "interrupt sent" flag for the process. Until this flag is reset, no more interrupts will be sent. (A moot point, since without the pid the DUI cannot know there is a process needing interrupt notification).

Lastly, the DUI will print this message when it has received a program request to enable_intr_notify, disable_intr_notify, suspend_prog, or request_user_info and cannot find the pid associated with the request (in the ipc message) in its table. Without the pid the DUI cannot process the request since the information needed to process the request is in the table.

The most likely reason for any of these situations to occur is that a timing problem in the underlying operating system caused the request or reply to be received by the DUI long after the DUI was informed that the diagnostic terminated. When the DUI is informed that a diagnostic is to terminate it always searches for and processes any messages associated with the diagnostic before processing the termination.

ACTION If the problem is reproducible, submit an SR against the DUI giving as much information as possible about what seems to be happening when the error occurs. If the problem ever occurs it is likely to be transient and difficult to reproduce - the SR should explain in as much detail as possible the context in which it happens.

312 ***** COULD NOT CREATE A PORT FOR THE DUI (DUIERR 312)**

CAUSE Call to build_port failed. Build_port returned an error status. (i.e. some status other than successful or dipc_dup_pname). If dipc_dup_pname had been returned, the DUI would just try again to build a port using a different port name.

ACTION There is something seriously wrong with either the inter-process communication system or with the diagnostic monitor. If the user can kill the monitor and its associated processes and remove the diagnostic ports and then restart the monitor the problem might go away. However, in most cases this will not be possible. Since the normal tracing and printing mechanisms have not been set-up this early in the code very little additional information can be gotten. Please log an SR against the DUI giving as much information as possible about what is happening. Include the operating system build version, whether the system had been recently rebooted or not, whether the diagnostic system had been run successfully before the problem appeared, etc.

---

319 ***** COULD NOT DO INITIALIZATION NEEDED FOR PRINTING (DUIERR 319)**

CAUSE Two possible causes.

1) could not form the file path for the DUI's message catalog

2) a message catalog for the DUI written in the system default language could not be found and the DUI was unable to determine if it was all right with the user to just use English language messages. (get_user_input returned with an unsuccessful status of some sort).

ACTION Check to see if a message catalog for the DUI is on the system. The catalog will be named CDUIFXXX where XXX is three digits corresponding to native language localization language codes. The default catalog is CDUIF000. If the catalog cannot be found or is not in the correct directory, have the system administrator put the file on the system with the correct access permissions. If the catalog is on the system, file an SR against the DUI.

---

320 ***** COULD NOT PRINT - PRINTING INSTRUCTIONS INCOMPLETE (DUIERR 320)**

CAUSE Neither a "print literal" nor a "print message from set" was specified when print_dui_msg was called. The DUI is the only code which uses this procedure.

ACTION Log an SR against the DUI giving a description of what was being done when the error was printed and a list of any messages which immediately preceded this one.

---

321 ***** COMMAND RECOGNIZED BY PARSING ROUTINES BUT NOT BY PROCESSING ROUTINES (DUIERR 321)**

CAUSE The command the user gave was in the table of command names recognized by the DUI's parser. However, it was not in the list of commands the DUI recognizes as being able to process.

ACTION Log an SR against the DUI listing this message and the command which was not recognized.

---

322 ***** INVALID LDEV SPECIFIED (DUIERR 322)**

CAUSE The ldev the user gave with the run command could not be converted into a pdev (which is actually used by the diagnostic system). The call to obtain_pdev returned a status of dac_invalid_ldev.

ACTION Determine the correct ldev (sysmap may be able to help if it is on the system) and run the diagnostic again or give the pdev instead of the ldev with the run command.

---

323 ***** COULD NOT CONVERT THE SPECIFIED LDEV INTO A PDEV (DUIERR 323)**

CAUSE The ldev the user gave with the run command could not be converted into a pdev (which is actually used by the diagnostic system). The call to obtain_pdev returned a non-successful status (some status other than dac_invalid_ldev or successful).

ACTION Determine the correct ldev (sysmap may be able to help if it is on the system) and run the diagnostic again or give the pdev instead of the ldev with the run command.

---

324 ***** FAILED TO SEND IPC MESSAGE TO A PORT (DUIERR 324)**

CAUSE Send_to_port failed (returned status something other than successful). The DUI was trying to send either one of the three program initialization messages or a user_interrupt notification to a diagnostic but failed for some unknown reason.

ACTION Re-run the diagnostic. If the failure occurs again, use the SETVAR TRACE ERROR command. Repeat the RUN command noting the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

325	*** COULD NOT LAUNCH THE DIAGNOSTIC (DUIERR 325)
CAUSE	The diagnostic system service launch_process failed to launch a diagnostic. The message printed just prior to this one should give some indication why (the message will be the one associated with the error status returned by launch_process).
ACTION	Whatever action is associated with the message printed immediately before this one.

326	*** UNRECOGNIZED IPC PROGRAM FUNCTION (DUIERR 326)
CAUSE	The DUI's procedure send_msg_to_program was asked to process a request other than prog_info1, prog_info2, prog_info3, or user_interrupt. Therefore, the DUI did not recognize the request.
ACTION	Re-run the diagnostic. If the failure occurs again, use the SETVAR TRACE ERROR command. Repeat the RUN command noting the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

327	*** CANNOT RUN THE DIAGNOSTIC (DUIERR 327)
CAUSE	The DUI tried to split the file path for the diagnostic - just in case the run command included the fully qualified name of the diagnostic (the file name along with the complete path to it in the file system). This needs to be done before the DUI checks the list of installed diagnostics to see if the name is in it. The call to split_file_path failed (non-successful status returned).
ACTION	Use the SETVAR TRACE ERROR command. Repeat the RUN command noting the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

328	*** CANNOT OPEN THE CATALOG (DUIERR 328)
CAUSE	The DUI could not open the diagnostic's message catalog. (The call to the service cat_open returned a non-successful status). The most likely cause is that there is no message catalog for the diagnostic written in the native language currently being used on the system.
ACTION	Make sure a properly generated message catalog for the diagnostic, written in the same language as the current system default language, has been installed.

329 ***** FAILED TO GET USER INPUT (DUIERR 329)**

CAUSE A call to get_user_input made from print_dui_msg failed. The DUI needed user input because the printing function set it received in print_dui_msg included a reply_pending function.

ACTION Use the SETVAR TRACE ERROR command. Repeat the same sequence of commands which led to the error noting the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

---

338 ***** ! COULD NOT BE SUSPENDED (DUIERR 338)**

CAUSE The request to the operating system to suspend the program failed. The program probably terminated between the the time the DUI checked to see if the program existed and the time the operating system received the suspension request.

ACTION If the error can be repeated, submit an SR against the DUI. Use the SETVAR TRACE ERROR command. Repeat the RUN command followed by the SUSPEND command noting the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

---

339 ***** ! COULD NOT BE RESUMED (DUIERR 339)**

CAUSE The request to the operating system to resume the program failed. The program probably terminated between the the time the DUI checked to see if the program existed and the time the operating system received the resumption request.

ACTION If the error can be repeated, submit an SR against the DUI. Use the SETVAR TRACE ERROR command. Repeat the RUN command followed by the RESUME command as before noting the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

---

340 ***** COULD NOT SUSPEND ! - IT IS NOT RUNNING (DUIERR 340)**

CAUSE The request to suspend a specified program failed because the program was not in a "running" state. The program is either already suspended or is aborting or terminating.

ACTION Nothing to be done.

341	***** COULD NOT RESUME ! - IT IS NOT SUSPENDED (DUIERR 341)**
CAUSE	The request to resume a specified program failed because the program was not in a "suspended" state.
ACTION	Nothing to be done.

343	***** THE SPECIFIED PROCESS IDENTIFIER IS NOT RECOGNIZED (DUIERR 343)**
CAUSE	The specified pid is not in the DUI's process table, so it cannot be suspended, resumed, or aborted.
ACTION	Use the SHOWACTIVE command to get a list of the pids of processes which may be suspended, resumed, or aborted.

344	***** THERE ARE NO RUNNING PROCESSES TO SUSPEND (DUIERR 344)**
CAUSE	The user gave the suspend command with the ALL option but the DUI's process table does not have any "running" processes in it so nothing can be suspended.
ACTION	Nothing to be done.

345	***** THERE ARE NO SUSPENDED PROCESSES TO RESUME (DUIERR 345)**
CAUSE	The user gave the resume command with the ALL option but the DUI's process table does not have any "suspended " processes in it so nothing can be resumed.
ACTION	Nothing to be done.

346	***** THERE IS MORE THAN ONE RUNNING PROCESS. PLEASE SPECIFY BY PROCESS IDENTIFIER WHICH OF THE FOLLOWING SHOULD BE SUSPENDED (DUIERR 346)**
CAUSE	The user gave the suspend command without an option, but the DUI's process table has more than one "running" process in it so the user must specify which process is to be suspended.
ACTION	Re-enter the SUSPEND command naming by pid the process to be suspended.

347 `*** THERE IS MORE THAN ONE SUSPENDED PROCESS. PLEASE SPECIFY BY PROCESS IDENTIFIER WHICH OF THE FOLLOWING SHOULD BE RESUMED (DUIERR 347)`

CAUSE The user gave the resume command without an option but the DUI's process table has more than one "suspended " process in it so the user must specify which process is to be resumed.

ACTION Use the SHOWACTIVE command to get a list of suspended processes and their pids. Re-enter the RESUME command naming the pid of the process to be suspended.

---

348 `*** AN INTERRUPT WAS RECEIVED FROM THE USER TERMINAL. (DUIERR 348)`

CAUSE A user interrupt was given at the DUI prompt.

ACTION Nothing to do.

---

349 `*** ! COULD NOT BE ABORTED (DUIERR 349)`

CAUSE The request to the operating system to abort the program failed. The program may have terminated between the time the DUI checked to see if it was running and the time the OS received the abort request.

ACTION Probably nothing to do. However, if this is repeatable (and not just the result of some very odd timing circumstance) submit an SR against the DUI. Use the SETVAR TRACE ERROR command. Repeat the same series of commands up to and including the ABORT command noting the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

---

350 `*** THERE IS MORE THAN ONE RUNNING OR SUSPENDED PROCESS. PLEASE SPECIFY BY PROCESS IDENTIFIER WHICH OF THE FOLLOWING SHOULD BE ABORTED (DUIERR 350)`

CAUSE The user gave the abort command without an option but the DUI's process table has more than one process in it so the user must specify which process(es) is(are) to be aborted.

ACTION Use the SHOWACTIVE command to get a list of running and suspended processes along with the associated pids. Re-enter the ABORT command naming one or more of the pids.

---

351 ***** THERE ARE NO PROCESSES (DUIERR 351)**

CAUSE The DUI's process table is empty. The user wanted to abort or suspend a process.

ACTION There is nothing to be done.

---

354 ***** RESERVE WORD LIST MESSAGE ! COULD NOT BE OBTAINED (DUIERR 354)**

CAUSE Call to CAT_READ to obtain the message containing a list of reserved words failed. A more specific error message will precede this one.

ACTION Do the action associated with the message printed immediately before this one.

---

355 ***** UNABLE TO OBTAIN RESERVED WORD FOR ITEM ! FROM MESSAGE ! (DUIERR 355)**

CAUSE The call to GET_TEXT to obtain a reserved word from a reserved word message failed. The DUI's catalog probably has a bug in it.

ACTION Submit an SR against the DUI.

---

356 ***** MESSAGE ! WAS NOT ACCEPTED BY SCANNER PACKAGE (DUIERR 356)**

CAUSE The procedure reset_scan failed while trying to accept the message given. This can usually only happen when an empty buffer is passed to the reset_scan function.

ACTION Submit an SR against the DUI.

---

359 ***** UNABLE TO READ DUI PROMPT FROM MESSAGE CATALOG (DUIERR 359)**

CAUSE The procedure was not able to successfully read the DUI prompt string from the message catalog. The fault might be in the procedure trying to read from the catalog or in the catalog itself.

ACTION Submit an SR against the DUI.

363	***** UNABLE TO READ DUI VERSION STRING FROM MESSAGE CATALOG (DUIERR 363)**
CAUSE	The procedure was not able to successfully read the DUI version string from the message catalog. The fault might be in the procedure trying to read from the catalog or in the catalog itself.
ACTION	Submit an SR against the DUI.

364	***** UNABLE TO READ DUI HEADER FROM MESSAGE CATALOG (DUIERR 364)**
CAUSE	The procedure was not able to successfully read the DUI header string from the message catalog. The fault might be in the procedure trying to read from the catalog or in the catalog itself.
ACTION	Submit an SR against the DUI.

365	***** UNABLE TO READ MONITOR VERSION FROM MESSAGE CATALOG (DUIERR 365)**
CAUSE	The procedure was not able to successfully read the diagnostic monitor's version display string from the message catalog. The fault might be in the procedure trying to read from the catalog or in the catalog itself.
ACTION	Submit an SR against the DUI.

366	***** UNABLE TO READ DUI WARNING MESSAGE FROM CATALOG (DUIERR 366)**
CAUSE	The procedure was not able to successfully read the DUI warning string from the message catalog. The fault might be in the procedure trying to read from the catalog or in the catalog itself.
ACTION	Submit an SR against the DUI.

370	***** UNABLE TO RETRIEVE MESSAGE (!) FROM SET (!) OF THE DUI MESSAGE CATALOG. (DUIERR 370)**
CAUSE	The service used to pull messages from the DUI's catalog failed to do so. The most likely reason (but not the only possible one) is that the message is not in the catalog.
ACTION	The specific reason the message could not be gotten from the DUI's catalog will be printed immediately before this message is. The action taken will depend on that message.

371 `*** UNABLE TO OBTAIN HELP INFORMATION FOR LDEV (!).`
`(DUIERR 371)`

CAUSE The ldev named by the user could not be translated into a corresponding pdev. The diagnostic system uses pdevs internally rather than ldevs. No processing can be done on any device nor can information be gotten about any device designated by ldev unless the diagnostic system can find a corresponding pdev.

ACTION Check to make sure that the ldev is correct and configured into the system. If it is, try again using the corresponding pdev rather than the ldev.

---

372 `*** UNABLE TO DETERMINE THE IDENTITY OF PDEV (!). FURTHER HELP INFORMATION CANNOT BE OBTAINED. (DUIERR 372)`

CAUSE Could not get a help message for the specified pdev.

ACTION The reason the help message could not be gotten will be printed immediately before this message. Appropriate action depends on what that previous message is.

---

373 `*** UNABLE TO OBTAIN HELP INFORMATION FOR PDEV (!).`
`(DUIERR 373)`

CAUSE The user asked for the "id" of a particular pdev but the diagnostic system failed to find the product number of the device represented by the pdev.

ACTION The actual reason for the failure will be printed immediately before this message. Appropriate action depends on what that message is.

---

374 `*** UNABLE TO RETRIEVE MESSAGE (!) FROM SET (!) OF THE CATALOG FOR (!). (DUIERR 374)`

CAUSE The service used to pull messages from diagnostic message catalogs failed to do so. The most likely reason (but not the only possible one) is that the message is not in the catalog.

ACTION The specific reason the message could not be gotten from the diagnostic's catalog will be printed immediately before this message is. The action taken will depend on that message.

375 ***** UNABLE TO OPEN THE MESSAGE CATALOG FOR (!).**
(DUIERR 375)

CAUSE Was unable to open a diagnostic's message catalog for reading. The most likely reason for this is that the catalog file has been accidentally purged from the diagnostic system. It is also possible that the diagnostic catalog has not been translated into the language currently being used by the diagnostic system.

ACTION Use the MODIFY command to reinstall the appropriate catalog. If the problem is that the catalog does not exist for the native language currently being used, either change the language being used to one for which a catalog exists or use the MODIFY command to install an existing catalog. When asked which language the catalog is written in, lie and give the current system language. This could cause problems for latter users, but will solve the immediate problem.

---

376 ***** UNABLE TO LOCATE THE MESSAGE CATALOG FOR (!).**
(DUIERR 376)

CAUSE Failed to fully qualify the message catalog name. (That is, the attempt to add the group and account or the directory path of the diagnostic system to the file name failed - the actual reason for the failure is unknown).

ACTION Use the command SETVAR TRACE ERROR. Repeat the command which resulted in the error the first time and note the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR noting all of the errors.

---

377 ***** (!) IS NOT CURRENTLY INSTALLED IN THE DIAGNOSTIC SYSTEM. (DUIERR 377)**

CAUSE The diagnostic was not found in the list of currently installed diagnostics on the present system.

ACTION If a typographical error was made, just redo the command correcting the spelling of the diagnostic name. Otherwise, install the diagnostic.

---

378 ***** YOU HAVE A DIAGNOSTIC SECURITY LEVEL OF !, BUT A MINIMUM DIAGNOSTIC SECURITY LEVEL OF ! IS NEEDED TO PERFORM THE REQUESTED FUNCTION (DUIERR 378)**

CAUSE The user does not have sufficient security to do requested function.

ACTION Log in as another user (one who has the necessary security level) or ask the system administrator to add the appropriate capability to your account.

---

379 ***** THE DIAGNOSTIC PROGRAM'S CATALOG DOES NOT CONTAIN INFORMATION ABOUT SECTIONS NEEDED TO PROCESS THIS REQUEST(DUIERR 379)**

CAUSE In the RUN command, the user named sections without explicitly naming steps for every section named. The diagnostic's catalog does not have the default step message (Set 2, msg 3). Or the user named sections for a diagnostic which does not have any - the catalog does not have a section message.

ACTION Use the HELP command to find out if the diagnostic does have sections. If it does, file an SR against the diagnostic; state that the diagnostic's message catalog is missing the required set 2, message 3. If the diagnostic does not have sections, run it again without naming any sections.

---

380 ***** THE TEST SYSTEM IS RUNNING SO NO TRACING OF THE DUI (OTHER THAN ERROR TRACING) CAN BE DONE AT THIS TIME (DUIERR 380)**

CAUSE The user requested some trace of the DUI, but the test system is already handling a codetest or a duitest and so cannot be called upon (through start_test_system) to also do a trace without generating an error.

ACTION Try again after the codetest or duitest is finished. The codetest or duitest is being run by a user under a different DUI on the same machine.

---

381 ***** INAPPROPRIATE I/O REQUEST (DUIERR 381)**

CAUSE The DUI was sent an info string as its input (implying that the DUI was invoked programmatically). Input from the user is needed, but no infile or usefile is open so input would have to come from the terminal; however, it is an error to try to get input from the terminal if an info string has been received - the DUI should just exit rather than getting more input at this point.

ACTION Create an infile or usefile with the input the DUI needs and correct the programmatic call to the DUI so that the command received in the info string refers to the infile or usefile.

---

501 ***** SYNTAX ERROR (DUISERR 501)**

CAUSE Something was syntactically wrong with a command or a sections/steps message in a diagnostic's catalog. The actual error may be printed ont before this message is printed.

ACTION Re-enter the command correctly (use HELP <command> SYNTAX to see the correct syntax). If the RUN command was used and the problem appears to be with the sections/steps, submit an SR against the diagnostic stating that there is a syntactical error in one of the sections/steps messages. Give an exact copy of the command as typed in the SR so that the engineer can narrow down the the problem to the most likely catalog message.

505	***** CANNOT PARSE MESSAGE CATALOG MESSAGE (DUISERR 505)**
CAUSE	A sections/steps message in a diagnostic's message catalog has an open parenthesis ("(") without a matching closing parenthesis (")").
ACTION	Submit an SR against the diagnostic explaining the problem. Give an exact copy of the command as originally typed in the SR so that the engineer can narrow down the problem to the most likely catalog message.

509	***** INVALID RANGE - ENDING VALUE < THAN STARING VALUE (DUISERR 509)**
CAUSE	A section or step number range of the form n/n was given in the RUN command or found in one of the diagnostic message catalog section/step messages. The second number in the range was smaller than the first number.
ACTION	If the range was given in the RUN command, re-enter the command using correct numerical values in the range. If the RUN command was correct, submit an SR against the diagnostic explaining the problem. Give an exact copy of the command as originally typed in the SR so that the engineer can narrow down the problem to the most likely section/step catalog message.

513	***** NO INPUT RECEIVED (DUISERR 513)**
CAUSE	A blank line rather than a command was given by the user. Not really an error.
ACTION	Ignore message and continue.

515	***** LINKED LIST BEING CHECKED OR MANIPULATED IS EMPTY (DUISERR 513)**
CAUSE	One of the DUI procedures which adds to or removes a section or step from the linked list which the DUI uses internally to determine which sections and steps the user wishes to run was passed a null linked section/step list.
ACTION	Use the command SETVAR TRACE ERROR. Repeat the command which resulted in the error the first time and note the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

522 ***** NEEDED A CLOSING PARENTHESIS ")" (DUISERR 522)**

CAUSE A set of steps in a diagnostic's message catalog section/step message or a step given with the HELP STEP command was preceded by an open-parenthesis ("(") but had no matching closing parenthesis (")").

ACTION If the HELP STEP command was at fault, re-enter the command with a closing parenthesis. Otherwise, enter the SETVAR TRACE ERROR command. Repeat the command which resulted in the error the first time and note the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the diagnostic noting all of the errors.

---

525 ***** NEEDED A CLOSING SQUARE BRACKET "]" (DUISERR 525)**

CAUSE The RUN command was given with an open square bracket ("[") used to signify the beginning of a list of steps. The step list was not terminated with a matching closing square bracket ("]").

ACTION Re-enter the RUN command correctly.

---

530 ***** PRINTING ENVIRONMENT EXPECTED (DUISERR 530)**

CAUSE The "env" keyword was given with either the HARDCOPY command or the RUN command with a HARDCOPY parameter, but no printing environment was specified.

ACTION Redo the command either specifying a printing environment or without the "env" keyword.

---

533 ***** EXPECTED THE NAME OF A FILE (DUISERR 533)**

CAUSE A command which requires a file name or a diagnostic name was given without naming a file or diagnostic.

ACTION Redo the command, giving the appropriate file name or diagnostic name. Enter the HELP <command> or HELP <command> SYNTAX for more information.

---

536 ***** LDEV SPECIFICATION EXPECTED (DUISERR 536)**

CAUSE The "ldev" keyword was given with a command, but no ldev was named.

ACTION Redo the command specifying an appropriate ldev.

---

538 ***** PDEV SPECIFICATION EXPECTED (DUISERR 538)**

CAUSE The "pdev" keyword was given with a command, but no pdev was named.

ACTION Redo the command specifying an appropriate pdev.

541	***** FILE NAME OR "OFF" EXPECTED HERE (DUISERR 541)**
CAUSE	The OUTFILE command was given or the RUN command was given with an OUTFILE parameter but without naming an outfile or specifying "off."
ACTION	Redo the command either specifying an outfile or "off."

545	***** "OFF" OR A NUMBER EXPECTED HERE (DUISERR 545)**
CAUSE	The RUN command was given with the "loop" parameter, but neither a number nor the word "off" was specified.
ACTION	Re-enter the command specifying either a number or the option "off."

546	***** "ON" OR "OFF" EXPECTED (DUISERR 546)**
CAUSE	The RUN command was given with either the "erronly" or "errpause" parameter, but the parameter was not followed by either the word "on" or "off."
ACTION	Re-enter the command specifying either "on" or "off" after the parameter.

553	***** TRACING OPTION EXPECTED HERE (DUISERR 553)**
CAUSE	The "trace" parameter was given with the SETVAR or RUN command but no tracing option was specified.
ACTION	Re-enter the command specifying a tracing option. Use the HELP TRACE command to get a list of tracing options.

555	***** QUOTED TEXT STRING EXPECTED HERE (DUISERR 555)**
CAUSE	Diagnostic specific parameters were assumed to be given with the RUN command since a double-quote (") was found in the command. However, there was something wrong with the quoted string - it might not have had a terminating quote or might have been null or might have had some other problem.
ACTION	Re-enter the command, correcting the string parameter.

560	***** APPROPRIATE MNEMONIC MESSAGE COULD NOT BE FOUND IN THE DIAGNOSTIC'S CATALOG (DUISERR 560)**
CAUSE	The user gave mnemonics in place of some section(s) or step(s) in the RUN command, but no message could be found in the diagnostic's message catalog which would allow the DUI to translate the mnemonic into the corresponding numbers.
ACTION	Re-enter the RUN command using section/step numbers instead of mnemonics. A list of the sections and steps in the diagnostic may be gotten by using the HELP <diagnostic name> SECTIONS command.

563 ***** COULD NOT FIND THE NUMBER BEING SEARCHED FOR IN THE CATALOG MESSAGE (DUISERR 563)**

CAUSE The DUI tried to remove a section or step number from the linked list of sections/steps it maintains while determining which sections and steps the user wishes to run with a diagnostic. The section/step number was not in the list. This is not a real error condition.

This happens when the DUI's list has been made using default sections and/or steps. The number not found was not found because it was not in the diagnostic's catalog message containing default sections or steps.

The user should never see this message.

ACTION Ignore the message if the diagnostic runs after the message appears. If the diagnostic does not run, submit an SR against the DUI stating that this error message appeared and giving an exact copy of the RUN command which resulted in the message appearing.

---

565 ***** NEITHER A NUMBER NOR A MNEMONIC WAS FOUND (DUISERR 565)**

CAUSE A number or mnemonic was searched for but not found in either a diagnostic catalog section/step message or in a command given by the user. This is not necessarily an error; for instance, it might just signify that the end of a list of sections or steps had been reached. However, this message will only be printed when the condition is an error - as when the user gives the RUN command with the "sections" parameter but fails to name any sections.

ACTION Re-enter the command, specifying a number where needed.

---

570 ***** EXPECTED A NUMBER OR MNEMONIC AFTER THE COMMA (DUISERR 570)**

CAUSE A number list was given with a command for which such a list is appropriate. One of the numbers was followed by a comma but no number appeared after the comma. If the RUN command was given, either a number or a mnemonic would have been acceptable after the comma, but neither appeared.

ACTION Re-enter the command. Either specify a number after the comma or do not type the last comma.

---

573 ***** EXPECTED A NUMBER AFTER THE RANGE SIGN (DUISERR 573)**

CAUSE A command was given with a number range ("n/n"), but the last number in the range was missing. No number followed the range sign so what was read was of the form "n/".

ACTION Re-enter the command specifying the last number in the range.

574	***** EXPECTED A NUMBER OR MNEMONIC AFTER THE SIGN (DUISERR 574)**
CAUSE	A "+" or "-" sign was found while trying to parse a command (most likely the RUN command), but the sign was not followed by a number or mnemonic.
ACTION	Re-enter the command either specifying a number or mnemonic after the sign or eliding the sign.

583	***** "OFF" OR A DEBUGGER NAME EXPECTED HERE (DUISERR 583)**
CAUSE	The RUN command was given with the "debug" parameter, followed by an equals sign ("debug="), but neither a debugger name nor the word "off" was found after the sign.
ACTION	Re-enter the command either specifying a debugger or giving the "off" keyword. The "debug" parameter may also be given without the equals sign - signifying that the default debugger should be used.

585	***** "OFF," "ON," "LONG," OR "SHORT EXPECTED HERE (DUISERR 585)**
CAUSE	The RUN command was given with the "errprint" parameter, followed by an equals sign ("errprint ="). But, none of the possible options to "errprint" was found after the equals sign.
ACTION	Use the HELP ERRPRINT command to see a list of all possible options and their use. Re-enter the command either naming an option after the "errprint" parameter or omitting the equals sign.

592	***** PROGRAM IS NOT INSTALLED (DUISERR 592)**
CAUSE	What was assumed to be a diagnostic name given with the RUN or HELP command was not found in the diagnostic system's list of installed diagnostics.
ACTION	If a typographical error was made re-enter the command. If the diagnostic is not installed, install it. A list of installed diagnostics may be gotten by using the LIST command.

593	***** OPTION IS INVALID FOR THIS COMMAND (DUISERR 593)**
CAUSE	A command was given containing a keyword which is valid for another command, but not for the given command.
ACTION	Re-enter the command using the correct syntax. The valid form for the command may be seen by using HELP <command> or HELP <command> SYNTAX.

---

595 ***** THE SECTION NUMBER WAS NOT FOUND IN THE MESSAGE CATALOG MESSAGE (DUISERR 595)**

CAUSE The DUI was unable to find a particular section number (probably given in the RUN command) in the diagnostic's message catalog message of valid sections.

ACTION Use the HELP <diagnostic> SECTIONS command to see a list of valid sections for the diagnostic. Re-enter the command.

---

598 ***** UNEXPECTED TOKEN FOUND (DUISERR 598)**

CAUSE While trying to parse a diagnostic catalog section/step message an unexpected punctuation mark was found (something other than a slash, open-parenthesis, close-parenthesis, or comma).

ACTION Use the SETVAR TRACE ERROR command. Repeat the RUN command noting the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the diagnostic noting all of the errors.

The RUN command may also he redone naming the sections in some alternate way. That is, if the original command used number ranges try listing the numbers explicitly. If the original command used mnemonics, try the numbers instead. If the original command relied on the defaults (i.e. didn't specify numbers or mnemonics), explicitly name the sections/steps to be run. The diagnostic catalog section/step messages which the DUI must parse in order to determine which sections/steps to run are dependent on the form of the input used in the RUN command. By changing the form of input, it is possible that the message with the error can be avoided. An SR should still be submitted though even if this works.

---

599 ***** UNRECOGNIZED INPUT (DUISERR 599)**

CAUSE A command was given which contained unrecognized garbage. Usually something extra and unexpected was found at the end of the command line. This could also appear if a valid parameter which can only be used once with the command is repeated.

ACTION Use the HELP <command> or HELP <command> SYNTAX to see the valid form of the command. Re-enter the command.

---

600 ***** PROGRAM IDENTIFIERS ALREADY GIVEN - "ALL" NOT VALID (DUISERR 600)**

CAUSE The ABORT command was given with the program identifiers of the programs to be aborted. Somewhere after the identifier list, the word "all" appeared. "All" is not valid if a program identifier has already been named.

ACTION Redo the command, eliding the "all."

---

601 ***** "ALL" ALREADY GIVEN - PROGRAM IDENTIFIERS ARE NOT VALID (DUISERR 601)**

CAUSE The ABORT command was given with the "all" parameter. Somewhere after the "all" a program identifier was named. Program identifiers are not valid if the "all" parameter has already been used.

ACTION Redo the command, deleting the program identifier(s).

---

602 ***** A BLANK IS NOT VALID AFTER THE ASTERISK (DUISERR 602)**

CAUSE A hardcopy environment was given beginning with an asterisk, but a space or tab appeared after the asterisk - this is not allowed.

ACTION Redo the command, removing the blank or tab.

---

603 ***** A TYPE (DIAGNOSTIC, EXERCISER, VERIFIER, OR UTILITY) IS NEEDED (DUISERR 603)**

CAUSE The user gave the LIST command with the "type" parameter but did not specify which type he wanted information about.

ACTION Use the HELP LIST or HELP LIST SYNTAX command to see a list of valid list types. Re-enter the LIST command.

---

604 ***** A PRODUCT NAME IS NEEDED (DUISERR 604)**

CAUSE The user gave the LIST command with the "product" parameter but did not specify which product he wanted information about.

ACTION Use the HELP LIST or HELP LIST SYNTAX command to see a list of valid products. Re-enter the LIST command.

---

605 ***** A PROGRAM NAME WAS EXPECTED AFTER THE COMMA (DUISERR 605)**

CAUSE The LIST command was given with program names. A comma followed one of the program names but was not itself followed by a program name in turn.

ACTION Redo the LIST command, either adding a diagnostic program name after the offending comma or omitting the comma.

---

606 ***** A PARAMETER WAS EXPECTED AFTER THE EQUALS SIGN (DUISERR 606)**

CAUSE An equals sign appearing after a command was not followed by a parameter.

ACTION Use HELP <command> or HELP <command> SYNTAX to see the valid form for the command. Re-enter the command.

607	***** A COMMAND IDENTIFIER WAS EXPECTED AFTER THE EQUALS SIGN (DUISERR 607)**
CAUSE	The DO or REDO command was given with the "cmd=" parameter. However, no command identifier appeared after "cmd="
ACTION	Redo the command, specifying a command identifier. Use the HELP DO or HELP REDO commands for a complete explanation of valid input.

608	***** AN EQUALS SIGN IS EXPECTED AFTER "CMD" AND "EDIT" (DUISERR 608)**
CAUSE	The DO or REDO command was given with the "cmd=" and/or the "edit=" parameter. However, the "=" was missing. The equals sign is not optional in this case.
ACTION	Redo the command, inserting the equals sign.

609	***** EDITING DIRECTIVES WERE EXPECTED HERE (DUISERR 609)**
CAUSE	The DO or REDO command was given without specifying editing directives after naming one of the parameters which must be followed by editing directives (";edit" or ";").
ACTION	Re-enter the command, specifying editing directives. Use the HELP DO or HELP REDO command for information about editing directives.

610	***** A STRING WAS EXPECTED HERE (DUISERR 610)**
CAUSE	The DO or REDO command was given with the "cmd=" and/or ";edit" or ";" parameter tokens. A legitimate command identifier/edit directive was not found after the parameter.
ACTION	Use the HELP DO or HELP REDO command to see a discussion of command identifiers and editing directives. Redo the DO or REDO command.

611	***** THIS STRING DOES NOT HAVE A TERMINATOR. (DUISERR 611)**
CAUSE	The DO or REDO command was given with a quoted string as a command identifier or editing directive. However, no terminating single or double quote was found to match the quote with which the string began.
ACTION	Redo the command, adding the terminating quotation character.

612	***** EXPECTED A NUMBER AFTER THE EQUALS SIGN.** **(DUISERR 612)**
CAUSE	The HELP <diagnostic> SECTION or HELP <diagnostic> SC command was given with an equals sign immediately following the SECTION or SC. No number appeared after the equals sign.
ACTION	Redo the command either inserting a number after the equals sign or deleting the equals sign.

613	***** EXPECTED A NUMBER HERE. (DUISERR 613)**
CAUSE	A number was not found in a command when one was required.
ACTION	Use HELP <command> or HELP <command> SYNTAX for information about the correct form of the command. Redo the command.

614	***** NEED A PROGRAM NAME (DUISERR 614)**
CAUSE	The RUN command was given, but no program was named.
ACTION	Redo the command, inserting the name of an installed diagnostic program. A list of installed programs may be gotten by using the LIST command. The proper form for the RUN command may be gotten by using the HELP RUN command.

616	***** THIS CANNOT BE A PROGRAM IDENTIFIER (DUISERR 616)**
CAUSE	A command was given with a program identifier as a parameter. But the program identifier given was too large to be a legitimate pid.
ACTION	Use the SHOWACTIVE command to get a list of active diagnostic processes and their identifiers. Re-enter the original command giving the correct pid.

802	***** CAN'T READ THE TEMPORARY DIAGNOSTIC LIST FILE.** **(DUIIERR 802)**
CAUSE	This error means that the temporary diagnostic list file could not be read from. The system may be corrupted at this point.
ACTION	Submit an SR explaining the problem.

803	***** THE FILE PURGE LIST FILE COULD NOT BE OPENED.** **(DUIIERR 803)**
CAUSE	This means the temporary purge file could not be opened. The file should exist.
ACTION	Submit an SR explaining the problem.

805	***** THE FILE PURGE LIST FILE COULD NOT BE CLOSED. (DUIIERR 805)**
CAUSE	The error means that the temporary purge file could not be closed.
ACTION	Submit an SR explaining the problem.

806	***** CAN'T DELETE THE FILE !. (DUIIERR 806)**
CAUSE	This means that a file that was involved with a deleted diagnostic could not be deleted. File protected may have been altered or the file actually doesn't exist which means the directory has been corrupted.
ACTION	You can delete the file by hand if it still exists but there is no reason that the file should have a problem being deleted. If you can't, submit an SR explaining the problem.

807	***** CAN'T DELETE THE TEMPORARY PURGE LIST FILE. (DUIIERR 807)**
CAUSE	This means that the temporary purge file could not be deleted.
ACTION	Submit an SR explaining the problem.

808	***** THE FILE PURGE LIST FILE COULD NOT BE READ. (DUIIERR 808)**
CAUSE	This means the temporary purge file was opened but can't be read from.
ACTION	Submit an SR explaining tbe problem.

902	***** THE TEMPORARY DIAGNOSTIC LIST FILE HAS BEEN CORRUPTED. (DUIIERR 902)**
CAUSE	This means that a copy of the temporary diagnostic list file was found to be corrupted during a modify catalog or modify file command.
ACTION	Submit an SR explaining the problem.

903	***** THE VERSION ! IS INVALID. TRY AGAIN. (DUIIERR 903)**
CAUSE	The program version in a catalog or one input from a modify code command has an invalid syntax. The syntax should be a.bb.cc. If the modify code command was chosen the version may be lower than the previous version.
ACTION	A higher version should be entered if in a modify code command otherwise the version in the catalog should be corrected.

904	***** CAN'T DELETE THE TEMPORARY MODIFY DIAGNOSTIC LIST FILE. (DUIIERR 904)**
CAUSE	This means that a temporary diagnostic list file could not be deleted during a modify command.
ACTION	Submit an SR explaining the problem.

908	***** ! IS ALREADY TESTED BY !. (DUIIERR 908)**
CAUSE	This means that a device which is already being tested by the program is being added to the list of devices the program tests. If this occurs during an install, then a duplicate device name exists in the device list.
ACTION	Please re-enter a device that is not tested by this program.

911	***** THE ! CATALOG ALREADY EXISTS. (DUIIERR 911)**
CAUSE	This means that a catalog is being added to the list of catalogs belonging to the program but the program already has a catalog with that language. If this is during an install then the catalog in the DUI's language needs to be edited.
ACTION	Please re-enter the language.

1001	***** THE TEMPORARY DIAGNOSTIC DIRECTORY FILE HAS BEEN CORRUPTED. (DUIIERR 1001)**
CAUSE	This means the copy of the diagnostic directory has been corrupted. It could also mean the real diagnostic directory is corrupted.
ACTION	Submit an SR explaining the problem.

1201	**••• THE CATALOG ROOT COULD NOT BE FORMED (DUIIERR 1201)**
CAUSE	This means the catalog root in the diagnostic directory could not be converted to a string.
ACTION	Submit an SR explaining the problem.

1202	**••• THE CATALOG NAME COULD NOT BE FORMED. (DUIIERR 1202)**
CAUSE	This means the destination name of a catalog file could not be formed.
ACTION	Submit an SR explaining the problem.

1204	**••• THE DUI CATALOG COULD NOT BE ACCESSED. (DUIIERR 1204)**
CAUSE	This means a message cannot be read from the DUI's catalog.
ACTION	Submit an SR explaining the problem.

1205	**••• THE FILE ! COULD BE OPENED BUT NOT PROPERLY CLOSED. (DUIIERR 1205)**
CAUSE	This means a file input by the user exists and was opened but the file header was corrupted or the file could not be closed properly.
ACTION	Submit an SR explaining the problem.

1206	**••• THE FILE ! EXISTS BUT IT COULD NOT BE OPENED PROPERLY. (DUIIERR 1206)**
CAUSE	The file exists but it could not be opened properly. The protection on the file may be bad.
ACTION	Submit an SR explaining the problem.

1209	**••• THE FILE PATB ! COULD NOT HE PARSED OR IS INCOMPLETE. (DUIIERR 1209)**
CAUSE	This means the path of the file is invalid for this operating system or it is incomplete.
ACTION	Correct the path if it is incorrect or invalid or incomplete. If not, submit an SR explaining the problem.

1212	***** ! COULD NOT BE COPIED TO !. (DUIIERR 1212)**
CAUSE	This is caused when the source cannot be copied to the destination. The source may not exist or can't be accessed. The destination may already exist or the disk space is used up.
ACTION	Submit an SR explaining the problem.

1213	***** THE DIAGNOSTIC SYSTEM FILE ! IS INVALID. (DUIIERR 1213)**
CAUSE	This means one of the system files to be added to the diagnostic list file is invalid.
ACTION	Submit an SR explaining the problem.

1215	***** THE DEVICE NAME ! IS TOO LONG. (DUIIERR 1215)**
CAUSE	This is caused when a device entered is too long. If it is during an installation then the catalog has a problem.
ACTION	Re-enter the device name or fix the catalog.

1218	***** THE FILE ! DOES NOT EXIST. (DUIIERR 1218)**
CAUSE	This means the input file does not exist.
ACTION	Input a valid existing file name that follows diagnostic naming rules discussed in the diagnostic development guide. If problems continue, submit an SR explaining the problem.

1219	***** THE DUI COULD NOT SCAN FOR USER COMMANDS. (DUIIERR 1219)**
CAUSE	This means the list of valid program types has been corrupted.
ACTION	Submit an SR explaining the problem.

1223	***** THE FILE PATH ! IS TOO LONG. (DUIIERR 1223)**
CAUSE	This means the file path of the input file is too long.
ACTION	Input a file with a valid shorter path. If the problems continues, submit an SR explaining the problem.

1224	***** THE TEMPORARY DIAGNOSTIC LIST FILE CANNOT BE APPENDED TO. (DUIIERR 1224)**
CAUSE	This means an internal temporary file cannot be appended to.
ACTION	Submit an SR explaining the problem.

1225	***** THE DIAGNOSTIC DIRECTORY FILE COULD NOT BE CLOSED PROPERLY. (DUIIERR 1225)**
CAUSE	This means the actual diagnostic directory file cannot be closed.
ACTION	Submit an SR explaining the problem.

1226	***** THE DIAGNOSTIC LIST FILE COULD NOT BE CLOSED PROPERLY. (DUIIERR 1226)**
CAUSE	This means the real diagnostic list file or a temporary one cannot be closed properly.
ACTION	Submit an SR explaining the problem.

1227	***** THE TEMPORARY DIAGNOSTIC DIRECTORY FILE COULD NOT BE CLOSED PROPERLY. (DUIIERR 1227)**
CAUSE	This means a temporary diagnostic directory file could not be closed properly.
ACTION	Submit an SR explaining the problem.

1228	***** THE TEMPORARY DIAGNOSTIC LIST FILE COULD NOT BE CLOSED PROPERLY. (DUIIERR 1228)**
CAUSE	This means a temporary diagnostic list file could not be closed properly.
ACTION	Submit an SR explaining the problem.

1229	***** THE USER CATALOG COULD NOT BE CLOSED PROPERLY. (DUIIERR 1229)**
CAUSE	This means the catalog the user input as the one with the same language as the DUI could not be closed properly.
ACTION	Submit an SR explaining the problem.

1231	***** THE DIAGNOSTIC DIRECTORY FILE COULD NOT BE CREATED. (DUIIERR 1231)**
CAUSE	This means the actual diagnostic directory file could not be created.
ACTION	Check if there is any disk space left.

1232	***** THE DIAGNOSTIC LIST FILE ! COULD NOT BE CREATED. (DUIIERR 1232)**
CAUSE	This means the actual diagnostic list file could not be created.
ACTION	Check to see if there is any disk space left.

1233	***** THE DIAGNOSTIC DIRECTORY FILE COULD NOT BE DELETED. (DUIIERR 1233)**
CAUSE	This means the old diagnostic directory file could not be deleted.
ACTION	Submit an SR explaining the problem.

1234	***** THE DIAGNOSTIC LIST FILE COULD NOT BE DELETED.** **(DUIIERR 1234)**
CAUSE	This means the old diagnostic list file could not be deleted.
ACTION	Submit an SR explaining the problem.

1235	***** THERE WAS NO DEVICE GIVEN IN THE MESSAGE CATALOG.** **(DUIIERR 1235)**
CAUSE	This means the user catalog had a device message which was empty. It can also be a problem parsing a device name.
ACTION	Correct the device name by editing the catalog. If the problem continues, submit an SR explaining the problem.

1236	***** THE DESTINATION FILE PATH COULD NOT BE CREATED.** **(DUIIERR 1236)**
CAUSE	This means the destination file path of a catalog, program, or downloadable file could not be parsed.
ACTION	Submit an SR explaining the problem.

1237	***** THE DIAGNOSTIC DIRECTORY FILE COULD NOT BE OPENED.** **(DUIIERR 1237)**
CAUSE	The actual diagnostic directory file could not be opened.
ACTION	Submit an SR explaining the problem.

1238	***** THE DIAGNOSTIC LIST FILE COULD NOT BE OPENED.** **(DUIIERR 1238)**
CAUSE	The actual diagnostic list file could not be opened.
ACTION	Submit an SR explaining the problem.

1239	***** THE TEMPORARY DIAGNOSTIC DIRECTORY COULD NOT BE OPENED. (DUIIERR 1239)**
CAUSE	A temporary diagnostic directory file could not be opened.
ACTION	Submit an SR explaining the problem.

1240	***** THE TEMPORARY DIAGNOSTIC LIST FILE COULD NOT BE OPENED. (DUIIERR 1240)**
CAUSE	A temporary diagnostic list file could not be opened.
ACTION	Submit an SR explaining the problem.

1241	***** THE USER CATALOG COULD NOT BE OPENED. (DUIIERR 1241)**
CAUSE	This means the user catalog matching the language of the DUI could not be opened.
ACTION	Submit an SR explaining the problem.

1242	***** THE TEMPORARY DIAGNOSTIC DIRECTORY FILE COULD NOT BE READ. (DUIIERR 1242)**
CAUSE	A temporary diagnostic directory file could not be read from.
ACTION	Submit an SR explaining the problem.

1243	***** THE USER CATALOG COULD NOT BE READ. (DUIIERR 1243)**
CAUSE	This means the user catalog matching the language of the DUI could not be read.
ACTION	Submit an SR explaining the problem.

1245 `*** THE TEMPORARY DIAGNOSTIC DIRECTORY COULD NOT BE WRITTEN TO. (DUIIERR 1245)`

CAUSE The temporary diagnostic directory file could not be written to.

ACTION Submit an SR explaining the problem.

1246 `*** ONE OF !'S DEVICE OPTIONS IS INVALID. (DUIIERR 1246)`

CAUSE This means that a device had a option that was invalid or a duplicate of one of its existing options. Valid options are AUTO and DECODE.

ACTION If this happened interactively re-enter the device and its correct options. If this happened during an install then the catalog needs to be fixed.

1247 `*** THE FILE PATH ! IS TOO LONG. (DUIIERR 1247)`

CAUSE This means the path to this file is too long.

ACTION Re-enter a shorter path.

1250 `*** THE DUI AND DIAGNOSTIC DIRECTORY FILE ARE INCOMPATIBLE. (DUIIERR 1250)`

CAUSE This means the version of the `DUI` used to create the current on-line diagnostic system is not the version of the current `DUI`.

ACTION Re-install the on-line diagnostic system from the install tape and add in any changes you have made.

1251 `*** THE TEMPORARY DIAGNOSTIC DIRECTORY COULD NOT BE DELETED. (DUIIERR 1251)`

CAUSE The temporary diagnostic directory file could not be deleted.

ACTION Submit an SR explaining the problem.

1252	***** THE TEMPORARY DIAGNOSTIC LIST FILE COULD NOT BE DELETED. (DUIIERR 1252)**
CAUSE	A temporary diagnostic list file could not be deleted.
ACTION	Submit an SR explaining the problem.

1253	***** THE DEFAULT SYSTEM PATH COULD NOT BE FORMED. (DUIIERR 1253)**
CAUSE	This means the default on-line diagnostic system location could not be parsed correctly.
ACTION	Submit an SR explaining the problem.

1254	***** THE FILE ADDITION LIST FILE COULD NOT BE OPENED. (DUIIERR 1254)**
CAUSE	A temporary diagnostic add file could not be opened.
ACTION	Submit an SR explaining the problem.

1255	***** THE FILE ADDITION LIST FILE COULD NOT BE READ. (DUIIERR 1255)**
CAUSE	A temporary diagnostic add file could not be read from.
ACTION	Submit an SR explaining the problem.

1256	***** THE FILE ADDITION LIST FILE COULD NOT BE WRITTEN TO. (DUIIERR 1256)**
CAUSE	A temporary diagnostic add file could not be written to.
ACTION	Submit an SR explaining the problem.

1257	***** THE FILE ADDITION LIST FILE COULD NOT BE CLOSED. (DUIIERR 1257)**
CAUSE	A temporary diagnostic add file could not be closed.
ACTION	Submit an SR explaining the problem.

1258	***** CAN'T FORM THE TEMPORARY DIAGNOSTIC DIRECTORY FILE PATH. (DUIIERR 1258)**
CAUSE	This means the file path for a temporary on-line diagnostic system file could not be parsed correctly.
ACTION	Submit an SR explaining the problem.

1259	***** CAN'T FORM THE TEMPORARY DIAGNOSTIC LIST FILE PATH. (DUIIERR 1259)**
CAUSE	This means the file path for a temporary on-line diagnostic system file could not be parsed correctly.
ACTION	Submit an SR explaining the problem.

1260	***** CAN'T FORM THE TEMPORARY DIAGNOSTIC PURGELIST FILE PATH (DUIIERR 1260)**
CAUSE	This means the file path for a temporary on-line diagnostic system file could not be parsed correctly.
ACTION	Submit an SR explaining the problem.

1261	***** CAN'T FORM THE TEMPORARY DIAGNOSTIC ADDLIST FILE PATH. (DUIIERR 1261)**
CAUSE	This means the file path for a temporary on-line diagnostic system file could not be parsed correctly.
ACTION	Submit an SR explaining the problem.

1262	***** THE FILE ADDITION LIST FILE COULD NOT BE DELETED.** **(DUIIERR 1262)**
CAUSE	A temporary diagnostic add file could not be deleted.
ACTION	Submit an SR explaining the problem.

1263	***** THE FILE ADDITION LIST FILE COULD NOT BE CREATED.** **(DUIIERR 1263)**
CAUSE	A temporary diagnostic add file could not be created.
ACTION	Check to see if there is disk space available.

1264	***** A FATAL DUI ERROR HAS OCCURRED ... ABORTING.** **(DUIIERR 1264)**
CAUSE	This means an internal error occurred in print_dui_message while attempting to get a reply from the user.
ACTION	Submit an SR explaining the problem.

1265	**COULD NOT RECEIVE MONITOR REPLY FOR A SEMAPHORE REQUEST (DUIERR 1265)**
CAUSE	The DUI could not receive an ipc message from the diagnostic monitor.
ACTION	Verify the diagnostic system is running correctly. If so, submit an SR explaining the problem.

1266	**COULD NOT SEND A SEMAPHORE REQUEST TO THE MONITOR (DUIERR 1266)**
CAUSE	The DUI could not send a message to the diagnostic monitor
ACTION	Verify the diagnostic system is running correctly. If so, submit an SR explaining the problem.

1267	**COULD NOT GET THE MONITOR PORT DURING A SEMAPHORE REQUEST (DUIERR 1267)**
CAUSE	The DUI could not get the diagnostic monitor's port number.
ACTION	Verify the diagnostic system is running correctly. If so, submit an SR explaining the problem.

4001	*** COULD NOT OPEN FILE ! (DUIERR 4001)
CAUSE	The DUI could not open a usefile for reading or an outfile for writing. The specific cause of the failure is not known.
ACTION	Use the SETVAR TRACE ERROR command. Repeat the command which resulted in the error and note the additional errors which will appear. An easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Unfortunately, OUTFILE and HARDCOPY might not work in this case. Submit an SR against the DUI noting all of the errors.

4002	*** COULD NOT READ FILE ! (DUIERR 4002)
CAUSE	Could not read from a usefile or infile which had previously been opened successfully for reading. The specific cause of the error is not known.
ACTION	First, check the file to see if there is anything obviously wrong with it (such as being in binary rather than ASCII). If the file seems to be correct use the SETVAR TRACE ERROR command. Repeat the command which resulted in the error appearing noting the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

4003	*** COULD NOT CLOSE FILE ! (DUIERR 4003)
CAUSE	Failed to close an infile, outfile, or usefile which had previously been opened successfully. The specific reason for the failure is not known.
ACTION	Check to see if the file still exists - it might have somehow been purged from the system after being opened. If the file still exists then use the SETVAR TRACE ERROR command. Repeat the command which resulted in the error appearing using a different infile, outfile, or usefile if necessary. Note the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called - OUTFILE might not work if the original problem involved an outfile). Submit an SR against the DUI noting all of the errors.

4004	*** FILE ! DOES NOT EXIST (DUIERR 4004)
CAUSE	Could not open a file because it did not exist.
ACTION	Create the file.

4005	*** SECURITY VIOLATION ON FILE ! (DUIERR 4005)
CAUSE	Could not open an outfile or usefile because the user does not have permission to read or write to it.
ACTION	Either change the permissions on the file, log in as a user with a higher security level, or have the system administrator increase your security capabilities.

4009 ***** ACCESS TO THE FILE CANNOT BE GRANTED BECAUSE IT IS CURRENTLY OPENED EXCLUSIVELY. (DUIERR 4009)**

CAUSE Either could not open or could not read a file because another user or process already has opened the file exclusively.

ACTION Retry the command after the file has been closed.

4010 ***** THE FILE IS NOT CURRENTLY OPEN (DUIERR 4010)**

CAUSE Could not read from, write to, or close a particular file because the file descriptor the procedure was trying to use was not recognized by the file system.

ACTION See if the file has already been closed or if it still exists. If the file is open then use the SETVAR TRACE ERROR command. Repeat the command which resulted in the error appearing using a different infile, outfile, usefile, or hardcopy device if necessary. Note the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called - these might not work if the original problem involved an outfile or hardcopy device). Submit an SR against the DUI noting all of the errors.

4012 ***** HARDCOPY ENVIRONMENT ! IS INVALID (DUIERR 4012)**

CAUSE There is something wrong with the environment specified with the HARDCOPY command or hardcopy parameter to the RUN command.

ACTION Re-enter the command either specifying a correct environment or not specifying any environment (i.e. use the default environment). The "correct" environment is operating system and device dependent; the diagnostic system has no control over this feature.

4015 ***** USEFILE IS NOT OPEN (DUIERR 4015)**

CAUSE Either could not read from or could not close a usefile because none were open.

ACTION See if the usefile has already been closed or has been purged from the system. If neither of these is the case then use the SETVAR TRACE ERROR command. Repeat the command which resulted in the error appearing using a different usefile if necessary. Note the additional errors which will appear (an easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called). Submit an SR against the DUI noting all of the errors.

4017 ***** INVALID COMMAND, ONLY RUN COMMAND AVAILABLE FOR CODETEST (DUIERR 4017)**

CAUSE Successfully parsed a command line from a codetest file, however, the command was not the run command. The only DUI command allowed within a codetest file is the run command.

ACTION Remove the command from the codetest file.

4031	***** FILE ! ALREADY EXISTS (DUIERR 4031)**
CAUSE	The OUTFILE command or the RUN command with the outfile parameter was given. But, an outfile could not be created using the name specified for it because a file with that name already exits.
ACTION	Redo the command, giving a different name for the outfile which is to be created.

4032	***** UNABLE TO CREATE THE SPECIFIED OUTFILE (DUIERR 4032)**
CAUSE	Could not create the specified outfile for some reason. The actual reason for the failure will be printed immediately before this message.
ACTION	Action taken will depend on the reason for the failure; see the cause/action statement for the message which will immediately precede this one.

5009	***** THE DUI RECEIVED A NEGATIVE OR ZERO VALUE FOR REDO STACK SIZE (DUIDOREDO 5009)**
CAUSE	The REDOSAVE or REDOSIZE command was given with a negative redo stack size specified. The redo stack size must be a value greater than zero.
ACTION	Repeat the command, giving a positive integer value for the size of the history stack.

5021	***** THE DUI TRIED TO PLACE A COMMAND STRING INTO THE SCANNER INPUT BUFFER, BUT ENCOUNTERED AN ERROR. (DUIDOREDO 5021)**
CAUSE	The DO or REDO command was given and the command to be repeated was successfully gotten from the history stack. However, the attempt to place the command in a buffer and point to the first character in the command so that it could be parsed failed. This is a particularly strange error which should never occur.
ACTION	Use the SETVAR TRACE ERROR command. Repeat the command which resulted in the error and note the additional errors which will appear. An easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Submit an SR against the **DUI** noting all of the errors.

5022 ***** THE DUI TRIED TO LOCATE THE NUMERIC COMMAND ID ON THE COMMAND HISTORY STACK, BUT THE NUMBER WAS NOT THERE. (DUIDOREDO 5022)**

CAUSE The DO or REDO command was given with a numeric command identifier, but no matching command could be found in the history stack.

ACTION Use the LISTREDO command to see the diagnostic system history stack. Then re-enter the command, giving a correct command identifier. Information about command identifiers may be found by using the HELP DO or HELP REDO commands.

5023 ***** THE DUI TRIED TO LOCATE THE STRING COMMAND ID ON THE COMMAND HISTORY STACK, BUT THE STRING WAS NOT THERE. (DUIDOREDO 5023)**

CAUSE The DO or REDO command was given with a command identifier string, but no matching command could be found in the history stack.

ACTION Use the LISTREDO command to see the diagnostic system history stack. Then re-enter the command, giving a correct command identifier. Information about command identifiers may be found by using the HELP DO or HELP REDO commands.

5024 ***** THE DUI TRIED TO SAVE THE COMMAND HISTORY STACK, BUT THERE WAS NOTHING ON THE STACK. (DUIDOREDO 5024)**

CAUSE The REDOSAVE command was given, but the diagnostic history stack could not be saved because it was empty.

ACTION If the REDOSAVE command is the first command given during the diagnostic session the history stack should be empty - nothing can or should be done to save the stack. If some commands have preceded the REDOSAVE command, then an SR should be submitted against the **DUI**.

Use the SETVAR TRACE ERROR command. Repeat the REDOSAVE command noting the additional errors which will appear. An easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Submit the SR against the **DUI** noting all of the errors.

6006 ***** INVALID PARAMETERS WERE PASSED. (DUIERR 6006)**

CAUSE The MODE command was given with both the "multi" and the "single" parameters.

ACTION Repeat the MODE command using either the "multi" or the "single" parameters, but not both.

6008 ***** LEVEL 0 PRIVILEGE IS NEEDED TO CHANGE THE MODE.**
(DUIERR 6008)

CAUSE The MODE SINGLE command was given while the system was in multi-user mode. The user has too low a security capability (less thau diag_level_0) for the request to be honored.

ACTION Log in as a user with a higher security level, or have the system administrator increase your security capabilities.

6015 ***** THE SECURITY LEVEL NUMBER WAS INVALID. (DUIERR 6015)**

CAUSE The MODE SINGLE command was given by a user with too low a security level for the request to be honored. The DUI was prevented from telling the user what his/her security level was because the security level could not be translated iuto an ASCII printable string.

ACTION The actual cause of the failure will be printed immediately before this message; see the cause/action statement for that message. However, if that proves unsatisfactory then invoke the SETVAR TRACE ERROR command. Repeat the MODE SINGLE command noting the additional errors which will appear. An easy way to get a copy of these would be by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Submit the SR against the DUI noting all of the errors.

---

801 ***** COULD NOT LAUNCH COMMAND INTERPRETER (DUIERR 801)**

CAUSE The CI command was given, but the DUI was unable to invoke the command interpreter.

ACTION Invoke the SETVAR TRACE ERROR command. Repeat the CI command and note the additional errors which will appear. An easy way to get a copy of these is by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Submit the SR against the DUI noting all of the errors.

9001 ***** ONLY TEST NUMBERS GREATER THAN ZERO ARE ALLOWED**
(DUICTERR 9001)

CAUSE The CODETEST command was given with a "0" or a negative number explicitly specified as a test number to be processed. Only positive integer values greater than 0 may be used as test numbers.

ACTION Repeat the CODETEST command using correct test numbers. If a CODETEST test script contains tests whose numbers are not integer values greater than 0, correct the script.

9002 ***** INVALID TEST RANGE - THE LAST NUMBER IN A RANGE MUST BE GREATER THAN THE FIRST (DUICTERR 9002)**

CAUSE The CODETEST command was given with a test number range ("n/n"), but the last number in the range was smaller than the first.

ACTION Repeat the command correcting the test range so that the second number in the range is greater than the first.

9501 ***** UNABLE TO CONVERT THE SPECIFIED LDEV (!) TO ITS ASSOCIATED PDEV. (DUIERR 9501)**

CAUSE The UNLOCK command was given with an ldev parameter, but the diagnostic system could not convert the specified ldev into its corresponding pdev. Pdevs rather than ldevs are used internally in the diagnostic system so the failure to translate the given ldev into a pdev precludes unlocking the device.

ACTION Repeat the command using a pdev parameter rather than the ldev parameter.

9502 ***** UNABLE TO UNLOCK PDEV (!). (DUIERR 9502)**

CAUSE The UNLOCK command was given. The DUI received a garbled handshaking reply from the underlying process which should have performed the "unlock." Because the handshake was garbled, the DUI is assuming that the device was not unlocked.

ACTION Run something which uses the device (such as its diagnostic) to see if it is really locked or not. Submit an SR against the DUI. If the device is actually still locked the SR is more urgently needed than if it is not.

Invoke the SETVAR TRACE ERROR command. Repeat the UNLOCK command and note the additional errors which will appear. An easy way to get a copy of these is by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Submit the SR against the DUI noting all of the errors.

10000 ***** THE FOLLOWING MESSAGE FROM THE DIAGNOSTIC PROGRAM'S CATALOG CONTAINS A SYNTAX ERROR (DUIERR 10000)**

CAUSE When the user enters the RUN command, the DUI obtains information concerning the sections and steps of the diagnostic from the diagnostic's message catalog file. The DUI detected a syntax error during the parsing of these section/step messages. This error message will be followed by a copy of the line from the catalog in which the syntax error was detected along with a detailed message describing the problem.

ACTION Submit an SR against the diagnostic including all the information given in this message and the two following.

The RUN command may also be redone naming the sections in some alternate way. That is, if the original command used number ranges try listing the numbers explicitly. If the original command used mnemonics, try the numbers instead. If the original command relied on the defaults (i.e. didn't specify numbers or mnemonics), explicitly name the sections/steps to be run. The diagnostic catalog section/step messages which the DUI must parse in order to determine which sections/steps to run are dependent on the form of the input used in the RUN command. By changing the form of input, it is possible that the message with the error can be avoided. An SR should still be submitted though even if this works.

10001 ***** DUI WAS NOT ABLE TO READ MESSAGE 3 OF SET 2 FROM THE DIAGNOSTIC PROGRAM'S CATALOG FILE (DUIERR 10001)**

CAUSE The RUN command was invoked. The DUI was not able to read the above mentioned message from the diagnostic program's message catalog. This message contains a list of the default sections for the diagnostic and is needed to determine which sections to run if none were named with the RUN command. An error message indicating the specific problem will follow this message.

ACTION Repeat the RUN command explicitly naming the sections to be run. A list of these may be seen by using the HELP <diagnostic> SECTIONS command.

Additional action may be needed depending on the error message which will follow this one. For example, if that message indicates that the set 2, message 3 message does not exist then an SR should be entered against the diagnostic.

10002 ***** DUI WAS NOT ABLE TO READ MESSAGE 2 OF SET 2 FROM THE DIAGNOSTIC PROGRAM'S CATALOG FILE (DUIERR 10002)**

CAUSE The RUN command was invoked. The DUI was not able to read the above mentioned message from the diagnostic program's message catalog. This message contains a list of the default steps for every section in the diagnostic and is needed to determine which steps to run if some steps were not explicitly named within the RUN command. An error message indicating the specific problem will follow this message.

ACTION Repeat the RUN command explicitly naming all steps to be executed. A list of these may be seen by using the HELP <diagnostic> SECTIONS command.

Additional action may be needed depending on the error message which will follow this one. For example, if that message indicates that the set 2, message 2 message does not exist, then an SR should be entered against the diagnostic.

10003 ***** MESSAGE 3 OF SET 2 FROM THE DIAGNOSTIC PROGRAM'S CATALOG IS A BLANK LINE (DUIERR 10003)**

CAUSE The RUN command was invoked on a diagnostic which contains sections, but no sections were named. The DUI found that the diagnostic catalog message which should have had a list of the default sections for the diagnostic was a blank line. This is not valid. The DUI cannot determine which sections to run in this case.

ACTION Repeat the RUN command explicitly naming all the sections to be executed. A list of these may be seen by using the HELP <diagnostic> SECTIONS command.

An SR should also be entered against the diagnostic.

10004 ***** MESSAGE 2 OF SET 2 FROM THE DIAGNOSTIC PROGRAM'S CATALOG IS A BLANK LINE (DUIERR 10004)**

CAUSE The DUI found that the above mentioned message was a blank line. This is not valid. The DUI is not able to validate the section and step numbers entered by the user without this information.

The RUN command was invoked on a diagnostic which contains sections and steps, but either no sections were named or some section was named without specifying steps. The DUI found that the diagnostic catalog message which should have had a list of the default steps for each section in the diagnostic was a blank line. This is not valid. The DUI cannot determine which steps to run in this case.

ACTION Repeat the RUN command explicitly naming all the sections and steps to be executed. A list of these may be seen by using the HELP <diagnostic> SECTIONS command.

An SR should also be entered against the diagnostic.

11010 ***** ERROR - COULD NOT PERFORM REQUESTED OPERATION ON THE SPECIFIED DIAGNOSTIC SYSTEM PROCESS (DUISYSERR 11010)**

CAUSE 1) The DIAGSYSTEM SHOWACTIVE command was given. The proper response should have been for the DUI to print out a list of active diagnostic system processes and their program identifiers. However, one of the identifiers could not be printed because it could not be converted into an ASCII printable string.

2) Either the DIAGSYSTEM RUN <program name> or the DIAGSYSTEM ABORT <program identifier> command was given. The DUI received a handshaking reply from the underlying process which should have performed the "run" or "abort" stating that the action was not completed successfully. The actual reason for the failure will be printed immediately before this message.

ACTION If the DIAGSYSTEM SHOWACTIVE command was given invoke the SETVAR TRACE ERROR command. Repeat the DIAGSYSTEM SHOWACTIVE command and note the additional errors which will appear. An easy way to get a copy of these is by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Submit the SR against the DUI noting all of the errors.

If the DIAGSYSTEM RUN <program name> or DIAGSYSTEM ABORT <program identifier> command was used then follow the action specified for the message which will immediately precede this one.

11011 ***** ERROR - COULD NOT ABORT THE SPECIFIED DIAGNOSTIC SYSTEM PROCESS (DUISYSERR 11011)**

CAUSE The DIAGSYSTEM ABORT <program identifier> command was given. The DUI received a garbled handshaking reply from the underlying process which should have performed the "abort." Because the handshake was garbled, the DUI assumes that the process was not aborted.

ACTION Use the CI command and whatever process status command works on the current operating system to see if the process has actual been aborted. The DIAGSYSTEM SHOWACTIVE command might also work, but its results could be suspect in the current case.

Submit an SR against the DUI. If the process was not actually aborted the SR is more urgently needed than if it was.

Invoke the SETVAR TRACE ERROR command. Repeat the DIAGSYSTEM ABORT <program identifier> command and note the additional errors which will appear. An easy way to get a copy of these is by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Give a complete list of the resultant errors in the SR submitted against the DUI.

11012 ***** ERROR - COULD NOT RUN THE SPECIFIED DIAGNOSTIC SYSTEM PROCESS (DUISYSERR 11012)**

CAUSE The DIAGSYSTEM RUN <program name> command was given. The **DUI** received a garbled handshaking reply from the underlying process which should have performed the "run." Because the handshake was garbled, the **DUI** assumes that the process was not launched.

ACTION Use the CI command and whatever process status command works on the current operating system to see if the process is running. The DIAGSYSTEM SHOWACTIVE command might also work.

Submit an SR against the **DUI**. If the process was not actually launched the SR is more urgently needed than if it was.

Invoke the SETVAR TRACE ERROR command. Repeat the DIAGSYSTEM RUN <program name> command and note the additional errors which will appear. An easy way to get a copy of these is by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Give a complete list of the resultant errors in the SR submitted against the **DUI**. It might be necessary to abort the diagnostic system process before repeating the DIAGSYSTEM RUN <program name> command.

11013 ***** ERROR - CANNOT LOCATE THE LIST OF DIAGNOSTIC SYSTEM PROCESSES (DUISYSERR 11013)**

CAUSE The DIAGSYSTEM SHOWACTIVE command was given. The **DUI** could not find the diagnostic system table which contains all the information about diagnostic system processes.

ACTION Invoke the SETVAR TRACE ERROR command. Repeat the DIAGSYSTEM SHOWACTIVE command and note the additional errors which will appear. An easy way to get a copy of these is by using the OUTFILE and/or HARDCOPY commands - preferably before the SETVAR command is called. Submit an SR against the **DUI** giving a complete list of the errors.

Contents

3

Memory Diagnostic

Introduction

The **MEMDIAG** diagnostic is designed to detect general memory device failures in PA-RISC Memory Array cards, PA-RISC Memory Controller cards, and in PA-RISC Memory Controller/Memory Array combination cards. The diagnostic offers a selection of high level tests that will call lower level routines to access the hardware. If an error is encountered, the Field Replaceable Unit (FRU) will be pinpointed and the test will continue to execute, if possible.

The diagnostic is divided into sections that the user can decide whether or not to run. The user can enter the numbers for the test sections to be run as parameters for the **run** command, used to invoke the diagnostic.

After the diagnostic has cycled through all the memory pages it was able to access, **MEMDIAG** will terminate, and the user will be returned to the Diagnostic User Interface (DUI).

Defects and Enhancements

Submit defect reports and enhancement requests concerning this diagnostic through the STARS database referencing product number 30600-10009.

Minimum Configuration

The hardware required to run **MEMDIAG** is the PA-RISC system to be tested, consisting of the memory controller card and the memory array card, or of the memory controller/memory array combination card, plus any other equipment required to get the operating system up and running.

Operating Instructions

There is no security level check mechanism within **MEMDIAG**. The DUI checks the user's security level before initiating **MEMDIAG**. Refer to the section on the DUI for a detailed description of user capabilities.

Default Tests

If the user does not specify any sections or steps to be run, the default sections are executed, based on the diagnostic mode which has been selected by the Online Diagnostic subsystem. (See the Online Diagnostics Overview discussion of diagnostics modes for details.) The default test is:

Section 9 Trouble Tree

RUN Command

To bring up the Online Diagnostic subsystem or DUI, enter the following command to the HP-UX system prompt:

```
%/usr/diag/bin/sysdiag
```

The system responds with the following Diagnostic User Interface prompt:

```
DUI >
```

Typing **HELP** causes a summary of the DUI and its commands to be printed. Refer to the DUI Section of this manual for details.

To run the diagnostic, enter:

```
DUI >RUN MEMDIAG
```

no parameters required to load test suite (though parameters can be given)

The user can enter **help memdiag section** and see a menu that briefly describes the section and steps available in the diagnostic.

Parameters to the RUN command

At the invocation of the online diagnostics subsystem diagnostic, the user can define several run time parameters. The **loop** parameter is used to control the number of times the selected sections and steps are executed.

For further information on parameters to the run command, refer to the online diagnostics subsystem DUI ERS. All parameters available in the run command are accepted as parameters when executing this diagnostic.

Test Execution

After the user enters the DUI run command, the diagnostic **MEMDIAG** will be invoked. Once the program is started, the following header and welcome message will be displayed on the screen.

```
*****************************************************************************
*****                                                                   *****
*****                                                                   *****
*****                    Memory Array Diagnostic                        *****
*****                                                                   *****
*****              (c) Copyright Hewlett-Packard Company 1987           *****
*****                          All Rights Reserved.                     *****
*****                                                                   *****
*****                          Version A.00.00                          *****
*****                                                                   *****
*****                                                                   *****
*****************************************************************************
Welcome, Today is Thursday, January 5, 1987 at 10:50:30
```

Messages will be sent to the screen informing the user of the current test being executed and, upon its completion, the results. If the user did not specify any sections or steps, the default sections and steps will be executed. In the case of **MEMDIAG**, the default section is the Trouble Tree (section 9). Each section has default steps set that will be run if the section is selected without specifying any steps. These default steps are listed under their respective sections in this document.

Whenever **MEMDIAG** prompts the user for input, the user may enter "exit" to terminate the **MEMDIAG** diagnostic and jump back into the DUI system environment, or enter "suspend" to temporarily suspend **MEMDIAG** execution. Either the entire word or any number of characters sufficient to uniquely identify the command may be entered.

The user may then perform tasks within the Diagnostic User Interface (DUI) and later resume/abort execution of **MEMDIAG**.

Early Termination

MEMDIAG runs extensive pattern tests that usually take more than a half hour to complete. Because of this, several options are available for exitiug the testing sequence before it terminates in a normal fashion.

1. If time is a critical factor in your testing, go into Section 8, User Interactive, and select specific memory ranges to test. This is a more direct approach, and is much more time-efficient: you can test memory pages mapping to a specific FRU.
2. Before any test is invoked, the following prompt will be displayed to ensure that the user is aware of the time these tests take to complete. The user can then choose to continue testing or not:

```
***
***       WARNING
***
***  The section you have selected will run memory tests that
***  take more than half hour to complete (on average).
***
***  If you don't have the time now, go into USER INTERACTIVE (Section 8)
***  and setup your own memory test by selecting parameters that will
***  specifically test certain ranges and, therefore, shorten test duration.
***
***
***  Do you want to continue with this Section now? (y/n) [y]:
***
```

3. The user may enter **exit** or **suspend** at any **MEMDIAG** prompt: this will either terminate or temporarily suspend execution of **MEMDIAG**.
4. If a test is in progress, aud the user would like to interrupt its execution, a system-dependent sequence of characters can be entered to interrupt the normal execution cycle of **MEMDIAG**. The interrupt sequences are <CNTL-C> for HP-UX and <CNTL-Y> for MPE XL.

Detailed Test Descriptions

The following is a detailed discussion of **MEMDIAG** sections and steps. While in the Diagnostic User Interface, the user may view a menu of sections/steps by using the **help memdiag section** command. The menu gives a brief summary of each section and step. The following table lists the sections/steps of this diagnostic:

MEMDIAG Menu

```
Section 1 :          Initialize Card

    step 10 :            initialize memory controller(s)
    step 11 :            memory to all ones
    step 12 :            memory to all zeros


Section 2:           Identify

    step 20 :           Configuration Information
    step 21 :            Identify Information


Section 3:           Status


Section 4:           Memory tests (1 Second Wait per Page)

    step 40 :            address uniqueness/complement
    step 41 :            walking ones/zeros pattern
    step 42 :            alternating ones/zeros pattern
    step 43 :            ALL section 4 patterns


Section 5:           Memory tests (5 Second Wait per Page)

    step 50 :            address uniqueness/complement
    step 51 :            walking ones/zeros pattern
    step 52 :            checkerboard/complement pattern
    step 53 :            ALL section 5 patterns


Section 6:           Verify/Sweeper\(Not implemented yet)


Section 7:           EDC Logic test\(Not implemented yet)


Section 8:           USER INTERACTIVE


Section 9:           Trouble Tree (Default Test)
```

Section 1 : INITIALIZE CARD

This section will initialize the memory space to all ones or all zeros and clear the syndrome register(s) of all memory controller(s) in the system. Step 12 will write all zeros to memory then read it back and verify proper storage. Step 11 will write all ones to memory then read it back and verify proper storage. Step 10 will clear the memory controller syndrome registers. This section has three steps, as shown below. The user can select any combination of steps.

```
Step 10 :        Initialize memory controller(s)

Step 11 :        Initialize memory to all 1's

Step 12 :        Initialize memory to all 0's
```

POSSIBLE OUTPUT MESSAGES:

```
Section 1  :   INITIALIZE CARD

      STEP 12 - Initialize memory to ZEROS

-------------------------------------------------------------------------------

      The following PAGE RANGES are being tested              ALL ZEROS

-------------------------------------------------------------------------------

    Start - End      Start - End      Start - End      Start - End

    2598 - 2603      2608 - 2613      2618 - 2622      2625 - 2636

    2638 - 2653      2655 - 2720      2722 - 3015      3018 - 3064

    3066 - 3212      3214 - 3344      3346 - 3425      3430 - 3504

    3506 - 3522      3524 - 3527      3529 - 3555      3557 - 3562

    3564 - 3568      3570 - 3593      3596 - 3625      3627 - 3634

                            ...etc...
```

DEFAULT:

Steps 10 and 12.

Section 2 : IDENTIFY

The Identify section displays configuration and model revision information about the memory system currently being tested. This section has two steps:

Step 20	Configuration Information
Step 21	Model Revision Information (Identify)

The Memory Controller table shown below is from a system where all memory is located onboard the memory controller cards only (there are no associated memory array cards).

SAMPLE OUTPUT MESSAGE 1, when memory interleave is off:

```
Section 2  : IDENTIFY

      STEP 20 - Configuration Information
```

Memory Controller #		1	2	3	4	5
Slot Number	=	3	4	5	7	9
Hard Physical Address	=	fff8c000	fff90000	fff94000	fff9c000	fffa4000
Associated MA Cards	=	0	0	0	0	0
MC chip size (in bits)	=	1048576	1048576	1048576	1048576	1048576
MC memory size (in Mbytes)	=	8	8	8	8	8
Start Page	=	0	4096	8192	12288	16384
End Page	=	4095	8191	12287	16383	20479
Page Range	=	4096	4096	4096	4096	4096

SAMPLE OUTPUT MESSAGE 2, when memory interleave is off:

```
Section 2  : IDENTIFY

      STEP 20 - Configuration Information
```

Memory Controller #		1	2
Slot Number	=	0	1
Hard Physical Address	=	fff80000	fff81000
Associated MA cards	=	3	1

```
>>>MEMORY ARRAY cards under control of MC in Slot # 0
>>>WARNING: If a memory array is disabled, then the corresponding start page
and the corresponding end page will be -1.

If a memory controller is disabled, then all the memory arrays
under this memory controller are also disabled.
```

Memory Array #		1	2	3
Slot Number	=	0	1	2
MA chip size (in bits)	=	1048576	1048576	1048576
MA memory size (in Mbytes)	=	16	16	16
Start Page	=	0	-1	8192
End Page	=	8191	-1	16383
Page Range	=	8192	0	8192

```
>>>MEMORY ARRAY cards under control of MC in Slot # 1
>>>WARNING: If a memory array is disabled, then the corresponding start page
and the corresponding end page will be -1.

If a memory controller is disabled, then all the memory arrays
under this memory controller are also disabled.
```

Memory Array #		1
Slot Number	=	0
MA chip size (in bits)	=	1048576
MA memory size (in Mbytes)	=	16
Start Page	=	16384
End Page	=	24575
Page Range	=	8192

SAMPLE OUTPUT MESSAGE 3, when memory interleave is on:

```
Section 2  : IDENTIFY

      STEP 20 - Configuration Information
```

Memory Controller #		1	2
Slot Number	=	0	1
Hard Physical Address	=	fff80000	fff81000
Associated MA cards	=	3	2

```
>>>MEMORY ARRAY cards under control of MC in Slot # 0
>>>WARNING: If a memory array is disabled, then the corresponding start page
and the corresponding end page will be -1.
If a memory controller is disabled, then all the memory arrays
under this memory controller are also disabled.
```

Memory Array #		1	2	3
Slot Number	=	0	1	2
MA chip size (in bits)	=	1048576	1048576	1048576
MA memory size (in Mbytes)	=	16	16	16
MA status	=	Enabled	Disabled	Enabled
Memory Interleave is ON.				

```
>>>MEMORY ARRAY cards under control of MC in Slot # 1
>>>WARNING: If a memory array is disabled, then the corresponding start page
and the corresponding end page will be -1.
If a memory controller is disabled, then all the memory arrays
under this memory controller are also disabled.
```

Memory Array #		1	2
Slot Number	=	0	2
MA chip size (in bits)	=	1048576	1048576
MA memory size (in Mbytes)	=	16	16
MA status	=	Enabled	Enabled
Memory Interleave is ON.			

The Memory Controller table shown below is from a system where the memory controller is on the mother board, and memory array cards, which are installed in pairs in a slot, are also on the mother board.

SAMPLE OUTPUT MESSAGE 4, MC/MA on the mother board:

```
Section 2  :    IDENTIFY


STEP 20 - Configuration Information
```

Memory Controller # on the mother board slot			
Slot Number	=	3	1
Hard Physical Address	=	6	1
Associated Bank Number	=	0	0
Associated SPA	=	0	16777216
Associated MA cards	=	2	2
Start Page	=	0	4096
End Page	=	4095	8191
Page Range	=	4096	4096

```
>>> MEMORY ARRAY cards under control of MC in Slot # 3

>>> WARNING:  If a memory array is disabled, then the corresponding start page
              and the corresponding end page will be -1.
              If a memory controller is disabled, then all the memory arrays
              under this memory controller are also disabled.
```

Memory Array #		1	2
Slot Number	=	A	B
MA chip size (in bits)	=	4194304	4194304
MA memory size (in Mbytes)	=	8	8
MA status	=	Enabled	Enabled

```
>>> MEMORY ARRAY cards under control of MC in Slot # 1

>>> WARNING:  If a memory array is disabled, then the corresponding start page
              and the corresponding end page will be -1.
              If a memory controller is disabled, then all the memory arrays
              under this memory controller are also disabled.
```

Memory Array #		1	2
Slot Number	=	A	B
MA chip size (in bits)	=	4194304	4194304
MA memory size (in Mbytes)	=	8	8
MA status	=	Enabled	Enabled

Finally, Step 21 will print out an Identify Table for each memory controller in the memory system, this will contain model/revision information for the system, as well as the SPA associated with the memory array card.

SAMPLE OUTPUT MESSAGE 1:

```
STEP 21 - Identify Information
```

IDENTIFY Information for Memory Controller in Slot #0		
Hardware Model	=	4
Hardware Revision	=	4
Software Model	=	9
Software Revision	=	0
Software Option	=	0
Soft Physical Address	=	0

SAMPLE OUTPUT MESSAGE 2, MC/MA on the mother board:

```
STEP 21 - Identify Information
```

IDENTIFY Information for Memory Controller in Slot # on the mother board		
Hardware Model	=	0
Hardware Revision	=	1
Software Model	=	0
Software Revision	=	0
Software Option	=	0
Soft Physical Address	=	0

DEFAULT: Steps 20 and 21.

Section 3 : STATUS

This section will return status information from *all* the Memory Controller status register(s) associated with the memory system to be tested. This means the error flag conditions reported in the syndrome registers will be displayed. An example is shown below of a system with two memory controllers.

SAMPLE OUTPUT MESSAGE 1:

```
SECTION 3 - STATUS

------------------------------------------------------------
          Memory Controller #1              Slot A
------------------------------------------------------------
     >>> NO Memory Errors registered.
------------------------------------------------------------
          Memory Controller #2              Slot 8
------------------------------------------------------------
     >>> SINGLE BIT ERROR due to CHECK bits



END Section 3
```

SAMPLE OUTPUT MESSAGE 2, MC/MA on the mother board:

```
SECTION 3 - STATUS

------------------------------------------------------------
        Memory Controller #  on the mother board  slot
------------------------------------------------------------
   >>> NO Memory Errors registered.



END Section 3
```

DEFAULT:

This section has no steps.

Section 4 : MEMORY TEST (1 Second Wait per BUSY Page)

This section is a standard selection of pattern tests to exercise memory, checking for basic functionality. It is broken down into three steps, where each step is actually running two tests: the specified pattern and its complement pattern. There is also a fourth step that will batch the patterns into one test.

If no steps are requested in the **run** command, this section will default to run all six tests. The patterns in this section will also be run in the Trouble Tree section.

The difference between running all tests by selecting step 43, or running all tests by selecting the section (which defaults to run steps 40, 41, and 42) is as follows. Running the Section will sweep memory from start to end six times, once for each pattern used. Runniug Step 43 will sweep memory from start to end only once, writing and reading back all six patterns in the same cycle. This means that:

- The time that it takes to run step 43 is less than the time that it takes to run the section, because step 43 allocates a page only once, instead of six times.
- Running step 43 may not cover as many busy pages as running the section.

The diagnostic will cycle through memory requesting memory pages as it goes; if a page is busy, this section will wait 1 second for that page to free up before it proceeds on to request the next page.

The tests are explained below.

Step 40 **Address Uniqueness/Complement test:** This test will write the address value of a memory location into itself. For example, the value 12 will be stored in memory address location 12. This test will verify that every storage cell exists as a separate and unique entity. Once this is verified, the complement of this test will be performed, where the complement of each address will be written into each address location.

Step 41 **Walking Ones/Zeros pattern test:** This test writes a walking one pattern into memory, then it is read back and verified. The diagnostic will then write and verify a complementary walking zero pattern. The Walking Ones pattern test will cycle through memory writing 32 patterns (each of 32 bits) into consecutive memory locations.

The patterns cycled through are as follows:

```
1000 0000 0000 0000 0000 0000 0000 0000
0100 0000 0000 0000 0000 0000 0000 0000

0010 0000 0000 0000 0000 0000 0000 0000
0001 0000 0000 0000 0000 0000 0000 0000

             . . . etc . . .

0000 0000 0000 0000 0000 0000 0000 1000
0000 0000 0000 0000 0000 0000 0000 0100

0000 0000 0000 0000 0000 0000 0000 0010
0000 0000 0000 0000 0000 0000 0000 0001
```

Step 42 **Alternating Ones/Zeros pattern test:** This test writes an Alternating Ones pattern into memory and then reads it back and verifies memory contents. Again, the complementary pattern of Alternating Zeros is written and read back upon successful completion of the initial pattern.

The Alternating Ones pattern is as follows :

```
1010 1010 1010 1010 1010 1010 1010 1010
```

Step 43 **All Patterns Section 4:** Starting at Page 0, request pages until end of memory is reached. For each page obtained, write and read back *all* patterns from Section 4 into this page.

POSSIBLE OUTPUT MESSAGES:

```
Section 4  :   MEMORY TEST   (1 second wait per BUSY page)

        STEP 40 - Address Uniqueness/Complement Test

-------------------------------------------------------------------------------

      The following PAGE RANGES are being tested              ADDRESS UNIQUENESS

-------------------------------------------------------------------------------

      Start - End       Start - End       Start - End       Start - End

      2598 - 2603       2608 - 2613       2618 - 2622       2625 - 2636

      2638 - 2653       2655 - 2720       2722 - 3015       3018 - 3064

      3066 - 3212       3214 - 3344       3346 - 3425       3430 - 3504

      3506 - 3522       3524 - 3527       3529 - 3555       3557 - 3562

      3564 - 3568       3570 - 3593       3596 - 3625       3627 - 3634

      3636 - 3673       3570 - 3593       3596 - 3625       3627 - 3634

                                  ...etc...

   END Section 4
```

DEFAULT:

Steps 40, 41, and 42.

Section 5 : MEMORY TEST (5 Second Wait per BUSY Page)

This section is broken down into three steps, where each step is actually running two tests: the specified pattern and its complement pattern. It will defanlt to run all six tests, if no steps are requested in the run command. There is also a fourth step that will batch all the patterns into a single test. This section will *not* be called from the Tronble Tree section.

The difference between running all tests by selecting Step 53, or running all tests by selecting the section (which defaults to run steps 50, 51, and 52) is as follows. Running the Section will sweep memory from start to end six times, once for each pattern used. Running Step 53 will sweep memory from start to end only once, writing and reading back all six patterns in the same cycle. This means:

- The time that it takes to rnn step 53 is less than the time that it takes to run the section, because step 53 allocates a page only once, instead of six times.
- Running step 53 may not cover as many busy pages as running the section.

The diagnostic will cycle through memory requesting memory pages as it goes; if a page is busy, this section will wait 5 seconds for that page to free up before it proceeds on to request the next page.

Step 50 **Address Uniqueness/Complement test:** This test will write the address value of a memory location into itself. For example the value 12 will be stored in memory address location 12. This test will verify that every storage cell exists as a separate and uniqne entity. Once this is verified, the complement of this test will be performed, where the complement of each address will be written into each address location.

Step 51 **Walking Ones/Zeros pattern test:** This test writes a walking one pattern into memory, then it is read back and verified. The diagnostic will then write and verify a complementary walking zero pattern. The Walking Ones pattern test will cycle through memory writing 32 patterns (each of 32 bits) into consecntive memory locations. The patterns cycled through are as follows:

```
1000 0000 0000 0000 0000 0000 0000 0000
0100 0000 0000 0000 0000 0000 0000 0000

0010 0000 0000 0000 0000 0000 0000 0000
0001 0000 0000 0000 0000 0000 0000 0000

             . . . etc . . .

0000 0000 0000 0000 0000 0000 0000 1000
0000 0000 0000 0000 0000 0000 0000 0100

0000 0000 0000 0000 0000 0000 0000 0010
0000 0000 0000 0000 0000 0000 0000 0001
```

The diagnostic will then write and verify a complementary walking zero pattern.

Step 52 **Checkerboard pattern/Complement test:** This test will write a checkerboard pattern to the memory space. Afterwards it will read back the memory and verify that the data was stored correctly; then the complement test will be done. The pattern written is the Alternating Ones pattern followed by the complement of this pattern. After each memory address location is written, the pattern is toggled and so on.

Step 53 **All patterns Section 5:** Starting at Page 0, request pages until end of memory is reached. For each page obtained, write and read back *all* patterns in Section 5 into this page.

POSSIBLE OUTPUT MESSAGES:

```
Section 5  :   MEMORY TEST  (5 second wait per BUSY page)

       STEP 50 - Address Uniqueness/Complement Test

-------------------------------------------------------------------------------

     The following PAGE RANGES are being tested              ADDRESS UNIQUENESS

-------------------------------------------------------------------------------

     Start - End       Start - End       Start - End       Start - End

     2598 - 2603       2608 - 2613       2618 - 2622       2625 - 2636

     2638 - 2653       2655 - 2720       2722 - 3015       3018 - 3064

     3066 - 3212       3214 - 3344       3346 - 3425       3430 - 3504

     3506 - 3522       3524 - 3527       3529 - 3555       3557 - 3562

     3564 - 3568       3570 - 3593       3596 - 3625       3627 - 3634

     3636 - 3673       3570 - 3593       3596 - 3625       3627 - 3634

     3636 - 3673       3676 - 3679       3681 - 3770       3772 - 3776

                              ...etc...

  END Section 5
```

DEFAULT:

Steps 50, 51, and 52.

Section 6 : SWEEPER FUNCTION (Unimplemented)

This section runs a basic sweeping function to read all memory locations. If any errors are found, those errors will be logged in an error log for future correction.

A sweeper function will *sweep* through memory from beginning to end and read all memory locations that it can obtain access to. By making a read of a memory location the data from that location will pass through the EDC logic and if there is a SBE error within the data read it will show up in the associated memory controllers status word. The sweeper function will read a memory range and then check the status register to see if an error occurred. If an error is flagged, the sweeper function will log the location where it occurred, clear the error flag and continue the sweep. This function is important because it can help monitor the level of soft errors occurring in memory, this can help reduce the probability of HPMCs (machine crashs) by allowing the user to see possible problem areas in memory. If au area is showing repetitive occurrence of soft errors, then possibly this card should be changed. The type of error, address location, and associated memory controller will be logged in a file for monitoring of soft error levels as a preventative maintenance measure. The diagnostic will display the message shown below at the completion of the sweeping function. The errors logged will be broken down into the categories of *buffer errors* and *card errors*.

This section is currently unimplemented and will be implemented dependent upon Hardware and Operating System changes that are needed to mask any HPMC that could occur as a result of this test. The reason this may cause HPMCs is that by sweeping through memory and reading each location you will be touching areas of memory on a regular basis. If you read a memory location and there is a hard error there (a stuck bit) and a transient error occures during the read (say a glitch) you will have a double bit error which will cause an HPMC on the system. So by regular sweeps through memory you are increasing the probability of causing HPMCs.

POSSIBLE OUTPUT MESSAGES:

```
Section 6  :   SWEEPER FUNCTION TEST

Sweeper function has completed.

TOTAL ERRORS logged ==>> 5
Card   errors :  5
Buffer errors :  0
END Section 6
```

DEFAULT:

```
This section has no steps
```

Section 7 : EDC LOGIC TEST (Unimplemented)

This section runs an error logic test sequence to verify functionality of the EDC logic on board the memory card(s). This test will write false data into the Syndrome Register usually by writing a force_error register that is architected in the system. This force_error register will cause the EDC hardware to flag a data error because the Syndrome will not match up to the data read. The data finally put out on the bus will be corrected by the EDC logic; but it is corrected only *on-the-fly* and not in the actual memory location. This means that the EDC logic can detect a SBE and correct that single bit value before putting it out onto the bus; however, the value stored in memory is still incorrect.

The force_error register name varies from system to system and some may not even have this feature designed in. If that is the case, the diagnostic will print a message stating that the test cannot be performed because it is not functionally supported.

This section is currently unimplemented and will be implemented dependent upon Hardware and Operating System changes that are needed to mask any HPMC that could occur as a result of this test.

POSSIBLE OUTPUT MESSAGES:

```
Section 7  :   EDC LOGIC TEST

EDC Logic Test completed successfully.

END Section 7
```

DEFAULT:

```
This section has no steps
```

Section 8 : USER INTERACTIVE

This section will present the user with a menu presenting different areas requiring user interaction. The user will input the area they wish to proceed in. They will then be prompted for whatever parameters are needed for that area to be performed. This section will be menu driven and grow with the functionality needs of each PA-RISC system that is added to the diagnostic repertoire.

The main menu is explained below along with the prompts that will be displayed with each user selection.

POSSIBLE OUTPUT MESSAGES:

```
   Section 8  :   USER INTERACTIVE

================================================================================
                            MEMDIAG MENU
================================================================================

 1.  Memory Test  (using parameter settings)
 2.  EDC Test     (using parameter settings)
 3.  Sweep Memory (using parameter settings)
 4.  Set MEMDIAG parameters
 5.  View MEMDIAG parameters
 6.  View Configuration Information
 7.  View Page Status Lists
 8.  View Page Status Summary
 9.  View Memory Status (over Range)
10.  HELP Menu
11.  Exit to DUI

Input number >>>
```

If users select *1* from the Main Menu (*Memory Test*), the test will write the selected pattern(s) to all available memory locations, beginning at the starting page and ending at (starting page + page range - 1). This selection will obtain the pages within the page range specified by the user before the test is done, and then will return ownership to the system after the test is performed.

The memory test will run until the End of Range (User Input) or End of Memory is encountered, whichever comes first.

SAMPLE OUTPUT MESSAGE:

```
Section 8 : MEMORY TEST

-------------------------------------------------------------------------------
      The following PAGE RANGES are being tested                ALL PATTERNS
-------------------------------------------------------------------------------

   Start - End       Start - End       Start - End       Start - End

   2598 - 2603       2608 - 2613       2618 - 2622       2625 - 2636

   2638 - 2653       2655 - 2720       2722 - 3015       3018 - 3064

   3066 - 3212       3214 - 3344       3346 - 3425       3430 - 3504

   3506 - 3522       3524 - 3527       3529 - 3555       3557 - 3562

   3564 - 3568       3570 - 3593       3596 - 3625       3627 - 3634

   3636 - 3673       3570 - 3593       3596 - 3625       3627 - 3634

   3636 - 3673       3676 - 3679       3681 - 3770       3772 - 3776

                          ...etc...
```

Remember that you might not be able to obtain all the pages you requested. Many pages are permanently reserved by the operating system.

To get a general idea of the size and location of your own operating system, you can go into selection *7* (*View Page Status Lists*) and view all Reserved Pages.

If users select *2* from the Main Menu (*EDC Test Memory Range*), this function will test EDC (Error Detection and Correction) logic corresponding to the memory locations falling between the starting page and (starting page + page range - 1).

When the data is written to a memory location, it will pass through EDC logic, where a Syndrome word is encoded (7 to 9 bits long) and stored in an EDC RAM. This Syndrome word will be read whenever its associated memory location is read. Both the data word and the Syndrome word will pass back through EDC logic where the Syndrome word will be decoded.

This decoded Syndrome word and the actual data word read from the memory location will be compared with each other. If they are not equal, an error exists. The PA-RISC architecture has special registers that can be written to change the correct Syndrome word to a false one. This will test the EDC logic to see if it detects this error.

This selection will obtain the pages within the page range specified by the user before the test is done, and then returns ownership to the system after the test is performed. Remember that you might not be able to obtain all the pages you requested. Many pages are permanently reserved by the operating system.

If users select *3* from the Main Menu (*Sweep Memory Range*), the function will read all available memory locations, beginning at the starting page and ending at (starting page + page range - 1). The memory contents will scroll on screen. Every eighth memory address will display above a line containing the contents of that address, and the next seven memory locations after that address. The output will be presented in a hexadecimal format for a more compact display. A sample is shown below:

```
   PAGE 0
-------------------------------------------------------------------------------------------------------

 Starting address for this row >>     0x 0000 0000

 00000000  00000001  00000002  00000003  00000004  00000005  00000006  00000007
 00000008  00000009  0000000A  0000000B  0000000C  0000000D  0000000E  0000000F
 00000020  00000021  00000022  00000023  00000024  00000025  00000026  00000027
 00000028  00000029  0000002A  000000A3  000000A4  000000A5  000000A6  000000A7

 Starting address for this row >>     0x 0000 00A8

 000000A8  000000A9  000000AA  000000AB  000000AC  000000AD  000000AE  000000AF
 000000B0  000000B1  000000B2  000000B3  000000B4  000000B5  000000B6  000000B7
 000000B8  000000B9  000000BA  000000BB  000000BC  000000BD  000000BE  000000BF
 000000C0  000000C1  000000C2  000000C3  000000C4  000000C5  000000C6  000000C7

                              ... etc ...
```

If the user selects *4* from the Main Menu (*Set MEMDIAG Parameters*), the option to set a variety of **MEMDIAG** parameters is presented, which permits the user to tailor the diagnostic to more closely conform to the environment in which it will be running:

```
Set MEMDIAG Parameters

-------------------------------------------------------------------------------

 1. Launch Memory Test NOW
 2. View Current Parameter Settings
 3. Change PATTERN parameter
 4. Change START PAGE parameter
 5. Change PAGE RANGE parameter
 6. Change PAGE WAIT parameter
 7. Change ALLOC SIZE parameter
 8. Change LOOP COUNT parameter
 9. Change REPEAT PROMPT? parameter
10. Exit to Main Menu
11. Exit to DUI

<CR> STEP THROUGH ALL PARAMETER CHOICES

-------------------------------------------------------------------------------
```

If the user wants to change all the parameters, he should hit <CR>, which will walk him through each parameter, displaying its current setting and also the possible input to choose from. If only one (or a few) parameters are to change, select the number corresponding to that parameter(s).

The following list explains the *Set Memdiag Parameter Menu* options:

1. Launch Memory Test NOW — This selection will launch the memory test with current parameter settings.

2. View Current Parameter Settings — This selection will display the Memdiag Parameter table.

3. Change PATTERN Parameter

The user will be presented with a Pattern Menu of 13 choices, The pattern selected will be used when running *Memory Test (#1)*.

```
================================================================================
                        MEMDIAG PATTERN MENU
================================================================================

                          1. All Ones
                          2. All Zeros
                          3. Alternating Ones
                          4. Alternating Zeros
                          5. Walking One
                          6. Walking Zero
                          7. Checkerboard
                          8. Checkerboard complement
                          9. Address Uniqueness
                         10. Address Uniqueness complement
                         11. ALL straight patterns
                         12. ALL complement patterns
                         13. ALL patterns listed above
                        <CR> Keep current  setting


                      Input number of pattern desired >>
```

Pattern reading and writing is the time-intensive part of memory testing. By increasing the number of test patterns used, you increase the time it takes the test to complete. It is recommended that you test system performance with ONE pattern, before trying to batch patterns in a memory test. The following warning will print out before you are presented with the pattern menu shown above:

```
***
***     WARNING                              ( MEMWARN 10285 )
***
***     Selecting groups of patterns in the User Interactive section
***     will cause CPU intensive inner loops and system performance
***     will be reduced.
***
***     If you have not already done so, run with ONE pattern to test
***     out performance degradation before trying multiple pattern tests.
***
```

4. Change START PAGE parameter The user will be prompted for a test Starting Page. Memory page mappings will be displayed to allow the user to test a specific memory controller or memory array card.

A sample output is shown below:

```
Memory Cont/Array in Slot 7 PAGES        0 to  4095      RANGE = 4096

Memory Cont/Array in Slot 8 PAGES     4096 to  8191      RANGE = 4096

Memory Cont/Array in Slot 9 PAGES     8192 to 12287      RANGE = 4096

Currently = 0

Input STARTING PAGE or <CR>to keep current setting >>>
```

5. Change PAGE RANGE parameter The user will be prompted for a test Page Range. Memory page mappings will be displayed to allow for the user to test a specific memory controller or memory array card.

A sample output is shown below:

```
Memory Cont/Array in Slot 7 PAGES        0 to  4095      RANGE = 4096

Memory Cont/Array in Slot 8 PAGES     4096 to  8191      RANGE = 4096

Memory Cont/Array in Slot 9 PAGES     8192 to 12287      RANGE = 4096

Currently = 8192

Input PAGE RANGE or <CR>to keep current setting >>>
```

6. Change PAGE WAIT parameter This is the number of seconds that the program will wait on a BUSY page while trying to obtain access to it, before continuing on to request the next virtual memory page.

7. Change ALLOC SIZE parameter	The normal memory tests will cycle through memory requesting ONE page at a time. This is due to an MPE operating system limitation. If you are on HP-UX or in Single User Mode, you will probably want to increase this parameter and run tests from within the User Interactive Section.

Warning

The MPE XL operating system does not have clean-up routines if `MEMDIAG` were to fail, meaning if you set ALLOC SIZE = 2048 and `MEMDIAG` is granted ownership of these pages and then dies, these pages are lost to the system. This is dangerous. The HPUX operating system is able to recover these pages, therefore, this is not a problem for HPUX. It is recommended that you use ONE page for ALLOC SIZE when running `MEMDIAG` on an MPE system.

8. Change LOOP COUNT parameter	The user will input a TEST loop counter for repeat test cycles. Loop Count will be preempted by the Repeat Prompt query.
9. Change REPEAT TEST PROMPT? parameter	If ON, prompt the user at the end of the test cycle to repeat last test. If OFF, continue normal diagnostic flow of execution. The Repeat Prompt Query will take precedence over Loop Count.

All user input parameter values will ONLY affect the TESTS run from WITHIN the User Interactive Section of **`MEMDIAG`** (Section 8). Specifically, the test selections affected are selections #1 (*Memory Test*), #2 (*EDC Test*), and #3 (*Memory Sweep*). The User Parameter Table does NOT affect other menu selections.

If the user selects *5* (*View MEMDIAG Parameters*) from the Main Menu, a menu similar to the following will be displayed:

USER PARAMETERS	Selected	Valid Range	Default
PATTERN(S)	All Ones	1 to 13	ALL ONES
START PAGE	100	0 to 20479	0
PAGE RANGE	3000	1 to 20480	20480
PAGE WAIT	5	0 to 10	0
ALLOC SIZE	1024	1 to 2048	1
LOOP COUNT	2	1 to 10000	1
REPEAT PROMPT?	ON	ON/OFF	OFF

Typing a carriage return will leave parameters at the current setting.

If the user selects *6* (*View Configuration Information*) from the Main Menu, ALL configuration information about any Memory Controller and/or Memory Array cards in the system will be displayed. The following is a sample display:

Memory Controller #		1	2	3	4	5
Slot Number	=	4	8	7	6	5
Hard Physical Address	=	fff90000	fffa0000	fff9c000	fff98000	fff94000
Associated MA Cards	=	0	0	0	0	0
MC chip size (in bits)	=	1048576	1048576	1048576	1048576	1048576
MC memory size (in Mbytes)	=	8	8	8	8	8
Start Page	=	0	4096	8192	12288	16384
End Page	=	4095	8191	12287	16383	20479
Page Range	=	4096	4096	4096	4096	4096

If the user selects 7 (*View Page Listings*) from the Main Menu, a list of the pages in the system that are status pages selected by the user will be printed.

```
=========================================================================
                        PAGE STATUS MENU
=========================================================================

                          1.  View Reserved   Pages
                          2.  View Available  Pages
                          3.  View Busy       Pages
                          4.  View Hole       Pages
                          5.  View Locked     Pages
                          6.  View Bad        Pages
                          7.  View Bad/Locked Pages
                          8.  View Page Status Summary
                          9.  Exit to Main Menu
                         10.  Exit to DUI

                       Input number >>>
```

A sample Reserved Page List follows:

```
-------------------------------------------------------------------------
                        RESERVED PAGE LIST
-------------------------------------------------------------------------

0     to 3180      4000 to 4200        4210 to 4211        4224 to 5000
5001  to 5009      6001 to 6001        8010 to 8020        9111 to 9111
9200  to 9201      9990 to 9999       10010 to 10100      12000 to 12001
13433 to 13435    13500 to 13555      13599 to 13600      13701 to 13701
                            ... etc ...
-------------------------------------------------------------------------
                    Reserved pages =   4000

                    TOTAL PAGES    =   20479
-------------------------------------------------------------------------
```

Page Status Types are defined under Main Menu selection *9 (View Memory Status)*.

If the user selects *8 (View Memory Status Summary)* from the Main Menu, a synopsis of the ALL System Page Status will be printed out. A sample output follows:

```
------------------------------------------------------------------------
                        STATUS SUMMARY SNAPSHOT
------------------------------------------------------------------------
                        Reserved pages    =   3761
                        Available pages   =  16716
                        Busy pages        =      2
                        Locked pages      =      0
                        Bad pages         =      0
                        Bad/Locked pages  =      0
                        Hole pages        =      0
------------------------------------------------------------------------
                        TOTAL PAGES       =  20479
```

The Reserved pages consist primarily of operating system space. Some pages may stay constantly busy because they are used for system or program stacks. All these page assignments are dynamic. Even the Reserved pages total may vary to a small extent between different Status Summary Snapshots.

Page Status Types are defined under Main Menu selection *9 (View Memory Status)*.

If the user selects *9* (*View Memory Status Over Range*) from the Main Menu, a starting page and a page range will be prompted for. The Page and Range values input for the Memory Status selection DO NOT change those parameters in the TEST parameter table (set in Main Menu selection #4). This function will check status on all the pages requested, and will present a snapshot of the memory at the time of the request.

The status of the pages will be scrolled on the screen. The output will have a starting page number, followed by the status of that page and of the following seven pages. Viewing memory range does not require that access be obtained to the page in order to view its status; therefore, you may view any page status within the operating system valid range of pages.

The majority of page status types will be Available, Busy, and Reserved. The other four page types are fairly uncommon, and are used for special cases within an operating system.

A sample is presented below:

```
-------------------------------------------------------------------------
(ROW)
Start                    PAGE STATUS
Page
-------------------------------------------------------------------------

0902    |  Reserv  Reserv  Reserv  Reserv  Reserv  Reserv  Reserv  Reserv
0911    |  Busy    Busy    Busy    Busy    Busy    Busy    Busy    Avail
0920    |  Busy    Busy    Avail   Reserv  Reserv  Avail   Busy    Avail
0929    |  Reserv  Avail   Avail   Avail   Avail   Avail   Reserv  Avail
0938    |  Reserv  Avail   Avail   Avail   Avail   Avail   Reserv  Avail
0947    |  Reserv  Avail   Avail   Avail   Avail   Avail   Reserv  Avail
0956    |  Reserv  Avail   Avail   Avail   Avail   Avail   Reserv  Avail
0965    |  Locked  Locked  Locked  Bad_Lk  Bad_Lk  Bad_Lk  Reserv  Avail

                        ... etc ...
-------------------------------------------------------------------------
```

Reserv	This means that the page is permanently reserved by the operating system, and cannot be obtained for testing (in an on-line environment).
Avail	This means that the page is NOT LOCKED, and is possibly available for testing because it has not been allocated to any other process. However, many pages may show Available status and not be LOCKABLE due to constant system usage.
Busy	This means that the page is currently busy and, therefore, is unavailable for testing until the process currently using it has completed. This is usually associated with I/O.
Hole	Page numbers with no physical manifestation are termed "holes". For example, if the maximum page of the system is 20479, and somehow page 20500 was requested, it would not exist and the status of "hole" would be returned.
Locked	This means that the page is currently locked by a process (that process may be your own diagnostic or an external process).
Bad	This means the page has been marked "bad" by the operating system. For example, if a diagnostic found excessive errors within a page, it could mark the page "bad" and lock out other processes from accessing this corrupted page. This is dependent upon the operating system's ability to bar user access to corrupted pages.
Bad_Locked	This means that a page is both Locked and deemed Bad within the system.

If the user selects *10* (*Help Menu*) from the Main Menu, the following menu is displayed:

```
==============================================================================
                          MEMDIAG HELP MENU
==============================================================================

                      1.   Memory Test  (using parameter settings)
                      2.   EDC Test     (using parameter settings)
                      3.   Sweep Memory (using parameter settings)
                      4.   Set MEMDIAG parameters
                      5.   View MEMDIAG parameters
                      6.   View Configuration Information
                      7.   View Page Status Lists
                      8.   View Page Status Summary
                      9.   View Memory Status (over Range)
                     10.   Exit to Main Menu
                     11.   Exit to DUI

                     Input the area you want >>>
```

Selecting any of these items will scroll the explanations presented in this manual for each of the listed areas.

If the user selects *11 (Exit to DUI)* from the Main Menu, control is returned to the DUI command line.

Section 9 : TROUBLE TREE (Default Test)

This section can be used to perform all possible diagnostics within the `MEMDIAG` repertoire. This is the **default** section for the Memory Array Diagnostic. If you just type `run memdiag`, then this section will be executed. The testing sequence for this section is as follows:

1. Initialize memory controller status to clear status errors.
2. Perform memory pattern test with
   a. All Zeros
   b. All Ones
3. Print configuration information
4. Perform memory pattern test with
   a. Address Uniqueness
   b. Address Uniqueness Complement
   c. Walking Ones
   d. Walking Zeros
   e. Alternating Ones
   f. Alternating Zeros

POSSIBLE OUTPUT MESSAGES:

```
-------------------------------------------------------------------------------

                              TROUBLE TREE

-------------------------------------------------------------------------------

   >>>  SECTION 9

   >>>  ERROR REGISTERS CLEARED SUCCESSFULLY

   >>>  ONES PATTERNS TEST COMPLETED SUCCESSFULLY

           (tested page list displayed here)

   >>>  ZEROS PATTERNS TEST COMPLETED SUCCESSFULLY

           (tested page list displayed here)

   >>>  MEMORY CONTROLLERS/MEMORY ARRAYS INITIALIZED SUCCESSFULLY

   >>>  SYSTEM CONFIGURATION TABLE

           (tables displayed here)

   >>>  ADDRESS UNIQUENESS PATTERN TEST COMPLETED SUCCESSFULLY

           (tested page list displayed here)

   >>>  WALKING ONES/ZEROS PATTERN TEST COMPLETED SUCCESSFULLY

           (tested page list displayed here)

   >>>  ALTERNATING ONES/ZEROS PATTERN TEST COMPLETED SUCCESSFULLY

           (tested page list displayed here)


   >>>
   >>>
   >>>  MEMDIAG HAS COMPLETED SUCCESSFULLY WITHOUT FINDING ANY MEMORY ERRORS.
   >>>
   >>>


   >>>  END OF SECTION 9
```

DEFAULT:

This section has no steps

Error and Warning Messages

Error messages may be displayed as a result of Syndrome Register reads, or as a result of memory errors occurring during the operation of **MEMDIAG**; memory errors may also generate warning messages.

Syndrome Register Read

One Syndrome Register is associated with each Memory Controller in a memory system. The Syndrome register will contain the status of the last memory read or write operation to memory under the supervision of the associated Memory Controller. The memory operation will result in a SUCCESSFUL completion (Message 10250) or in a FAILURE to complete (Messages 10251 thru 10258).

The possible status values are listed below. These messages are actual decoding of the hardware status returned after the memory transaction.

Cause: Memory hardware failed to complete memory transaction.

Action: Replace chip or entire card indicated in the message immediately following this one that will pinpoint the exact Field Replaceable Unit (FRU) that is malfunctioning.

ERROR MESSAGE 10250	`NO Memory Errors registered.`
ERROR MESSAGE 10251	`SINGLE BIT ERROR due to UNKNOWN cause.`
ERROR MESSAGE 10252	`SINGLE BIT ERROR due to DATA bits.`
ERROR MESSAGE 10253	`SINGLE BIT ERROR due to CHECK bits.`
ERROR MESSAGE 10254	`SINGLE BIT ERROR logged on PARITY read.`

ERROR MESSAGE 10255	SINGLE BIT ERROR logged on PARITY write.
ERROR MESSAGE 10256	MULTI BIT ERROR due to UNKNOWN cause.
ERROR MESSAGE 10257	FATAL ERROR registered.
ERROR MESSAGE 10258	UNKNOWN ERROR registered.

Error/Warning Messages

The following are possible Error/Warning messages that may be displayed during the operation of **MEMDIAG**.

MEMORY ERRORs map to one of two different error types: Card Error or Buffer Error. The most important difference is that Card Errors are physical errors and an FRU will be pinpointed; however, Buffer Errors are virtual errors and no FRU can be determined.

```
ERROR MESSAGE 10259

***
***  Unable to retrieve message # ! from MEMDIAG catalog
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE The message number indicated was not found in the message catalog of the diagnostic.
ACTION Get the version number of the diagnostic, indicate what you were attempting to do, and file an SR.

```
ERROR MESSAGE 10260

***
***  Memdiag is unable to OPEN memory, BAD exit status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***  (3) Check to see if /dev/dmem is there, or if /dev/dmem is included in
***      the kernel for an HPUX system.
***
```

CAUSE An operation requested by Memdiag to open memory was not successfully completed due to hardware or software below Memdiag. Usually this means the hardware system is not on (recognized) by the supported list Memdiag officially tests.
ACTION Get the version number of the diagnostic, indicate what you were attempting to do, and file an SR.

WARNING MESSAGE 10261

```
***
***        W A R N I N G :
***
***     It is possible that MEMDIAG will force a parity error to
***     occur.  This may cause the machine to freeze or generate
***     a High Priority Machine Check.
***
```

CAUSE — This warning will print out when the memory system has parity data checking instead of EDC error checking. Performing memory tests will exercise memory that may not normally be accessed. This will increase the probability of coming across a memory failure. However, with parity memory checking the error will be detected but not corrected and could, therefore, cause an HPMC.

ACTION — This warning is basically for old systems that are running with parity cards (ie: Burgundy), which are not "officially" supported but the warning is printed to inform user of possible results.

WARNING MESSAGE 10262

```
***
***   Bad RANGE input, this range goes over End Of Memory.
***
***        Please enter a SMALLER number.
***
```

CAUSE — User has input a Page Range that when added to the current requested Starting Page will put requested pages over valid page numbers in the memory system.

ACTION — Input a smaller Page Range value.

```
ERROR MESSAGE 10263

***
***  At the current time, the Operating System was not able to
***  ALLOC the minimum buffer size needed to run MEMDIAG.
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE The minimum buffer required to execute Memdiag was not obtained, the memory space is not available in the present state of the environment. The system is saying it ran out of memory pages to give away.

ACTION Exit the diagnostic and restart again. this will deallocate all memory pages obtained by the diagnostic and try again. If it still does not work collect data about the problem and submit to STD Online Support.

```
ERROR MESSAGE 10264

***
***  At the current time, the Sherlock Interface was not able
***  to obtain a pattern buffer from the Operating System of
***  the minimum buffer size needed to run MEMDIAG.
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE The minimum buffer required to execute Memdiag was not obtained, the memory space is not available in the present state of the environment. The system is saying it ran out of memory pages to grant ownership to.

ACTION Exit the diagnostic and restart again, this will deallocate all memory pages obtained by the diagnostic and try again. If it still does not work collect data about the problem and submit to STD Online Support.

```
WARNING MESSAGE 10265

***
***   At the current time, the Operating System was not
***   able to allocate the buffer size requested so try
***   requesting a smaller ALLOC SIZE parameter.
***
```

CAUSE At the current time, the operating system does not seem to have the amount of pages necessary to support running Memdiag with the ALLOC SIZE you requested.
ACTION Input a smaller ALLOC SIZE parameter.

```
WARNING MESSAGE 10266

***
*** WARNING -- You have reached your maximum error total
***
```

CAUSE The user specified error limit has been reached.
ACTION If you would like to allow for more errors before termination, re-run the diagnostic assigning a larger value to the errcount parameter of the run command.

WARNING MESSAGE 10267

```
***
*** WARNING -- Invalid response.  Please try again.
***
```

CAUSE Valid ranges acceptable for this prompt will be displayed either above the prompt line or at the end of the prompt line.

ACTION Input value from within displayed range.

ERROR MESSAGE 10268

```
***
***  Memdiag has encountered a BUFFER error while testing.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE This data error maps to Virtual Memory which is outside the scope of the Memory Array Diagnostic. It can indicate a failure anywhere within the HPPA system (ie bus, hard disk, cables, etc).

ACTION Rerun the test to see if the Buffer Error will repeat itself but this time map to physical memory where we can pinpoint an FRU. If the error does not repeat, chances are you will never know what caused it. If it does repeat, and repeats again as a Buffer Error, run other diagnostics to try and locate the failure within the system.

```
ERROR MESSAGE 10269

***
***  The Driver indicates that there are NO Memory Controllers in the
***  system.  Check that Memory Controller card is properly installed.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE When the driver has gone out to poll the bus line for Memory Controller cards, it has not found one present. This is obviously an impossibility because a MC card must be present within the system for it to function at all; however, the reporting of the MC card can fail in transit to the Memdiag query. This indicates some sort of system failure outside the scope of Memdiag.

ACTION Check for MC card and make sure it is completed pushed into slot, if it still does not work collect data about the problem and submit to STD Online Support.

```
ERROR MESSAGE 10270

***
***  A CARD error was detected and logged in the Syndrome register.
***     The following messages will give details of error location.
***
```

CAUSE A physical memory has occurred during a read or write operation to memory.

ACTION Replace Memory Controller or Memory Array card indentified in the subsequent error messages that will pinpoint the FRU.

ERROR MESSAGE 10271

```
***
***  A BUFFER error was detected during a comparision of Expected
***                      data to Actual Data read back from memory.
***
***     This type of error indicates some system malfunction OTHER than the
***     Memory Array cards or memory on board the Memory Controller card.
***     If these were faulty, the error would have shown during the readback
***     of the Syndrome Error Register.
***
***     The MC and/or MA card slots that map to the ERROR PAGE are given
***     only as an environment "dump" of all configuration information
***     associated with the ERROR PAGE at the time of the BUFFER error.
***
***     It is recommended that you retest the ERROR PAGE or the Page Range
***     of the associated Memory Controller and/or Memory Array where the
***     BUFFER error occurred (go into User Interactive, Section 8).
***
***     The following messages will give details on the BUFFER ERROR.
***
```

CAUSE This data error maps to Virtual Memory which is outside the scope of the Memory Array Diagnostic. It can indicate a failure anywhere within the HPPA system (ie bus, hard disk, cables, etc).

ACTION Rerun the test to see if the Buffer Error will repeat itself but this time map to physical memory where we can pinpoint an FRU. If the error does not repeat, chances are you will never know what caused it. If it does repeat, and repeats again as a Buffer Error, run other diagnostics to try and locate the failure within the system.

ERROR MESSAGE 10272

```
***
***   ERROR found at Page 1840
***
***     Expected Data = 0x        0
***
***     Actual   Data = 0x        1
***
***     Test Pattern  = All Zeros
***
```

CAUSE This data error maps to Virtual Memory which is outside the scope of the Memory Array Diagnostic. It can indicate a failure anywhere within the HPPA system (ie bus, hard disk, cables, etc).

ACTION Rerun the test to see if the Buffer Error will repeat itself but this time map to physical memory where we can pinpoint an FRU. If the error does not repeat, chances are you will never know what caused it. If it does repeat, and repeats again as a Buffer Error, run other diagnostics to try and locate the failure within the system.

ERROR MESSAGE 10273

```
***
***   MEMORY ERROR DETECTED
***
***     Test Pattern                  =   All Zeros
***
***     Bit  Location                 =   63
***     Chip Location                 =   u0301
***     Bank Location                 =   0
***
***     Memory Controller Slot        =   0
***     Memory Array Slot             =   0
***
```

CAUSE A physical memory has occurred during a read or write operation to memory.

ACTION Replace Memory Controller or Memory Array card indentified in the subsequent error messages that will pinpoint the FRU.

ERROR MESSAGE 10275

*** Under Memory Controller in Slot 3 maps to the BUFFER ERROR PAGE.

CAUSE This data error maps to Virtual Memory which is outside the scope of the Memory Array Diagnostic. It can indicate a failure anywhere within the HPPA system (ie bus, hard disk, cables, etc).

ACTION Rerun the test to see if the Buffer Error will repeat itself but this time map to physical memory where we can pinpoint an FRU. If the error does not repeat, chances are you will never know what caused it. If it does repeat, and repeats again as a Buffer Error, run other diagnostics to try and locate the failure within the system.

ERROR MESSAGE 10276

*** Memory Array in Slot A maps to the BUFFER ERROR PAGE.

CAUSE This data error maps to Virtual Memory which is outside the scope of the Memory Array Diagnostic. It can indicate a failure anywhere within the HPPA system (ie bus, hard disk, cables, etc).

ACTION Rerun the test to see if the Buffer Error will repeat itself. If the error does not repeat, chances are you will never know what caused it. If it does repeat, and repeats again as a Buffer Error, run other diagnostics to try and locate the failure within the system. If it repeats as a Card Error then the FRU will be identified.

```
ERROR MESSAGE 10277

***
***   FAILURE in INITIALIZATION of Memory Controller
***
***   Replace Memory Controller chip or entire card.
***
```

CAUSE A physical memory has occurred dnring a read or write operation to memory.
ACTION Replace Memory Controller or Memory Array card indentified in the subsequent error messages that will pinpoint the FRU.

```
ERROR MESSAGE 10278

***
***  FAILURE in retrieving Configuration Information
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE The configuration tables did not print out successfully. This indicates some sort of print utility failure, it will not effect successfnl completion of diagnostic.
ACTION Get the version number of the diagnostic, indicate what you were attempting to do, and file an SR.

ERROR MESSAGE 10279

```
***
***   FAILURE in Initialization of Memory
***
```

CAUSE A physical memory has occurred during a read or write operation to memory.
ACTION Replace Memory Controller or Memory Array card indentified in the subsequent error messages that will pinpoint the FRU.

ERROR MESSAGE 10280

```
***
***   FAILURE in Pattern Testing of Memory
***
```

CAUSE A physical memory has occurred during a read or write operation to memory.
ACTION Replace Memory Controller or Memory Array card identified in the subsequcut error messages that will pinpoint the FRU.

```
ERROR MESSAGE 10281

***
***     The TROUBLE TREE has NOT completed successfully.
***
```

CAUSE	This indicates an error occurred somewhere in the testing sequence of the Trouble Tree section. This message is the global summation of all test sequences called for Memdiag, in case the User has run overnight and returned to the console to see the last message printed out before Memdiag terminated.
ACTION	Look back through previous messages to find Specific Error Message that will pinpoint a Field Replaceable Unit.

```
ERROR MESSAGE 10282

***
***  Unable to successfully acquire DAR buffer.
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE	An operation requested by Memdiag was not successfully completed due to hardware or software below Memdiag.
ACTION	Get the version number of the diagnostic, indicate what you were attempting to do, and file an SR.

ERROR MESSAGE 10283

```
***
***  Unable to initialize Memdiag in diagostic environment.
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE An operation requested by Memdiag was not successfully completed due to hardware or software below Memdiag.

ACTION Get the version number of the diagnostic, indicate what you were attempting to do, and file an SR.

ERROR MESSAGE 10284

```
***
***  Loading of a Section or Step failed.
***  MEMDIAG will now terminate. Error Status = !
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE An operation requested by Memdiag was not successfully completed due to hardware or software below Memdiag.

ACTION Get the version number of the diagnostic, indicate what you were attempting to do, and file an SR.

WARNING MESSAGE 10285

```
***
***    W A R N I N G
***
***    Selecting GROUPS of patterns in the User Interactive section
***    will effect system performance and increase test duration.
***
***    If you have not already done so, run with ONE pattern to test
***    out your systems  performance before trying multiple pattern tests.
***
```

CAUSE Memory tests are CPU intensive tests that may effect system performance adversely.

ACTION It is suggested you run Memdiag using ONE pattern first to test out system performance under your own unique circumstances of system load, system operating system and specific hardware.

ERROR MESSAGE 10286

```
***
***  Interrupt POLL failed with Status = !
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE An operation requested by Memdiag was not successfully completed due to hardware or software below Memdiag.

ACTION Get the version number of the diagnostic, indicate what you were attempting to do, and file an SR.

WARNING MESSAGE 10287

```
***
***          W A R N I N G
***
***   The section you have selected will run memory tests that
***   take over one half hour to complete (on average).
***
***   If you don't have the time now, go into User Interactive (Sect 8)
***   and setup your own memory test by selecting parameters that will
***   specifically test certain ranges and, therefore, shorten test duration.
***
***   Do you want to continue with this Section now? (y/n) [y]:
***
```

CAUSE This is to give the CE a way out if he is not aware of the time required to run these memory tests.

ACTION If you can run the diagnostic in the background, do so. If you are looking for immediate results, go into the User Interactive (Section 8) and run specific test ranges.

ERROR MESSAGE 10289

```
***
***  Memdiag is unable to read Syndrome Register
***  in order to display status,  Bad Exit Status = !
***  MEMDIAG will continue ...
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE Memdiag requested lower level software to read the Syndrome Register in order to decode the error flags set, if any. The request was not successfully completed. This is only for display so, therefore, Memdiag can continue.

ACTION Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10290

***
***  Memdiag is unable to read Model Information
***  in order to display Revision Info,  Bad Exit Status = !
***  MEMDIAG will continue ...
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE Memdiag requested lower level software to read the Model Information in order to display revision information. The request was not successfully completed. This is only for display so, therefore, Memdiag can continue.

ACTION Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10291

***
***  Memdiag is unable to Allocate pages.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE Memdiag was not able to successfully allocate a page from the operating system. This means it could not be granted ownership of it in order to test it.

ACTION Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10292

***
***  Memdiag is unable to Deallocate pages.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE	Memdiag was not able to successfully deallocate a page to the operating system. This means it could not be return ownership of it.
ACTION	Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10293

***
***  Memdiag is unable to User Allocate pages.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE	Memdiag was not able to successfully allocate a page from the operating system. This means it could not be granted ownership of it in order to test it.
ACTION	Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10294

***
***  Memdiag is unable to Clear Syndrome Register.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE Memdiag requested lower level software to clear the Syndrome Register. Memdiag must terminate because if the syndrome can't be cleared it will read an error at each cycle of the testing.

ACTION Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10295

***
***  Memdiag is unable to read System DETAIL Info.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE Memdiag requested lower level software to read the system detail information in order to test memory. Memdiag cannot test without knowing what memory controller/array cards are present.

ACTION Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10296

***
***  Memdiag is unable to read System CONFIGURATION Info.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE	Memdiag requested lower level software to read the system configuration information in order to test memory. Memdiag cannot test without knowing what memory controller/array cards are present.
ACTION	Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10297

***
***  Memdiag is unable to read System MODEL Info.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE	Memdiag requested lower level software to read the Model Information in order to display revision information. The request was not successfully completed. This is needed for Memdiag to order the SPAs of each memory range, therefore, Memdiag cannot continue without it.
ACTION	Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10298

***
***  Memdiag is unable to read Syndrome Register.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE Memdiag requested lower level software to read the Syndrome Register in order to decode the error flags set, if any. The request was not successfully completed. This is for error decoding after the test and is required.

ACTION Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10299

***
***  Memdiag is unable to READ Memory.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE Memdiag requested lower level software to read the memory after a pattern test write.

ACTION Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10300

***
***  Memdiag is unable to WRITE Memory.  Bad Exit Status = !
***  MEMDIAG will now terminate.
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE Memdiag requested lower level software to write the memory for pattern testing.
ACTION Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10301

***
***  Memdiag is unable to read Page Status for page !
***  Bad Exit Status = !,  MEMDIAG will continue ...
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE Memdiag requested lower level software to read the type of page status associated with the indicated page.
ACTION Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10304

***
***     Memdiag is unable to log memory error to MEMLOGP.
***     Bad Exit Status = !,  MEMDIAG will continue ...
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE	Memdiag sent an IPC message to log a memory error to MEMLOG file, but MEMLOGP was not running.
ACTION	Type in DIAGSYSTEM RUN MEMLOGP at DUI prompt. Try running the section again. If it still fails, record exit status and submit to STD Online Support.

```
ERROR MESSAGE 10306

***  Bank Number 0 maps to the BUFFER ERROR PAGE.
```

CAUSE	This data error maps to Virtual Memory which is outside the scope of the Memory Array Diagnostic. It can indicate a failure anywhere within the HPPA system (ie bus, hard disk, cables, etc).
ACTION	Rerun the test to see if the Buffer Error will repeat itself. If the error does not repeat, chances are you will never know what caused it. If it does repeat, and repeats again as a Buffer Error, run other diagnostics to try and locate the failure within the system. If it repeats as a Card Error then the FRU will be identified.

```
ERROR MESSAGE 10309

***
***  Memdiag is unable to get Page Size from MEMDAR
***  Bad Exit Status = !,  MEMDIAG will not continue ...
***
***  Action:
***  (1) Try to run the section again with ERRPRINT command modifier. or
***  (2) Try to run the section again with TRACE=ALL command modifier.
***
```

CAUSE	Memdiag requested lower level software to obtain page size for page allocation.
ACTION	Try running the section again. If it still fails, record exit status and submit to STD Online Support.

Contents

4

LAN Diagnostic

Introduction

The LAN Diagnostic (Local Area Network Device Adapter Diagnostic, LANDAD) tests the local area network interface controller (LANIC), part number 27125-60201; the HP-PB LANIC, part number 28652-60001; or the VSC (Viper System Connect) LANIC, part number A1094-66530. The diagnostic will run on any HP Precision Architecture RISC computer system which supports the Online Diagnostic subsystem. LANDAD is capable of detecting a failure in one or more field replaceable units (FRUs). For LANDAD, an FRU is considered to be the LAN interface controller (LANIC) card (which may contain non-LAN circuitry as well), the LANIC connector cable, the attachment unit interface (AUI) cable, the transceiver, and the coaxial tap or BNC tee. LANDAD will accomplish the following:

- Identify the product type and node address of the LANIC
- Report the status of the LANIC
- Report the link statistics of the LANIC
- Reset the LANIC
- Perform selftest on the LANIC
- Execute a local or external loopback
- Send TEST or XID (exchange identification) packets to a remote node and interpret the results
- Perform AUI cable and transceiver fault tests

Defects and Enhancements

Submit defect reports and enhancement requests for this diagnostic through the STARS database referencing Product Number 30600-10013.

Minimum Configuration

The minimum required hardware and software consists of the following:

- A functional HP Precision Architecture RISC computer system
- LANIC (part number 27125-60201), HP-PB LAN (part number 28652-60001), or VSC LANIC (part number A1094-66530)
- Connector/stub cable (part number 27125-63009), which is not applicable for HP-PB LAN
- HP 92254A 6-Meter AUI Cable (only used with the HP 30241A transceiver)
- HP 30241A Transceiver ("known good transceiver") for use on ThickLAN cable
- HP 92257B Transceiver Test Fixture for use with the HP 30241A Transceiver,
- HP 24861A Thintransceiver ("known good transceiver") for use on ThinLAN cable
- HP 92227Q Thintransceiver Test Fixture for use with the HP 24861A Thintransceiver
- HP 28664A Transceiver for use on Starlan 10
- HP 5061-4977 Starlan 10 Transceiver Test Fixture for use with HP 28664A Starlan 10 Transceiver
- HP 28685A Ethertwist Transceiver
- HP 5061-4977 Test Fixture for use with the HP 28685A Transceiver
- HP 28683A Fiber Optic Transceiver
- TBD Test Fixture for HP 28683A Transceiver
- MPE XL or HP-UX operating system
- Online diagnostics subsystem (includes LANDAD)

Operating Instructions

Before attempting to run the diagnostic, ensure that all LAN components are installed, connected, and that all LAN links have been configured.

Note

As of MPE XL 1.2, `LANDAD` can no longer verify the LAN card hardware, before the software is installed and configured. Specifically, the network interface must be configured, and the `NETCONTROL START;NET=LAN` command must be issued, before `LANDAD` can verify the hardware.

As of MPE XL 2.0, this will no longer be a concern.

Default Tests

If you do not specify sections and steps to be run, the following default sections and steps will be executed:

Section 3	Identify
Section 4	Local Loopback
Section 6	Status

RUN Command

To bring up the Online Diagnostic subsystem, enter the following command to the system prompt:

```
sysdiag
```

The system responds with the following prompt indicating that access has been gained to the Online Diagnostic User Interface (DUI).

```
DUI>
```

Typing HELP causes a summary of the DUI and its commands to be printed. Refer to the DUI chapter of this manual for details.

To run the diagnostic, enter:

```
DUI > RUN LANDAD pdev=4.4 <RUN Command Options>
                          |           |
                          |     none required for
                          |     default test suite
                          |
                  insert physical location of
                  device adapter to be tested here,
                  or for MPE XL, type the ldev number
                  or for HP-UX, type the ldev name
```

Test Execution

If the LANIC is already in use by the diagnostic system, the system will not grant access to it and the following message will be returned:

```
*** ERROR -- LANIC ALREADY IN USE BY DIAGNOSTIC SYSTEM (LANDADERR 5000)
*** Someone is already diagnosing the LANIC that you requested.
*** It is illegal to have two copies of LANDAD diagnosing the same LANIC
*** at the same time.
```

The diagnostic terminates after outputting this error message. Control will return to the Online Diagnostic subsystem upon completion of the requested/default sections and steps.

If the user specifies a destructive section among those requested, the security level of the user is checked to verify that it is adequate to perform the test (Level 0 or Level 1). If the security level is not adequate, the following error message is printed and the user is returned to the DUI prompt.

```
*** INSUFFICIENT SECURITY LEVEL.  (LANDADERR 5043)
```

If the user specified destructive sections, the following message is printed and the user is prompted for a yes or no answer.

```
A destructive section has been selected.
Do you wish to continue (Y/N) [ N ]?
```

The user must type "y" or "yes" to continue. To terminate the diagnostic, the user must type "n" or "no" or a <cr>. Any other user response will cause the following message to be echoed to the screen:

```
Please answer yes or no
```

The user will then be reprompted.

On MPE XL systems, when the diagnostic is invoked, it checks if the user is at a terminal that is connected to the computer via a Distributed Terminal Controller (DTC) port that uses the LANIC that is to be diagnosed. If the terminal is not connected via such a port, the diagnostic proceeds normally. If the user is at a port of this type, the diagnostic checks to see if any destructive sections are specified. If not, the diagnostic proceeds normally. If there are, the following message is displayed:

```
*** WARNING -- DESTRUCTIVE SECTIONS CAN NOT BE RUN FROM YOUR TERMINAL.
*** DESTRUCTIVE SECTIONS MAY ONLY BE RUN FROM A TERMINAL THAT IS NOT
*** CONNECTED THROUGH THE LANIC TO BE DIAGNOSED.  THE FOLLOWING
*** SECTIONS CAN BE PERFORMED FROM YOUR TERMINAL: 1,3,4,6,7,9,10.
*** NO OTHER SECTIONS CAN BE SPECIFIED. (LANDADWARN 6000)
```

If at any time the number of errors generated reaches the limit specified by you in the ERRCOUNT parameter, the following message will be output:

```
*** THE MAXIMUM NUMBER OF ERROR MESSAGES HAS BEEN EXCEEDED.  (LANDADERR 99)
```

The diagnostic will then terminate. If the ERRPAUSE parameter of the RUN command was assigned to "on", the diagnostic will stop after each error is generated and ask if the test should be continued:

```
Do you wish to continue (Y/N) [Y]?
```

If the response is "Y", the test will resume (if possible); if the response is "N", the diagnostic will terminate. If the sections and steps specified by you were executed the number of times specified in the LOOP parameter of the RUN command without the number of errors exceeding the ERRCOUNT value, the diagnostic will terminate normally.

At any time that the diagnostic is prompting for information, you may enter "exit" to terminate the diagnostic. Either the entire word or only the first letter of the word 'exit' need be entered, in either upper or lower case. If you exit in this manner, the following message is displayed:

```
Exiting LANDAD per user request...
```

Any time that the diagnostic is not prompting for information, you may enter an interrupt character (CTRL Y for MPE XL and usually CTRL C for HP-UX). When the diagnostic detects the interrupt, one of two actions will occur. If the diagnostic is at a point where it can suspend, it will print the following message and return control to the Online Diagnostic subsystem.

```
LANDAD suspended per user request...
```

You may then either resume or abort the diagnostic. If the diagnostic cannot suspend at this point, the following message will be printed:

```
Unable to suspend in current state, Aborting LANDAD...
```

At this point, LANDAD will be aborted and control will be returned to the Online Diagnostic subsystem.

Caution

On MPE XL, you should never abort LANDAD when sections 3, 4, 9, or 10 are specified. This can cause the diagnostic to lose functionality the next time the diagnostic is run.

Upon termination of the diagnostic, control will return to the Online Diagnostic subsystem.

Test Execution

The following example illustrates running sections 1 through 12 of the diagnostic on an HP-UX system, using the CIO LANIC number 27125.

All user input appears to the right of any system, subsystem, or diagnostic program prompt. Section 1—More Help pauses after printing the first paragraph. If you need information about other sections, enter the appropriate section number. If you want to continue to section 2, merely press (Return) (see the example below).

```
%sysdiag
```

```
*****************************************************************
******                                                     ******
******              ONLINE DIAGNOSTIC SUBSYSTEM            ******
******                                                     ******
******            (C) Hewlett Packard Co. 1987             ******
******                 All Rights Reserved.                ******
******                   DUI version 1.0                   ******
******                                                     ******
*****************************************************************
```

```
DUI 1>run landad pdev=8.4 sect=1/12
```

```
*****************************************************************
******                                                     ******
******       LANDAD LAN Device Adapter Diagnostic          ******
******                                                     ******
******            (C) Hewlett Packard Co. 1986,1987,1988   ******
******                All Rights Reserved.                 ******
******                  Version A.01.00                    ******
******                                                     ******
*****************************************************************

Welcome, Today is Wed Nov 04 11:01:19 1987

A destructive section has been selected.
Do you wish to continue (Y/N) [ N ]? y

Section 1 -- More Help

   This Section is no longer supported.  For additional information
   on a given section, use the sections option of the help command
   from the DUI prompt.  For example, if you want more information
   about section 9, type the following:

   DUI> help landad sections 9

End of Section 1 -- More Help

Section 2 -- Reset

End of Section 2 -- Reset
```

```
Section 3 -- Identify

   ID byte  = $46 (CIO LANIC).
   Hardware revcode  = 2.
   CIO firmware datecode = 2716.
   CIO firmware ID   = 1.
   NOVRAM (permanent) station address     = $08-00-09-00-AE-5B.
   RAM (currently active) station address = $08-00-09-00-AE-5B.

End of Section 3 -- Identify

Section 4 -- Local Loopback

    Logging SSAP with driver...
    Sending data to LAN Interface Controller...
    Receiving data from LAN Interface Controller...
    A frame has been successfully transmitted onto the network media.
    Path to LAN Interface Controller is functional.

End of Section 4 -- Local Loopback

Section 5 -- Selftest

    Selftest Completed Successfully.
    The LAN Interface Controller is functional.

End of Section 5 -- Selftest

Section 6 -- Status

    LANIC status has been read successfully.
    LANIC passed selftest.
    LANIC is online.
    transceiver power fuse is OK.
    Free transmit buffers =     4; Maximum =     4.
    Full receive buffers  =     0; Maximum =    16.
    Read data ARQ frame threshold =     1.
    Read data ARQ timeout limit   =     1.
```

```
                            RAM value           NOVRAM value
    ==========================(Current)==========(Default)=========
    Station address           $08-00-09-00-AE-5B  $08-00-09-00-AE-5B
    Receive bad frames        Disabled            Disabled
    Receive multicast frames  Enabled             Disabled
    Receive broadcast frames  Enabled             Disabled
    Receive all frames        Disabled            Disabled
    ================================================================
    The following multicast addresses are recognized:
    $09-00-09-00-00-01

End of Section 6 -- Status

Section 7 -- Link Statistics

   Step 71 - Read and Display Link Statistics
      Link level statistics have been read successfully.

      Transmit Statistics====================================
      TOTAL frames transmitted without error.................0
        Deferred transmits...................................0
        One collision transmits..............................0
        More than one collision transmits....................0
      TOTAL frames NOT transmitted...........................0
        Retry errors.........................................0
        Late collisions......................................0
        Loss of carrier during transmit......................0
      No heartbeat detected after transmission...............0
      No free transmit buffers...............................0
      TDR of last retry error................................0
      Infinite deferral errors...............................0
    Receive Statistics=======================================
      TOTAL frames received without error....................0
      Frames rejected by address filter......................0
      Frames rejected due to CRC errors......................0
      Frames rejected due to alignment errors................0
      Frames rejected due to oversize length.................0
      LAN controller indicated one or more frames lost.......0
      No free receive buffers................................0
   End of Step 71 - Read and Display Link Statistics

End of Section 7 -- Link Statistics

Section 8 -- External Loopback

   A link frame has been successfully transmitted
   and received from the network cable.

End of Section 8 -- External Loopback
```

```
Section 9 -- Remote Node Test

   This section sends a TEST frame and waits for a response from
   a specified remote node for a specified number of iterations.

   The following success/failure indicators are used:
   "." = The test frame bounced successfully.
   "#" = The test frame was not received before the timeout period.

   Remote Node Address (Six HEX bytes) => 08000900ae5b
   Number of test frames to send ("0" for infinite) [10] =>
   Length of test frames in bytes (60..1514) [500] =>

   Press the interrupt character (usually <Control-C>)
   to prematurely stop the test.

   ..........

   10 out of 10 TEST frames echoed successfully (100%).

End of Section 9 -- Remote Node Test

Section 10 -- Remote XID Test

   This section sends an IEEE 802.2 XID frame to a user specified
   remote node and waits for an IEEE 802.2 XID response frame from
   that remote node.

   Remote Node Address (Six HEX bytes) => 08000900ae5b
   Remote DSAP Address (one even hex byte between $00 and $FE) [$00] =>
   Sending XID command frame...
   Received XID response frame...

   Remote DSAP $00 has class I service.

End of Section 10 -- Remote XID Test
```

```
Section 11 -- AUI Cable Fault Isolation

   This section sends a number of external loopback frames out to a
   transceiver connected to a terminated loopback fixture
   to form a loopback hood.  The status from each frame is displayed on the
   screen.  By moving the transceiver and loopback fixture to various
   AUI cable connector junctions, faults in the AUI cable can be located.

   The following activity indicators are used:
   "P" = External loopback was successful.
   "L" = Loss of carrier detected.
         This usually indicates that there is a loose AUI connector or a broken
         AUI cable between the LANIC and the loopback connector.  In
         a Starlan 10 (Twisted Pair Transceiver) or Fiber-Optic installation,
         it will occur if loopback is attempted without a loopback fixture
         attached while the loopback switch on the transceiver is in the "Test"
         position.  In an EtherTwist installation, it may indicate that the
         setting of the "linkbeat" switch on the local transceiver is
         incompatible with the remote transceiver.
   "R" = Retry fault detected.  (Check loopback transceiver and
         loopback fixture.)
         In a Thicklan or Thinlan installation, this usually indicates that
         there is a loose or broken terminator on the loopback hood, or a
         transceiver or transceiver tap/tee problem.  In a Starlan 10 (Twisted
         Pair Transceiver) or Fiber-Optic installation, it will occur if
         loopback is attempted with a loopback fixture attached while the
         loopback switch on the transceiver is in  the "Normal" position.
         In an EtherTwist installation, it will occur if loopback is attempted
         with a loopback fixture attached.
   "I" = Infinite deferral detected.  (Check terminators.)
         In a Thicklan installation, this could occur if the transceiver is not
         connected to the transceiver tap.  In either a Thicklan or a Thinlan
         installation, it could occur if the network cable is not terminated at
         both ends.  This could also indicate that the LANIC is
         defective.

   A new indicator is posted after each frame is sent.

   Press the interrupt character (usually <Control-C>)
   to stop the test.

   PPPPPInterrupt

P

    Type <return> after the AUI cable has been replaced:

End of Section 11 -- AUI Cable Fault Isolation
```

```
Section 12 -- Offline transceiver Test

    This section requires that the transceiver be removed from
    the LAN cable and connected to a terminated loopback fixture, or if
    your transceiver has a loopback switch, it must be set to 'normal'.

    Please input type of transceiver that this LANIC is connected to:
    Choices are:
    1) HP30241A (Thicklan Transceiver)
    2) HP28641A (Thinlan Transceiver)
    3) HP28664A (Starlan 10 or Twisted Pair Transceiver)
    4) HP28685A (EtherTwist Transceiver)
    5) HP28683A (Fiber-Optic Transceiver)
    6) OTHER

   Step 121 - Two Terminator Test

     Please connect the HP30241A transceiver to a HP92257B terminated
     loopback fixture.  Be certain that both 50-ohm terminators are
     firmly attached.


        The following activity indicators are used:
        "P" = External loopback was successful.
        "R" = Retry fault detected.  (Expected with one missing terminator.)
        "L" = Loss of carrier detected.  (Check AUI cable connections.)
        "I" = Infinite deferral detected.  (Check terminators.)

     A healthy transceiver should show 8 "P"'s...
     PPPPPPPP
   End of Step 121 - Two Terminator Test

   Step 122 - One Terminator Test

      Please connect the HP30241A transceiver to a HP92257B terminated
      loopback fixture.  Remove one 50-ohm terminator from the
      loopback fixture.
      Type <return> after terminator has been removed:

        The following activity indicators are used:
        "P" = External loopback was successful.
        "R" = Retry fault detected.  (Expected with one missing terminator.)
        "L" = Loss of carrier detected.  (Check AUI cable connections.)
        "I" = Infinite deferral detected.  (Check terminators.)

      A healthy transceiver should show 8 "R"'s...

      RRRRRRRR
      Type <return> after the terminator has been replaced:

   End of Step 122 - One Terminator Test
```

```
   End of Section 12 -- Offline Transceiver Test

   landad terminated (pid 7495).  Exit status = 0.
DUI 2>exit #
```

Test Section Descriptions

There are twelve diagnostic program sections available with LANDAD. You may also select individual steps to be run for sections 7 and 12. LANDAD's sections and steps are summarized below.

Section 1	More Help
Section 2	Reset
Section 3	Identify
Section 4	Local Loopback (to LANIC and back)
Section 5	Selftest
Section 6	Status
Section 7	Link Statistics
Step 71	Read and decode link statistics
Step 72	Reset link statistics
Section 8	External Loopback
Section 9	Remote Node Test
Section 10	Remote XID Test
Section 11	AUI Cable Fault Isolation Test
Section 12	Offline Transceiver Test
Step 121	Two Terminator Test
Step 122	One Terminator Test

Section 1—MORE HELP

Since the online diagnostics subsystem now has section help, this section is no longer needed. It is now unsupported, and if users specify this section, the following message will be displayed:

```
This Section is no longer supported.  For additional information
on a given section, use the section option of the help command
from the DUI prompt.  For example, if you want more information
about section 9, type the following:

DUI> help landad sections 9
```

Section 2—RESET

Reset causes a reset of the LANIC to its power on state. All pertinent data needed by the LANIC to operate properly will then be downloaded to the LANIC. If, after a reset of a LANIC that is offline due to bad hardware, the LANIC indicates that it passed its selftest, the system will put the LANIC into the online state.

Note

It is better to do a **SELFTEST** command (Section 5) to bring the LANIC back online because it checks status of the LANIC and displays what has failed if the LANIC is really broken.

If Section 2 completes successfully, it will put the LANIC into the online state, even if it is in the offline state when Section 2 is called.

If no errors are generated, the diagnostic will output the following message:

```
Section 2 -- Reset

End of Section 2 -- Reset
```

Section 3—IDENTIFY

Identify tells the system to issue a Status command to the LANIC. This command then decodes the information obtained and displays it in a manner that is informative to the user. This section can be used to determine what the LANIC hardware and firmware datecodes are, if applicable (only CIO and NIO have firmware and hardware datecodes). It is also useful in that if it executes successfully, the path from the diagnostic to the LANIC is at least partially functional.

Output:

If you are diagnosing a CIO LANIC:

```
Section 3 -- Identify

ID byte               = $06 (CIO LANIC).
Hardware revcode      = 2.
CIO firmware datecode = 2716.
CIO firmware ID       = 1.
NOVRAM (permanent) station address     = $08-00-09-00-AE-5B.
RAM (currently active) station address = $08-00-09-00-AE-5B.

End of Section 3 -- Identify
```

Note

If you are running this diagnostic on an MPE XL system, you will receive a driver revcode value in addition to the information displayed above.

If you are diagnosing an HP-PB LANIC:

```
Section 3 -- Identify

ID byte               = $06 (HP-PB LANIC).
HP-PB firmware part number 1 = 28652-81002.
HP-PB firmware part number 2 = 28652-81003.
Hardware revcode      = 2.
NOVRAM (permanent) station address     = $08-00-09-00-AE-5B.
RAM (currently active) station address = $08-00-09-00-AE-5B.

End of Section 3 -- Identify
```

Note

If you are running this diagnostic on an MPE XL system, you will receive a driver revcode value in addition to the information displayed above.

If the "NO SQE" jumper is configured, the following message will be displayed:

```
"NO SQE" (ETHERNET 1.0) jumper is configured.
```

If the card ID byte is not $06, $46, or $47, the following message is displayed:

```
*** ID byte              = ! (UNKNOWN PRODUCT);
*** Expecting $06 (CIO LANIC) or $46 (HP-PB LANIC) or $47 (VSC LANIC).
```

If you are running on an VSC LANIC:

```
Section 3 -- Identify

 ID byte              = $47 (VSC LANIC).
 NOVRAM (permanent) station address     = $08-00-09-FF-FF-FF
 RAM (currently active) station address = $08-00-09-FE-FE-FE
 Driver revcode       = $1

 End of Section 3 -- Identify
```

If the LANIC has previously issued a Protocol ERror (PER) or a Dead Or Dying (DOD) AES status message, the following message will be printed:

```
Last failure code = $n
```

The datecodes are displayed in decimal; the station addresses are displayed as 6 hex bytes.

Note

Although Identify does not send an external loopback frame onto the network cable, it does report the status of the last external loopback that was performed. It is possible that an error could be reported for an error condition that has already been fixed. If this is the case, an external loopback (Section 8) should be performed. This sends out an external loopback frame and will clear the error (assuming that the problem has been fixed).

If an application has previously logged SSAP $F4, LANDAD will give the user three chances to enter a new SSAP to log. If SSAP $F4 is currently in use, the following message is displayed:

```
*** WARNING -- SSAP n ALREADY IN USE BY ANOTHER PROCESS
    (LANDADWARN 6006)

Please enter a different SSAP to log to (one hex byte) =>
```

If you enter the same SSAP, the following error message is displayed and the question is asked again:

```
** New SSAP must be different from old SSAP.
```

If you enter a non-valid SSAP address, the following error message is displayed and the question is asked again:

```
** n is not a valid SSAP.  A valid SSAP is an even number
** between $02 and $FE.  The SSAP must be entered in hex.
** The leading '$' is optional.
```

If you enter a Return, the following error message is displayed and the question is asked again:

```
** There is no default for this question.
```

If you try three times and none of these is successful, the following message is displayed and the rest of this section is not executed:

```
*** WARNING -- DIAGNOSTIC SSAP IN USE BY ANOTHER PROCESS (LANDADWARN 6004)
*** Another process has bound to the diagnostic SSAP.
*** This section can only be run after the other process finishes.
```

Note

Once a good SSAP has been found to log, the diagnostic will use that same SSAP when logging within the same instance of the diagnostic. It is possible that between the time that the diagnostic has terminated and when it is run again, another process will come along and log to that SSAP. If that is the case, the diagnostic will ask the above question again.

Section 4—LOCAL LOOPBACK

Local Loopback opens up the system for normal use and transmits a frame addressed to itself. The LANIC will loop this frame back on the card and send it back to the diagnostic. This test will test the data path from the diagnostic to the card and back. The data in the frame is known and a byte-for-byte comparison of the data is made to be certain that the data was not corrupted. Since the LANIC only loops back the frame if the transmission onto the network medium is successful, this test also checks all components from the network medium to the driver. If the transmission is not successful, LANDAD prints out what it thinks the problem was with the transmission. If no errors are generated, the diagnostic will output the following message:

```
Section 4 -- Local Loopback

   Binding to DAM...(MPE XL) or Logging SSAP with driver...(HP-UX)
   Sending data to LAN Interface Controller...
   Receiving data from LAN Interface Controller...
   A frame was successfully transmitted onto the network media.
   Path to LAN Interface Controller is functional.
   Unbinding from DAM...(MPE XL)

End of Section 4 -- Local Loopback
```

Note

Since the packet is *not* sent onto the network when using a VSC LANIC, only the path to the driver is tested. Instead of the display above, you will see the following:

```
                              ************
                              *** NOTE ***
                              ************
  Since you are running this section on a VSC LANIC, this section
  will only test the path through the driver.  To test the path to
  the hardware, run  EXTERNAL LOOPBACK (section 8) or any
  destructive test (sections 2,5,8,11,12).
  Sending data to Driver...
  Receiving data from Driver...
  A frame has been successfully transmitted through the Driver.
  Path to Driver is functional.
```

If an application has previously logged SSAP $F4, LANDAD will give the user three chances to enter a new SSAP to log. If SSAP $F4 is currently in use, the following message is displayed:

```
*** WARNING -- SSAP n ALREADY IN USE BY ANOTHER PROCESS
    (LANDADWARN 6006)

    Please enter a different SSAP to log to (one hex byte) =>
```

If you enter the same SSAP, the following error message is displayed and the question is asked again:

```
** New SSAP must be different from old SSAP.
```

If you enter a non-valid SSAP address, the following error message is displayed and the question is asked again:

```
** n is not a valid SSAP.  A valid SSAP is an even number
** between $02 and $FE.  The SSAP must be entered in hex.
** The leading '$' is optional.
```

If you enter a Return, the following error message is displayed and the question is asked again:

```
** There is no default for this question.
```

If you try three times and none of these is successful, the following message is displayed and the rest of this section is not executed:

```
*** WARNING -- DIAGNOSTIC SSAP IN USE BY ANOTHER PROCESS (LANDADWARN 6004)
*** Another process has bound to the diagnostic SSAP.
*** This section can only be run after the other process finishes.
```

Note

Once a good SSAP has been found to log, the diagnostic will use that same SSAP when logging within the same instance of the diagnostic. It is possible that between the time that the diagnostic has terminated and when it is run again, another process will come along and log to that SSAP. If that is the case, the diagnostic will ask the above question again.

Section 5—SELFTEST

Selftest tells the LANIC to perform a hardware selftest. If the returned selftest status is abnormal, messages indicating the problem are displayed. Since selftest brings the card offline and aborts all current information transfers, it should only be done when necessary. If selftest passes, it will put the LANIC into the online state. If message LANDADWARN 6005 is ever displayed, Section 6 (Status) should be executed to see why the card went offline and then selftest should be performed to determine if the problem is still present.

Note

If this section completes successfully, it will put the LANIC into the online state, even if it is in the offline state when the section is called.

If no errors are generated, the diagnostic will output the following message:

```
Section 5 -- Selftest

   Selftest completed successfully.
   The LAN Interface Controller is functional.

End of Section 5 -- Selftest
```

Section 6—STATUS

Status is used to obtain information about the current state of the LANIC. If no errors are generated, the diagnostic will output the following message:

If you are diagnosing a CIO LANIC:

```
Section 6 -- Status

     LANIC status has been read successfully.
     LANIC passed selftest.
     LANIC is online.
     transceiver power fuse is OK.
     Free transmit buffers = n; Maximum = n.
     Full receive buffers  = n; Maximum = nn.
     Read data ARQ frame threshold = n.
     Read data ARQ timeout limit  n.
                          RAM value              NOVRAM value
=========================(Current)=============(Default)========
Station address           $nn-nn-nn-nn-nn-nn$nn-nn-nn-nn-nn-nn
Receive bad frames        Disabled               Disabled
Receive multicast frames  Enabled                Enabled
Receive broadcast frames  Enabled                Enabled
Receive all frames        Disabled               Disabled
=================================================================
The following multicast addresses are recognized:
$09-00-09-00-00-01
$09-00-09-00-00-02

   End of Section 6 -- Status
```

If you are diagnosing an HP-PB LANIC:

```
     Section 6 -- Status

     LANIC status has been read successfully.
     LANIC passed selftest.
     LANIC is online.
     transceiver power fuse is OK.
     Free transmit buffers = n; Maximum = n.
     Full receive buffers  = n; Maximum = nn.
     Using (internal/external) transceiver.
                             RAM value            NOVRAM value
     ========================(Current)===========(Default)=========
     Station address         $nn-nn-nn-nn-nn-nn$nn-nn-nn-nn-nn-nn
     Receive multicast frames  Enabled              Not Applicable
     Receive broadcast frames  Enabled              Not Applicable
     ================================================================
     The following multicast addresses are recognized:
     $09-00-09-00-00-01
     $09-00-09-00-00-02

     End of Section 6 -- Status
```

If the LANIC failed selftest, the following message will replace the LANIC Passed Selftest message:

```
*** LANIC failed selftest. *** REPLACE THE LANIC ***
```

If the LANIC's power fuse is blown, the following message is displayed:

```
*** Transceiver power fuse is BLOWN. *** REPLACE FUSE ***
```

If the LANIC is offline, the following message replaces the LANIC is online message:

```
LANIC is offline.
```

The entire multicast list is printed. If no multicast addresses are in the multicast list, the following message is printed:

```
NO multicast addresses were found.
```

If the no SQE jumper is installed, the following message is printed:

```
"NO SQE" (Ethernet 1.0) jumper is configured.
```

The HP-PB LANIC may also display:

```
Remote reset enabled.
```

or

```
***INTERNAL/EXTERNAL TRANSCEIVER JUMPER MISSING OR MISALIGNED.
```

If you are diagnosing a VSC LANIC:

```
Section 6 -- Status

LANIC status has been read successfully.
LANIC passed selftest.
LANIC is online.
AUI power fuse is OK.
Using External Transceiver.
"NO SQE" (Ethernet 1.0) jumper is configured.
                          RAM value           NOVRAM value
=========================(Current)===========(Default)=========
Station address           $08-00-09-FE-FE-FE  $08-00-09-FF-FF-FF
Receive multicast frames  Enabled             Not applicable
Receive broadcast frames  Enabled             Not applicable
=================================================================
The following multicast addresses are recognized:
No multicast addresses were found.

End of Section 6 -- Status
```

If the AUI power fuse is not blown, the following message will be displayed:

```
AUI power fuse is OK.
```

If the LANIC's AUI power fuse is blown, the following message will be displayed:

```
*** AUI power fuse is BLOWN. *** REPLACE FUSE ***
```

Explanation of Status Values for CIO LANIC

This is a line by line explanation of the fields that the status command returns. The following is a listing of a typical status display with line numbers added. This listing is used to show which line number is being explained.

```
 1 =>   Section 6 -- Status
 2 =>
 3 =>   LANIC status has been read successfully.
 4 =>   LANIC passed selftest.
 5 =>   LANIC is online.
 6 =>   Transceiver power fuse is OK.
 7 =>   Free transmit buffers =     4; Maximum =     4.
 8 =>   Full receive buffers  =     0, Maximum =    16.
 9 =>   Read data ARQ frame threshold =     1.
10 =>   Read data ARQ timeout limit   =     1.
11 =>                            RAM value             NOVRAM value
12 =>   ========================(Current)============(Default)========
13 =>   Station address          $08-00-09-00-14-01   $08-00-09-00-14-01
14 =>   Receive bad frames       Disabled              Disabled
15 =>   Receive multicast frames Enabled               Enabled
16 =>   Receive broadcast frames Enabled               Enabled
17 =>   Receive all frames       Disabled              Disabled
18 =>   ================================================================
19 =>   The following multicast addresses are recognized:
20 =>   $09-00-09-00-00-01
21 =>   $09-00-09-00-00-02
22 =>
23 =>   End of Section 6 -- Status
```

LANIC selftest status (line 4)

This line indicates whether the LANIC passed or failed selftest. If the LAN passed selftest, the following message will be displayed:

```
LANIC passed selftest.
```

If it failed selftest, the following message will be printed:

```
*** LANIC failed selftest. *** REPLACE THE LANIC ***
```

If this message is displayed, the LANIC interface card should be replaced. Note also that the rest of the information may or may not be valid.

LANIC online or offline status (line 5)

This indicates if the LANIC is online or offline. Online means that the LANIC is ready to transmit and receive frames from the network media.

Transceiver power fuse status (line 6)

This line indicates the status of the transceiver power fuse. If the fuse is OK, the following message is displayed:

```
Transceiver power fuse is OK.
```

If the fuse is blown, the following message is printed:

```
*** Transceiver power fuse is BLOWN. *** REPLACE FUSE ***
```

Free transmit buffers (line 7)	This is the number of buffers that the LANIC has currently available for transmit frames. There is a one for one correspondence between frames and buffers. The maximum will equal 4 for the current CIO LANIC. It is normal for this number to vary between 0 and the maximum during normal network usage.
Full receive buffers (line 8)	This is the number of buffers that the LANIC has received from the network media that have yet to be read by the driver. There is a one for one correspondence between frames and buffers. The maximum will equal 16 for the current CIO LANIC. It is normal for this number to vary between 0 and the maximum during normal network usage.
Read data ARQ frame threshold (line 9)	This is the number of frames that must be received before the CIO LANIC will interrupt the host. If this value is non-zero and the read data ARQ timeout limit is zero, the CIO LANIC will interrupt as soon as a frame arrives.
Read data ARQ timeout limit (line 10)	This is the number of 10 millisecond periods that the CIO LANIC will interrupt the host after receiving a frame from the link but not receiving read ARQ frame threshold frames from the network media. If this value is non-zero and the frame threshold is zero, the CIO LANIC will interrupt as soon as a frame arrives. Parameters in lines 13 through 17 have two values. The first is the RAM value. This is the value that the LANIC is currently using. The second is the NOVRAM value. This is the default value for that parameter that the LANIC will use when it is initially powered up.
Station address (line 13)	This is the six byte address that the LANIC will respond to. This address will also be used when sending out frames to the network media.
Receive bad frames (line 14)	This is either enabled or disabled. When enabled, the CIO LANIC will save bad frames that it receives and pass them up to the driver. When disabled, bad frames will be counted in statistics, but will be discarded.
Receive multicast frames (line 15)	This is either enabled or disabled. When enabled, the LANIC will receive frames sent to multicast addresses that it has been set up to receive. A list of up to 64 multicast addresses can be downloaded to the LANIC.
Receive broadcast frames (line 16)	This is either enabled or disabled. When enabled, the LANIC will receive frames sent to the broadcast address.
Receive all frames (line 17)	This is either enabled or disabled. When enabled, the CIO LANIC will attempt to receive all frames from the network media. When disabled, the LANIC will only receive frames sent to it's address and if receive broadcast frames is enabled, also frames sent to the broadcast address.
The following multicast addresses are recognized (line 19)	This is a list of all multicast addresses that the LANIC will respond to.

Note On HP-UX, only the first 16 multicast addresses can be displayed.

Explanation of Status Values for HP-PB LANIC

This is a line by line explanation of the fields that the status command returns. The following is a listing of a typical status display with line numbers added. This listing is used to show which line number is being explained.

```
 1 =>   Section 6 -- Status
 2 =>
 3 =>   LANIC status has been read successfully.
 4 =>   LANIC passed selftest.
 5 =>   LANIC is online.
 6 =>   transceiver power fuse is OK.
 7 =>   Free transmit buffers =     4; Maximum =     4.
 8 =>   Full receive buffers  =     0; Maximum =    32.
 9 =>   Using INTERNAL transceiver.
10 =>                           RAM value              NOVRAM value
11 =>   ========================(Current)==============(Default)========
12 =>   Station address          $08-00-09-00-14-01   $08-00-09-00-14-01
13 =>   Receive multicast frames  Enabled               Not applicable
14 =>   Receive broadcast frames  Enabled               Not applicable
15 =>   ================================================================
16 =>   The following multicast addresses are recognized:
17 =>   $09-00-09-00-00-01
18 =>   $09-00-09-00-00-02
19 =>
20 =>   End of Section 6 -- Status
```

LANIC selftest status (line 4)

This line indicates whether the LANIC passed or failed selftest. If the LANIC passed selftest, the following message will be displayed:

```
LANIC passed selftest.
```

If it failed selftest, the following message will be printed:

```
*** LANIC failed selftest. *** REPLACE THE LANIC ***
```

If this message is displayed, the LANIC interface card should be replaced. Note also that the rest of the information may or may not be valid.

LANIC online or offline status (line 5)

This indicates if the LANIC is online or offline. Online means that the LANIC is ready to transmit and receive frames from the network media.

Transceiver power fuse status (line 6)

This line indicates the status of the transceiver power fuse. If the fuse is OK, the following message is displayed:

```
Transceiver power fuse is OK.
```

If the fuse is blown, the following message is printed:

```
*** Transceiver power fuse is BLOWN. *** REPLACE FUSE ***
```

Free transmit buffers (line 7) This is the number of buffers that the LANIC has currently available for transmit frames. There is a one for one correspondence between frames and buffers. The maximum will equal 4 for the current HP-PB LANIC. It is normal for this number to vary between 0 and the maximum during normal network usage.

Full receive buffers (line 8) This is the number of buffers that the LANIC has received from the network media that have yet to be read by the driver. There is a one for one correspondence between frames and buffers. The maximum will equal 32 for the current HP-PB LANIC. It is normal for this number to vary between 0 and the maximum during normal network usage.

Using INTERNAL or EXTERNAL Transceiver (line 9) This indicates the state of the internal/external transceiver jumpers. If the jumper is on the INTERNAL transceiver block, the Using INTERNAL transceiver message is printed. If the jumper is on the EXTERNAL block, the Using EXTERNAL transceiver message is displayed. If the jumper is missing or misaligned, the following error message is displayed:

```
*** INTERNAL/EXTERNAL TRANSCEIVER JUMPER MISSING OR MISALIGNED.
```

Parameters in lines 12 through 14 have two values. The first is the RAM value. This is the value that the LANIC is currently using. The second is the NOVRAM value. This is the default value for that parameter that the LANIC will use when it is initially powered up.

Station address (line 12) This is the six byte address that the LANIC will respond to. This address will also be used when sending out frames to the network media.

Receive multicast frames (line 13) This is either enabled or disabled. When enabled, the LANIC will receive frames sent to multicast addresses that it has been set up to receive. A list of up to 64 multicast addresses can be downloaded to the LANIC. Upon power-on or reset, this is disabled.

Receive broadcast frames (line 14) This is either enabled or disabled. When enabled, the LANIC will receive frames sent to the broadcast address. Upon power-on or reset, this is enabled.

The following multicast addresses are recognized (line 16) This is a list of all multicast addresses that the LANIC will respond to.

Note On HP-UX, only the first 16 multicast addresses can be displayed.

Explanation of Status Values for VSC LANIC

This is a line by line explanation of the fields that the status command returns. The following is a listing of a typical status display with line numbers added. This listing is used to show which line number is being explained.

```
 1 =>   Section 6 -- Status
 2 =>
 3 =>   LANIC status has been read successfully.
 4 =>   LANIC passed selftest.
 5 =>   LANIC is online.
 6 =>   AUI power fuse is OK.
 7 =>   Using INTERNAL transceiver.
 8 =>                              RAM value             EEPROM value
 9 =>   ==========================(Current)=============(Default)========
10 =>   Station address            $08-00-09-00-14-01    $08-00-09-00-14-01
11 =>   Receive multicast frames   Enabled                Not applicable
12 =>   Receive broadcast frames   Enabled                Not applicable
13 =>   ================================================================
14 =>   The following multicast addresses are recognized:
15 =>   $09-00-09-00-00-01
16 =>   $09-00-09-00-00-02
17 =>
18 =>   End of Section 6 -- Status
```

LANIC selftest status (line 4) This line indicates whether the LANIC passed or failed selftest. If the LANIC passed selftest, the following message will be displayed:

```
LANIC passed selftest.
```

If it failed selftest, the following message will be printed:

```
*** LANIC failed selftest. *** REPLACE THE LANIC ***
```

If this message is displayed, the LANIC interface card should be replaced. Note also that the rest of the information may or may not be valid.

LANIC online or offline status (line 5) This indicates if the LANIC is online or offline. Online means that the LANIC is ready to transmit and receive frames from the network media.

AUI power fuse status (line 6) This line indicates the status of the AUI power fuse. If the fuse is OK, the following message is displayed:

```
AUI power fuse is OK.
```

If the fuse is blown, the following message is printed:

```
*** AUI power fuse is BLOWN. *** REPLACE FUSE ***
```

Using INTERNAL or EXTERNAL Transceiver (line 7)

This indicates the state of the internal/external transceiver jumpers. If the jumper is on the INTERNAL transceiver block, the Using INTERNAL Transceiver message is printed. If the jumper is on the EXTERNAL block, the Using EXTERNAL Transceiver message is displayed. If the jumper is missing or misaligned, the following error message is displayed:

```
*** INTERNAL/EXTERNAL TRANSCEIVER JUMPER MISSING OR MISALIGNED.
```

Parameters in lines 10 through 12 have two values. The first is the RAM value. This is the value that the LANIC is currently using. The second is the EEPROM value. This is the default value for that parameter that the LANIC will use when it is initially powered up.

Station address (line 10)

This is the six byte address that the LANIC will respond to. This address will also be used when sending out frames to the network media.

Receive multicast frames (line 11)

This is either enabled or disabled. When enabled, the LANIC will receive frames sent to multicast addresses that it has been set up to receive. A list of up to 64 multicast addresses can be downloaded to the LANIC. Upon power-on or reset, this is disabled.

Receive broadcast frames (line 12)

This is either enabled or disabled. When enabled, the LANIC will receive frames sent to the broadcast address. Upon power-on or reset, this is enabled.

The following multicast addresses are recognized (line 14)

This is a list of all multicast addresses that the LANIC will respond to.

Note

On HP-UX, only the first 16 multicast addresses can be displayed.

Section 7—LINK STATISTICS

Link Statistics allows you to read and display link statistics that the LANIC keeps. It also allows you to reset these link statistics.

This function has two steps: Step 71 is the default step. It reads link statistics from the LANIC and decodes them. To invoke this step, enter:

```
DUI > run landad ldev=/dev/diag/lan0 section=7
```

The second function is the reset statistics function, Step 72. This function is disruptive since it modifies data on the LANIC. Since it is not a default step, you must specify step=72 in the RUN command, as follows:

```
DUI > run landad ldev=/dev/diag/lan0 section=7 [72]
```

Step 71 - Read and Display Link Statistics

This step requests link level statistics from the LANIC through the system and displays the statistics. If no errors are generated, the diagnostic will output the following message:

```
Section 7 -- Link Statistics

   Step 71 - Read and Display Link Statistics
   Link level statistics have been read successfully.

   Transmit Statistics ==================================
     TOTAL frames transmitted without error.....n
        Deferred transmits.....n
        One collision transmits.....n
        More than one collision transmits.....n
     TOTAL frames NOT transmitted.....n
        Retry errors.....n
        Late collision.....n
        Loss of carrier during transmit.....n
     No heartbeat detected after transmission.....n
     No free transmit buffers.....n
     TDR of last retry error.....n
     Indefinite deferral errors.....n
   Receive Statistics====================================
     TOTAL frames received without error.....n
     Frames rejected by address filter.....n
     Frames rejected due to CRC errors.....n
     Frames rejected due to alignment errors.....n
     Frames rejected due to oversize length.....n
     LAN Controller indicated one or more frames lost..n
     No free receive buffers.....n
   End of Step 71 - Read and Display Link Statistics

End of Section 7 -- Link Statistics
```

Step 72 - Reset Link Statistics

This section resets the link statistics on the LANIC. If no errors are generated, the diagnostic will output the following message:

```
Section 7 -- Link Statistics

   Step 72 - Reset Link Statistics
   Link statistics reset successfully.
   Link level statistics have been read successfully

   Transmit Statistics ================================
     TOTAL frames transmitted without error.....0
        Deferred transmits.....0
        One collision transmits.....0
        More than one collision transmits.....0
     TOTAL frames NOT transmitted.....0
        Retry errors.....0
        Late collision.....0
        Loss of carrier during transmit.....0
     No heartbeat detected after transmission.....0
     No free transmit buffers.....0
     TDR of last retry error.....0
     Indefinite deferral errors.....0
   Receive Statistics==================================
     TOTAL frames received without error.....0
     Frames rejected by address filter.....0
     Frames rejected due to CRC errors.....0
     Frames rejected due to alignment errors.....0
     Frames rejected due to oversize length.....0
     LAN Controller indicated one or more frames lost..0
     No free receive buffers.....0
   End of Step 72 - Reset Link Statistics

End of Section 7 -- Link Statistics
```

Section 8—EXTERNAL LOOPBACK

External Loopback first takes the card offline and then tells the LANIC to perform an external loopback test. This test transmits and receives a frame from the network cable. If this test passes, the following things have a high probability of being functional:

- This network cable segment
- Both 50 ohm terminators
- This node's transceiver tap
- This node's transceiver
- This node's AUI cable(s)
- This node's stub cable
- This node's LANIC

Note

If this section completes successfully, it will put the LANIC into the online state, even if it is in the offline state when the section is called.

If no errors are generated, the diagnostic will output the following message:

```
Section 8 -- External Loopback

   A link frame has been successfully transmitted
   and received from the network cable.

End of Section 8 -- External Loopback
```

Section 9—REMOTE NODE TEST

Remote Node tests the ability of this node to bounce a packet off another node connected to the same physical (or logical if there are repeaters and/or bridges in the network) network. This is useful for two reasons: First, it illustrates that the node can communicate with a remote node. Second, it can point to upper level software problems. If a frame can be bounced off another node using the diagnostic, but normal NS communications do not work, the problem is not the hardware, it's the upper level software.

This section sends an IEEE 802.2 test frame. This test frame can be any length from 60 bytes (a minimum 802.3 frame) to 1514 bytes (a maximum length 802.3 frame). The default is 500 bytes. When a test response frame is received from the remote station, its length is checked for being either a minimum size frame or for being the specified length -0/+1. If the response frame is not a minimum size frame, then the data is checked against the data sent. If it is not the same, then the test frame part of the test fails.

This section will allow communication only to individual network addresses. If you input a broadcast or multicast address as a response to the Remote Node Address prompt, an error message will be issued and you will be prompted again for a valid remote node address.

Note

Part of what this section tests is the receive threshold of the transceiver at the two nodes involved in the test. The worst case for this is when the two nodes are the maximum 500 meters (185 meters for ThinLAN) apart. Therefore, it is best if you attempt to bounce frames off a distant node.

Caution

The remote node MUST be capable of responding to IEEE 802.2 test frames, and that node must be in a state to answer those frames. For example, most systems must have the LANIC device driver installed and operating before test frames will be answered.

If no errors are generated, the diagnostic will output the following message:

```
Section 9 -- Remote Node Test

   This Section sends a TEST frame and waits for a response from
   a specified remote node for a specified number of iterations.

   The following success/failure indicators are used:
   "." = The test frame bounced successfully.
   "#" = The test frame was not received before the timeout period.

   Remote Node Address (Six HEX bytes) => 0800 0900 1401
   Number of test frames to send ("0" for infinite) [10] =>
   Length of test frames in bytes (60..1514) [500] =>
   Press the interrupt character (usually <Control-C>)
   to prematurely stop the test.

   ..........

   10 out of 10 TEST frames echoed successfully (100%).

End of Section 9 -- Remote Node Test
```

If the test is being run on an HP-UX system, the message is:

```
Press the interrupt character (usually <CTRL-C>)
to stop the test.
```

If you input an invalid remote node address, the following message will be displayed and you will be prompted again:

```
Address must be 12 hexadecimal digits with any combination of delimiters
Hexadecimal digits are in the set:
['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'].
Hexadecimal digits may be either upper or lower case.

Delimiters are in the set: [' ','-','$']
Examples:
  080009123ABC
  08-00-09-12-3A-BC
  $0800-0912-3ABC
  08 00    09 12   3a Bc
  0-$ 800--$-09-1$2--3abc
```

Only the first twelve digits are read including zeros. Since the address must be an individual address, the low bit of the high byte of the address is checked to see if it is set. If it is, the following message is displayed and you are re-prompted for the address:

```
Address must be an individual address i.e. the most significant byte
of the address must be even.  (An odd first byte indicates a group
address.)
Examples:
  Correct:
    08 00 09 00 12 AB

  Incorrect:
    FF FF FF FF FF FF
    09 00 09 00 00 01
    01 02 03 04 05 06
```

If an application has previously logged SSAP $F4, LANDAD will give you three chances to enter a new SSAP to log to. If SSAP $F4 is currently in use, the following message is displayed:

```
*** WARNING -- SSAP n ALREADY IN USE BY ANOTHER PROCESS
    (LANDADWARN 6006)

    Please enter a different SSAP to bind to =>
```

If you enter the same SSAP, the following error message is displayed and the question is asked again:

```
** New SSAP must be different from old SSAP.
```

If you enter a non-valid SSAP address, the following error message is displayed and the question is asked again:

```
** n is not a valid SSAP.  A valid SSAP is an even number
** between $02 and $FE.  The SSAP must be entered in hex.
** The leading '$' is optional.
```

If you enter a (Return), the following error message is displayed and the question is asked again.

```
** There is no default for this question.
```

If you try three times and none of these is successful, the following message is displayed and the rest of this section is not executed:

```
*** WARNING -- DIAGNOSTIC SSAP IN USE BY ANOTHER PROCESS (LANDADWARN 6004)
*** Another process has bound to the diagnostic SSAP.
*** This section can only be run after the other process finishes.
```

Note

Once a good SSAP has been found to log, the diagnostic will use that same SSAP when logging within the same instance of the diagnostic. It is possible that between the time that the diagnostic has terminated and when it is run again, another process will come along and log to that SSAP. If that is the case, the diagnostic will ask the above question again.

Section 10—REMOTE XID TEST

Remote XID Test allows you to send IEEE 802.2 XID command frames to a specified remote node and receive the response frame from the remote node. This section also decodes the response and displays what type of service is available at the remote node.

You are prompted for both the 6-byte remote node address and the 1-byte DSAP of the service on the remote system to which the XID frame should be sent. The addresses that you give must be individual addresses, i.e., they cannot be broadcast or multicast. Should you input one of these illegal addresses, an error message will be issued and you will be prompted again for a valid address. If no errors are generated, the diagnostic will output the following message:

```
Section 10 -- Remote XID Test

  This section sends an IEEE 802.2 XID frame to a user specified
  remote node and waits for an IEEE 802.2 XID response frame from
  that remote node.

  Remote Node Address (Six HEX bytes) => 0800 0900 1401
  Remote DSAP Address (one even hex byte between $00 and $FE) [$00] =>
  Sending XID command frame...
  Received XID response frame...

  Remote DSAP n has class I service.

End of Section 10 -- Remote XID Test
```

If the remote DSAP has class II service, the following message is displayed:

```
Remote DSAP n has class II service, window size = n.
```

If the remote DSAP has class III service, the following message is displayed:

```
Remote DSAP n has class III service.
```

If the remote DSAP has class IV service, the following message is displayed:

```
Remote DSAP n has class IV service.
```

If the remote node does not send back a response frame, the following message is displayed:

```
No response received from remote node.
```

If you input an invalid remote node address, the following message will be displayed and you will be prompted again:

```
Address must be 12 hexadecimal digits with any combination of delimiters
Hexadecimal digits are in the set:
['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'].
Hexadecimal digits may be either upper or lower case.
Delimiters are in the set: [' ','-','$']
Examples:
  080009123ABC
  08-00-09-12-3A-BC
  $0800-0912-3ABC
  08 00    09 12   3a Bc
  0-$ 800--$-09-1$2--3abc
```

Since the address must be an individual address, the low bit of the high byte of the address is checked to see if it is set. If it is, the following message is displayed and you are re-prompted for the address:

```
Address must be an individual address i.e. the most significant byte
of the address must be even.  (An odd first byte indicates a group address.)
Examples:
  Correct:
   08 00 09 00 12 AB

  Incorrect:
    FF FF FF FF FF FF
    09 00 09 00 00 01
    01 02 03 04 05 06
```

If an application has previously logged SSAP $F4, LANDAD will give you three chances to enter a new SSAP to log to. If SSAP $F4 is currently in use, the following message is displayed:

```
*** WARNING -- SSAP n ALREADY IN USE BY ANOTHER PROCESS
    (LANDADWARN 6006)

Please enter a different SSAP to bind to =>
```

If you enter the same SSAP, the following error message is displayed and the question is asked again:

```
** New SSAP must be different from old SSAP.
```

If you enter a non-valid SSAP address, the following error message is displayed and the question is asked again:

```
** n is not a valid SSAP.  A valid SSAP is an even number
**  between $02 and $FE.  The SSAP must be entered in hex.
**  The leading '$' is optional.
```

If you enter a (Return), the following error message is displayed and the question is asked again:

```
** There is no default for this question.
```

If you try three times and none of these is successful, the following message is displayed and the rest of this section is not executed:

```
*** WARNING -- DIAGNOSTIC SSAP IN USE BY ANOTHER PROCESS (LANDADWARN 6004)
*** Another process has bound to the diagnostic SSAP.
*** This section can only be run after the other process finishes.
```

Note

Once a good SSAP has been found to log to, the diagnostic will use that same SSAP when logging within the same instance of the diagnostic. It is possible that between the time that the diagnostic has terminated and when it is run again, another process will come along and log to that SSAP. If that is the case, the diagnostic will ask the above question again.

Section 11—AUI CABLE FAULT ISOLATION

AUI Cable Fault Isolation isolates a broken cable in the AUI cable segment. This is done by repeatedly sending external loopback frames and checking to see if the frame loopback was successful. If it was successful, a "P" is printed. If it was not, the reason that it was not is given on the screen. If the reason was loss of carrier error, an "L" is printed. This might indicate a broken AUI cable. If the reason is retry fault, an "R" is printed. This may indicate that the problem is a bad loopback hood.

To run this section, you first connect a terminated loopback fixture to the end of the stub cable. Then the test is started. Next, you disconnect the loopback hood at the stub cable, reconnect the stub cable to the AUI cable, and connect the loopback fixture to the opposite end of the AUI cable. If there are multiple AUI cables, you continue to do this until you get to the AUI cable which connects to the transceiver. After doing all of this, the test is stopped (via [&[CTRL]]-C on HP-UX or [&[CTRL]]-Y on MPE XL) and the pattern of activity indicators is analyzed. If the pattern looks like this:

```
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
```

The problem is most likely a bad stub cable. If the pattern looks like this:

```
PPPPLLLLPPPPLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
```

The problem is most likely the second AUI cable, or if there is only a single AUI cable, then the transceiver. (This may not be obvious. The first set of Ps is where the stub cable passed the test. The next set of Ls is where the loopback hood was disconnected to connect it back to the AUI cable. The next set of Ps is where the first AUI cable passed. At this point, the problem is beyond the first AUI cable.)

Note: The last thing that this section does is reset the LANIC and the system. If the reset is successful, the LANIC is put into the online state, even if it is in the offline state when the section is started.

If no errors are generated, the diagnostic will output the following message:

```
Section 11 -- AUI Cable Fault Isolation

   This section sends a number of external loopback frames out to a
   transceiver connected to a terminated loopback fixture
   to form a loopback hood.  The status from each frame is displayed on the
   screen.  By moving the transceiver and loopback fixture to various
   AUI cable connector junctions, faults in the AUI cable can be located.

   The following activity indicators are used:
   "P" = External loopback was successful.
   "L" = Loss of carrier detected.
         This usually indicates that there is a loose AUI connector or a broken
         AUI cable between the LANIC and the loopback connector.  In
         a Starlan 10 (Twisted Pair Transceiver) or Fiber-Optic installation,
         it will occur if loopback is attempted without a loopback fixture
         attached while the loopback switch on the transceiver is in the "Test"
         position.  In an EtherTwist installation, it may indicate that the
         setting of the "linkbeat" switch on the local transceiver is
         incompatible with the remote transceiver.
```

```
   "R" = Retry fault detected.  (Check loopback transceiver and
         loopback fixture.)
         In a Thicklan or Thinlan installation, this usually indicates that
         there is a loose or broken terminator on the loopback hood, or a
         transceiver or transceiver tap/tee problem.  In a Starlan 10 (Twisted
         Pair Transceiver) or Fiber-Optic installation, it will occur if
         loopback is attempted with a loopback fixture attached while the
         loopback switch on the transceiver is in  the "Normal" position.
         In an EtherTwist installation, it will occur if loopback is attempted
         with a loopback fixture attached.
   "I" = Infinite deferral detected.  (Check terminators.)
         In a Thicklan installation, this could occur if the transceiver is not
         connected to the transceiver tap.  In either a Thicklan or a Thinlan
         installation, it could occur if the network cable is not terminated at
         both ends.  This could also indicate that the LANIC is
         defective.

   A new indicator is posted after each frame is sent.

   Press <control-Y> to stop the test.

   PPPPLLLLPPPPLLLLLLLLLL CTRL-Y
   Type <Return> after the AUI cable has been replaced.
End of Section 11 -- AUI Cable Fault Isolation
```

If the diagnostic is being run on an HP-UX machine, the "<CTRL-Y> to stop" message is replaced by the following message:

```
   Press the interrupt character (usually <CTRL-C>) to stop the test.
```

If the device being diagnosed is an HP-PB LANIC or an VSC LANIC, and the internal transceiver is configured, the following message is displayed and the section is not executed:

```
   This test is not valid when using the Internal Thintransceiver
   on the HP-PB LANIC or the VSC LANIC

   No AUI Cable Fault Isolation Test Performed.
```

Note The HP-PB LANIC has no stub cable.

Section 12—OFFLINE TRANSCEIVER TEST

The Offline Transceiver Test provides a way to verify that a transceiver is operating properly. In order to run this test, the transceiver should be taken off the network cable and a terminated loopback hood (HP 92257B or HP 92257Q) should be attached to it.

There are two different steps in this section. The first step requires that

- For the HP30241A (Thicklan Transceiver) and HP28641A (Thinlan Transceiver), the loopback fixture should be connected to the transceiver. Both terminators must be attached to the loopback fixture.
- For the HP28664A (Starlan 10 or Twisted Pair Transceiver) or the HP28683A (Fiber-Optic Transceiver), the loopback fixture should be attached to the transceiver. The loopback switch on the transceiver should be set to the "Test" position.
- For the HP28685A (EtherTwist Transceiver), no loopback fixture should be used. The linkbeat switch on the transceiver should be set to the "disabled" position.

Eight external loopback frames are then sent to the transceiver. These packets should be sent successfully, which will be indicated by "P"'s being displayed.

The second section requires that

- For the HP30241A (Thicklan Transceiver) and HP28641A (Thinlan Transceiver), one of the 50-ohm terminators should be removed from the loopback fixture.
- For the HP28664A (Starlan 10 or Twisted Pair Transceiver) or HP28683A (Fiber-Optic Transceiver), the loopback fixture should be attached to the transceiver. The loopback switch on the transceiver should be set to the "Normal" position.
- For the HP28685A (EtherTwist Transceiver), the loopback fixture should be attached to the transceiver. The linkbeat switch on the transceiver should be set to the position in which it is used in normal operation.

Eight more external loopback frames are then sent to the transceiver. These should fail because of retry errors, indicated by "R"'s being displayed. If the diagnostic displays "P"'s for the first section and "R"'s for the second section, the transceiver passes. If the diagnostic displays "P"'s for both sections, the transceiver's collision detection circuitry is not working and the transceiver should be replaced. If the diagnostic displays "R"'s for both sections, the TLF may have a loose terminator or be broken, or the transceiver may be broken. If the diagnostic displays "L"'s for both sections, there may be a short in the TLF, or an AUI cable may be bad, or the transceiver may be broken.

If you are using the internal Thin transceiver on the HP-PB LANIC or the VSC LANIC, the transceiver type question is not asked.

Note

The last thing that this section does is to reset the LANIC and the DAM or driver. If the reset is successful, the LANIC is put into the online state, even if it was in the offline state when the section was called.

If no errors are generated, the diagnostic will output the following message:

```
Section 12 -- Offline Transceiver Test

   This section requires that the transceiver be removed from the LAN
   cable and connected to a terminated loopback fixture.

   Please input type of transceiver that this LANIC is connected to:
   Choices are:
    1) HP30241A (Thicklan Transceiver)
    2) HP28641A (Thinlan Transceiver)
    3) HP28664A (Starlan 10 or Twisted Pair Transceiver)
    4) HP28685A (EtherTwist Transceiver)
    5) HP28683A (Fiber-Optic Transceiver)
    6) OTHER
   Please select transceiver type (1-3) => 1

   Step 121 - Two Terminator Test

      Please connect the HP30241A transceiver to a HP92257B terminated
      loopback fixture.  Be certain that both 50-ohm terminators are
      firmly attached.
      Type <return> when loopback hood is connected:

      The following activity indicators are used:
      "P" = External loopback was successful.
      "R" = Retry fault detected.  (Expected with one missing terminator.)
      "L" = Loss of carrier detected.  (Check AUI cable connections.)
      "I" = Infinite deferral detected.  (Check terminators.)

      A healthy transceiver should show 8 "P"s...

      PPPPPPPP

   End of Step 121 - Two Terminator Test
```

```
    Step 122 - One Terminator Test

      Please connect the HP30241A transceiver to a HP92257B terminated
      loopback fixture.  Remove one 50-ohm terminator from the loopback
      fixture.
      Type <return> after terminator has been removed:

      The following activity indicators are used:
      "P" = External loopback was successful.
      "R" = Retry fault detected.  (Expected with one missing terminator.)
      "L" = Loss of carrier detected.  (Check AUI cable connections.)
      "I" = Infinite deferral detected.  (Check terminators.)

      A healthy transceiver should show 8 "R"s...

      RRRRRRRR
      Type <Return> after the terminator has been replaced.
   End of Step 122 - One Terminator Test

End of Section 12 -- Offline Transceiver Test
```

Error Messages

The following is a list of error messages which may appear when using LANDAD. The messages are listed in numerical order, and, where applicable, the probable cause and recommended action are part of the message. The wording of some messages is slightly different for HP-UX and MPE XL. Where this is the case, it is pointed out by enclosing the MPE XL version in square brackets. Only those errors which are defined by LANDAD are shown here; however, other error messages may also be displayed. For other errors, consult the DUI section of this manual and the operating system manuals.

99	***** THE MAXIMUM NUMBER OF ERROR MESSAGES HAS BEEN EXCEEDED. (LANDADERR 99)**
CAUSE	The maximum errors which were indicated in the run string by the user have been exceeded.
ACTION	Either increase the maximum errors parameter in the run string or let it default to infinity.

407	***** ERROR -- LENGTH OF RECEIVED FRAME <> LENGTH OF SENT FRAME. (LANDADERR 407)** ***** Length of actual frame sent . = !.** ***** Length of actual frame received = !.**
CAUSE	This could be caused by a number of problems: errors on the network, faulty software on the remote node, or faulty software on the local node.
ACTION	Retry the test. If the error still occurs, contact HP support representative.

920	***** ERROR -- REMOTE RESPONDED, BUT SSAP IS WRONG. (LANDADERR 920)** ***** Expected response SSAP = $01; received SSAP = !.** ***** Probably bound over another application.**
CAUSE	The remote node responded from the wrong SSAP.
ACTION	Reset the card, and retry the test.

921	***** ERROR -- REMOTE RESPONDED, BUT CONTROL FIELD IS WRONG. (LANDADERR 921)** ***** Expected control field = $F3; received control field = !.** ***** Probably bound over another application.**
CAUSE	The remote node responded with the wrong CONTROL field.
ACTION	Reset the card, and retry the test.

922	***** ERROR -- REMOTE RESPONDED, BUT LENGTH FIELD IS WRONG. (LANDADERR 922)** ***** Expected length field is one of ! or $0003.** ***** Received length field = !.**
CAUSE	Remote node responded to a TEST packet, but sent the wrong length data.
ACTION	Re-run the test again. If the same results are obtained, run diagnostics on the remote node. If the same results are obtained, contact your HP Support Representative.

923	***** ERROR -- REMOTE RESPONDED, BUT ACTUAL LENGTH <> LENGTH FIELD.** ***** (LANDADERR 923)** ***** Frame length field = !; frame actual length = !.**
CAUSE	Remote node responded to a TEST packet, but sent the packet size was incorrect.
ACTION	Re-run the test again. If the same results are obtained, run diagnostics on the remote node. If the same results are obtained, contact your HP Support Representative.

924	***** ERROR -- REMOTE RESPONDED, BUT SEQUENCE NUMBER WRONG.** **(LANDADERR 924)** ***** Expected sequence number = !; received sequence number = !.**
CAUSE	Remote node responded to a TEST packet, but sequence number was was incorrect.
ACTION	Re-run the test again. If the same results are obtained, run diagnostics on the remote node. If the same results are obtained, contact your HP Support Representative.

1020	***** ERROR -- REMOTE RESPONDED, BUT SSAP IS WRONG. (LANDADERR 1020)** ***** Expected response SSAP = !; received SSAP = !.** ***** Probably bound over another application.**
CAUSE	The remote node responded from the wrong SSAP.
ACTION	Reset the card, and retry the test.

1021	***** ERROR -- REMOTE RESPONDED, BUT FRAME IS TOO SHORT.** **(LANDADERR 1021)** ***** Expected frame length > 20 bytes; received frame length = !.**
CAUSE	Remote node responded to a TEST packet, but frame length was too short.
ACTION	Re-run the test again. If the same results are obtained, run diagnostics on the remote node. If the same results are obtained, contact your HP Support Representative.

1022	***** ERROR -- REMOTE RESPONDED, BUT LENGTH FIELD IS WRONG.** **(LANDADERR 1022)** ***** Expected length field = 6; received length field = !.**
CAUSE	Remote node responded to an XID packet, but the length field in the packet was incorrect.
ACTION	Re-run the test again. If the same results are obtained, then run diagnostics on the remote node. If the same results are obtained, contact your HP Support Representative.

1023	***** ERROR -- REMOTE RESPONDED, BUT CONTROL FIELD IS WRONG. (LANDADERR 1023)** ***** Expected control field = $BF; received control field = !.** ***** Probably logged over another application.**
CAUSE	Remote node responded to an XID packet, but the control field in the packet was incorrect.
ACTION	Re-run the test again. If the same results are obtained, then run diagnostics on the remote node. If the same results are obtained, contact your HP Support Representative.

1024	***** ERROR -- REMOTE RESPONDED, BUT FORMAT IDENTIFIER IS WRONG. (LANDADERR 1024)** ***** Expected format identifier = $81; received format identifier = !.**
CAUSE	Remote node responded to an XID packet, but the format identifier field in the packet was incorrect.
ACTION	Re-run the test again. If the same results are obtained, then run diagnostics on the remote node. If the same results are obtained, contact your HP Support Representative.

1025	***** ERROR -- REMOTE RESPONDED, BUT UNKNOWN CLASS OF SERVICE. (LANDADERR 1025)** ***** Expected either $01, $03, $05, or $07; received !.**
CAUSE	The remote node responded to an XID packet with an unknown class of service. This could be caused by data corruption on the network or by the remote node having implemented a new class of service unknown to this node.
ACTION	Retry the test. If the same results occur, contact your HP support representative.

5000	***** ERROR -- LANIC ALREADY IN USE BY DIAGNOSTIC SYSTEM. (LANDADERR 5000)** ***** Someone is already diagnosing the LANIC that you requested.** ***** It is illegal to have two copies of LANDAD diagnosing the same LANIC** ***** at the same time.**
CAUSE	LANIC already in use by diagnostic system.
ACTION	Wait until the other user is finished.

5001	***** ERROR -- LANIC DID NOT PASS SELFTEST. (LANDADERR 5001)** ***** The LANIC failed its internal selftest.** ***** Please replace the LANIC.**
CAUSE	Hardware Failure on LANIC.
ACTION	Replace the LANIC.

5002	***** ERROR -- LANIC DOES NOT RESPOND. (LANDADERR 5002)** ***** The problem may be one of the following:** ***** 1) The driver may not be configured properly.** ***** 2) The device is not a LANIC (LAN Interface Card).** ***** 3) The LANIC is completely inoperable.** ***** 4) There is no LANIC in the proper slot of the card cage.**
CAUSE	The problem may be one of the following: 1) The driver may not be configured properly. 2) The device is not a LANIC (LAN Interface Controller). 3) The LANIC is completely inoperable. 4) There is no LANIC in the proper slot of the card cage.
ACTION	1) Check driver configuration. 2) Verify that the device specified is a LANIC. 3) Verify that the LANIC is in the proper slot of the card cage. 4) If the above are correct, replace the LANIC. 5) If this also fails, contact your Hewlett-Packard Service Representative.

5004	***** INTERNAL ERROR -- BAD SOFTWARE STATUS RECEIVED FROM LAN_DAR.** ***** (LANDADERR 5004)**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5005	***** INTERNAL ERROR -- LAN_DAR INITIALIZATION FAILED.** **(LANDADERR 5005)**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5006	***** ERROR -- TRANSCEIVER POWER FUSE BLOWN ON LANIC. (LANDADERR 5006)** ***** Replace Fuse on the LANIC.** ***** Beware of these other possible problems:** ***** 1) Broken transceiver.** ***** 2) Shorted power lines in cabling to the transceiver** ***** (AUI cable or stub connector cable).** ***** 3) Broken LANIC.**
CAUSE	One of the following: 1) Broken transceiver. 2) Shorted power lines in cabling to the transceiver (AUI cable or stub connector cable). 3) Broken LANIC.
ACTION	Check for the following hardware problems: 1) Broken transceiver. 2) Shorted power lines in cabling to the transceiver (AUI cable or stub connector cable). 3) Broken LANIC.

5007	*** ERROR -- DATA COMPARE ERRORS. (LANDADERR 5007)
	$ Note: message 26 and message 27 usually follow this
	$ error message. These messages indicate what the data
	$ compare errors actually are and the total number of errors.

CAUSE	The cause could be one of the following: network corruption, software error on the local system, or software error on the remote system.
ACTION	Run selftest to attempt to isolate the problem.

5008	*** ERROR -- RETRY FAULT DURING TRANSMIT. (LANDADERR 5008) *** Time Domain Reflectometer (TDR) = !. *** Check the following: *** 1) Unterminated NETWORK CABLE; check terminators. *** 2) Open or shorted terminator; check terminator resistance. *** 3) transceiver not connected to NETWORK CABLE; check transceiver tee or tap. *** 4) NETWORK CABLE open or shorted. *** 5) Faulty transceiver.
CAUSE	Retry fault detected during data transmission.
ACTION	Be sure to check for the hardware error conditions in the order listed in the diagnostic message.

5009	*** ERROR -- LOSS OF CARRIER DETECTED DURING TRANSMIT. (LANDADERR 5009) *** Check the following: *** 1) AUI cable not connected to frontplane connector. *** 2) AUI cable not connected to transceiver. *** 3) Broken transceiver. *** 4) Broken AUI cable. *** 5) transceiver TAP shorted. *** 6) NETWORK CABLE shorted. *** 7) INT/EXT transceiver jumper block defective. *** 8) LANIC broken.
CAUSE	Loss of carrier experienced during transmission.
ACTION	Be sure to check for the hardware error conditions in the order listed in the diagnostic message.

5010	***** ERROR -- LOSS OF CARRIER DETECTED DURING TRANSMIT.** **(LANDADERR 5010)** ***** Check the following:** ***** 1) Stub cable not connected to frontplane connector.** ***** 2) AUI cable not connected to stub cable.** ***** 3) AUI cable not connected to transceiver.** ***** 4) Broken transceiver.** ***** 5) Broken AUI cable.** ***** 6) Broken stub connector cable.** ***** 7) Transceiver TAP shorted.** ***** 8) NETWORK CABLE shorted.** ***** 9) LANIC broken.**
CAUSE	Loss of carrier experienced during transmission.
ACTION	Be sure to check for the hardware error conditions in the order listed in the diagnostic message.

5011	***** ERROR -- LOSS OF CARRIER DETECTED DURING TRANSMIT.** **(LANDADERR 5011)** ***** Check the following:** ***** 1) Transceiver TEE shorted.** ***** 2) NETWORK CABLE shorted.** ***** 3) INT/EXT transceiver jumper block defective.** ***** 4) LANIC broken.**
CAUSE	Loss of carrier experienced during transmission.
ACTION	Be sure to check for the hardware error conditions in the order listed in the diagnostic message.

5012	***** ERROR -- LATE COLLISION DETECTED DURING TRANSMISSION.** **(LANDADERR 5012)**
CAUSE	This node was transmitting and detected another node on the network transmitting past the time when the other node should have detected that this node was transmitting.
ACTION	Check the following: 1) The transceiver connections are loose or broken. 2) The (Thicklan or Thinlan) network cable is not terminated at both ends. 3) The network cable is shorted. 4) The transceiver that the LAN/Console card is connected to has jabbed the LANIC card. Cycle power to the transceiver to reset it. 5) Some other node on the network is sending data onto the network cable either continually or at inappropriate times. If other nodes on the network exhibit this same failure, this is probably the cause. Run diagnostics on all nodes on the network.

5013	***** ERROR -- LANIC FAILED SELFTEST. (LANDADERR 5013)** ***** REPLACE LANIC.** ***** Internal Status:** ***** Interface Exception Status = !.** ***** Miscellaneous Status = !.** ***** Selftest/loopback Status = ! !.**
CAUSE	LANIC selftest failed.
ACTION	Replace the LANIC.

5014	***** ERROR -- LOOPBACK FRAME WAS TRANSMITTED SUCCESSFULLY** ***** BUT WAS NOT RECEIVED BEFORE THE TIMEOUT. (LANDADERR 5014)** ***** This problem is probably a software routing problem.**
CAUSE	This is an internal error, probably a software routing problem.
ACTION	Call your HP support representative.

5015	***** ERROR -- STATUS CALL FAILED. (LANDADERR 5015)**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5016	***** ERROR -- BIND CALL FAILED. (LANDADERR 5016)**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5017	***** ERROR -- UNBIND CALL FAILED. (LANDADERR 5017)**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5018	***** ERROR -- TRANSMIT CALL FAILED. (LANDADERR 5018)**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5019	***** ERROR -- RESET CALL FAILED. (LANDADERR 5019)**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5020	***** ERROR -- SELFTEST CALL FAILED. (LANDADERR 5020)**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5021	***** ERROR -- TRANSMIT ERROR, BUT CANNOT DETERMINE CAUSE.** (LANDADERR 5021)" ***** A transmission error has occurred which caused the transmitted frame** ***** not to reach the NETWORK CABLE. For some reason (probably "sticky"** ***** statistic counters that have reached their maximum count) the actual** ***** error can not be determined.**
CAUSE	Transmission error due to undetermined cause.
ACTION	Reset statistics and re-run the test again to determine cause.

5022	***** ERROR -- EXTERNAL LOOPBACK CALL FAILED. (LANDADERR 5022)**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5023	***** ERROR -- BABBLE ERROR DURING EXTERNAL LOOPBACK. (LANDADERR 5023)** ***** This error is caused by out of date hardware.** ***** Please replace LANIC.**
CAUSE	LANIC is out of date.
ACTION	Replace LANIC.

5024	***** ERROR -- FRAMING ERROR DURING EXTERNAL LOOPBACK. (LANDADERR 5024)**
CAUSE	Either a bad transceiver or a bad card.
ACTION	Replace the transceiver; if this does not fix the problem, replace the card.

5025	***** ERROR -- OVERFLOW ERROR DURING EXTERNAL LOOPBACK. (LANDADERR 5025)**
CAUSE	The LAN controller chip on the card has failed.
ACTION	Replace the LANIC.

5026	***** ERROR -- CRC ERROR DURING EXTERNAL LOOPBACK. (LANDADERR 5026)**
CAUSE	The transceiver has malfunctioned.
ACTION	Replace the transceiver.

5027	***** ERROR -- UNDERFLOW ERROR DURING EXTERNAL LOOPBACK. (LANDADERR 5027)**
CAUSE	The LAN controller chip on the card has failed.
ACTION	Replace the LANIC.

5028	***** ERROR -- INCORRECT DEVICE TYPE. (LANDADERR 5028)** ***** The following may be the cause:** ***** 1) The device specified is not a LANIC.** ***** 2) There is a software configuration problem.** ***** Evsn though the device is a LANIC, the system** ***** thinks that it is some other type of device.**
CAUSE	Either the LANIC has not be configured correctly, or the card in the slot specified is not a LANIC.
ACTION	Either reconfigure the LANIC or place a LANIC in the specified slot.

5029	***** ERROR -- EXTERNAL LOOPBACK FAILURE BUT CANNOT ISOLATE TO A CAUSE.** ***** (LANDADERR 5029)** ***** Check the following:** ***** 1) The transceiver is connected to the transceiver TAP.** ***** 2) The network cable is terminated at both ends.** ***** 3) The network cable is shorted.** ***** 4) The transceiver that the LANIC is connected to has jabbed the LANIC.** ***** Cycle power to the transceiver to reset it.** ***** 5) Broken cabling to the transceiver (AUI cable or stub connector cable).** ***** 6) Some other node on the network is babbling (sending data onto the** ***** network cable continually). If other nodes on the network exhibit** ***** this same failure, this is probably the cause.**
CAUSE	One of the following: 1) The transceiver is not connected to the transceiver TAP. 2) The network cable is not terminated at both ends. 3) The network cable is shorted. 4) The transceiver that the LANIC is connected to has jabbed the LANIC. Cycle power to the transceiver to reset it. 5) Broken cabling to the transceiver (AUI cable or stub connector cable). 6) Some other node on the network is babbling (sending data onto the network cable continually). If other nodes on the network exhibit this same failure, this is probably the cause.
ACTION	Check for all of the above causes in the order specified.

5030	*** ERROR -- POWERFAIL MESSAGE RECEIVED FROM DAM. (LANDADERR 5030) *** Please run diagnostic again. *** LANDAD Aborting.
CAUSE	Power has failed.
ACTION	Re-run LANDAD.

5031	*** ERROR -- RECEIVE CALL FAILED. (LANDADERR 5031)
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5032	*** ERROR -- RESET STATISTICS CALL FAILED. (LANDADERR 5032)
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5033	*** ERROR -- THE NOVRAM IC ON THE LANIC HAS FAILED. (LANDADERR 5033) *** Please replace either the NOVRAM or the LANIC. *** The NOVRAM is HP part number ! or equivalent. It is the *** socketed 18 pin Integrated Circuit (IC) on the LANIC. *** IMPORTANT NOTE: When you replace the NOVRAM or replace the LANIC, *** your station address will change.
CAUSE	The NOVRAM on the LANIC has failed.
ACTION	Replace the NOVRAM and run selftest.

5034	*** INTERNAL ERROR -- TRANSFER_COUNT WRONG ON GET_LAN_DA_STATUS CALL. *** (LANDADERR 5034) *** transfer_count = !; expecting between 96 and 480 inclusive.
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5035	*** INTERNAL ERROR -- TRANSFER_COUNT WRONG ON BIND_TO_DAM CALL. *** (LANDADERR 5035) *** transfer_count = !; expecting 12.
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5036	***** INTERNAL ERROR -- TRANSFER_COUNT WRONG ON TRANSMIT_FRAME CALL.** ***** (LANDADERR 5036)** ***** transfer_count = !; expecting !.**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5037	***** INTERNAL ERROR -- TRANSFER_COUNT WRONG ON RECEIVE_FRAME CALL.** ***** (LANDADERR 5037)** ***** transfer_count = !; expecting between 17 and 1514 inclusive.**
CAUSE	Internal software error.
ACTION	Contact your HP support representative.

5038	***** ERROR -- THE LANIC HAS ISSUED A PER EVENT AND TURNED ITSELF OFF.** ***** (LANDADERR 5038)** ***** The LANIC has sent an event to the driver indicating that a** ***** Protocol ERror (PER) has occurred. The actual ARQ status byte** ***** received by the driver was !.** ***** Please report this to your Hewlett-Packard Service Representative.** ***** Reset the LANIC.**
CAUSE	The LANIC firmware has detected a discrepency in the protocol with the driver.
ACTION	Reset the LANIC and try the test again. If the error still occurs, report the ARQ status byte to your Hewlett-Packard Service Representative.

5039	***** ERROR -- THE LANIC HAS ISSUED A DOD EVENT AND TURNED ITSELF OFF.** ***** (LANDADERR 5039)** ***** The LANIC has sent an event to the driver indicating that it** ***** was Dead or Dying (DOD). The actual ARQ status byte** ***** received by the driver was !.** ***** Please report this to your Hewlett-Packard Service Representative.** ***** Reset the LANIC.**
CAUSE	Hardware or firmware problem due to inconsistant state.
ACTION	Reset the LANIC and re-run the test. If the same results are obtained, replace the LANIC.

5040	***** ERROR -- THE CIO LANIC HAS ISSUED A DOD EVENT AND TURNED ITSELF OFF** ***** INDICATING THAT IT TRIED 128 CONSECUTIVE TIMES TO TRANSMIT** ***** A FRAME ONTO THE NETWORK MEDIA BUT COULD NOT BECAUSE IT WAS** ***** DEFERRING TO CARRIER. (LANDADERR 5040)** ***** Check that both terminators are connected to the** ***** LAN cable and run LANDAD Section 5 (Selftest).** ***** Also, one of the following may be at fault:** ***** 1) The transceiver that the LANIC is connected to has jabbed the LANIC.** ***** Cycle power to the transceiver to reset it.** ***** 2) Some other transceiver on the network is jabbering.** ***** 3) Broken cabling to the transceiver (AUI cable or stub connector cable).** ***** 4) There is a DC voltage level on the LAN cable that is causing** ***** the transceiver to assert its collision signal.** ***** 5) The LANIC is broken.**
CAUSE	One of the hardware malfunctions listed in the error message has occurred.
ACTION	Check for all of the possible hardware malfunctions listed in the error message. If the failure is still not found, contract your HP representative.

5041	***** ERROR -- THE DAM REPORTS LOSS OF COMMUNICATION WITH LANIC.** ***** (LANDADERR 5041)** ***** The actual ARQ status byte reported by the DAM was !.** ***** If this byte is $00 or $10, this indicates an I/O system error.** ***** Any other value indicates a LANIC error.** ***** Please report the state of the LEDs on the LANIC card and the** ***** ARQ status value to your Hewlett-Packard Service Representative.**
CAUSE	I/O System error.
ACTION	Please report the state of the LEDs on the LANIC card and the ARQ status value to your Hewlett-Packard Service Representative.

5042	***** ERROR -- THE LANIC INDICATES THAT INFINITE DEFERRAL HAS *** BEEN DETECTED. (LANDADERR 5042) *** Check that both terminators are connected to the *** LAN cable and run LANDAD Section 5 (Selftest). *** Otherwise, one of the following may be at fault: *** 1) The transceivsr that the LANIC is connected to has jabbed the LANIC. *** Cycle power to the transceiver to reset it. *** 2) Some other transceiver on the network is jabbering. *** 3) Broken cabling to the transceiver (AUI cable or stub connector cable). *** 4) There is a DC voltage level on the LAN cable that is causing *** the transceiver to assert its collision signal. *** 5) If the transceiver is not an HP transceiver, both terminators are not connected *** to the LAN cable. *** 6) The LANIC is broken.**
CAUSE	One of the hardware malfunctions listed in the error message has occurred.
ACTION	Check for all of the hardware malfunctions in the error message, fix the problem, and re-run selftest.

5043	***** INSUFFICIENT SECURITY LEVEL. (LANDADERR 5043)**
CAUSE	Your user security level does not give you the ability to run destructive tests.
ACTION	Contact your system administrator the give you the ability to run destructive tests, or only run non-destructive tests.

5044	***** INT/EXT TRANSCEIVER JUMPER IS MISSING OR MISALIGNED (LANDADERR 5044) *** The jumper that determines if you are using the internal Thin transceiver *** or external AUI connector is either missing or misaligned.**
CAUSE	The transceiver jumper is missing or is misaligned.
ACTION	Determine whether you are using an internal or an external transceiver and set the jumper accordingly.

5045	***** ERROR -- THE LANIC HAS ISSUED A STF EVENT AND TURNED ITSELF OFF. *** (LANDADERR 5045) *** The LANIC has sent an event to the driver indicating that its *** Selftest Failed (STF). The actual ARQ status byte *** received by the driver was !. *** Please report this to your Hewlett-Packard Service Representative. *** Please replace the LANIC.**
CAUSE	LANIC selftest failed.
ACTION	Report the ARQ status byte to your Hewlett-Packard Service Representative and replace the LANIC.

5046	***** ERROR -- THE LANIC HAS ISSUED A WTF EVENT AND GONE OFFLINE.** ***** (LANDADERR 5046)** ***** The LANIC has sent an event to the driver indicating that a** ***** Warning Test Failed (WTF) occurred. The actual ARQ status byte** ***** received by the driver was !.**
CAUSE	An error has occurred which affects some portion of the LANIC. This message is usually followed by a message to replace a specific part.
ACTION	Replace the part as directed in the message following this message. If there is no additional message, run Selftest to determine the failure.

5047	***** ERROR -- CREATE_MANAGER CALL FAILED. (LANDADERR 5047)**
CAUSE	This can occur on an MPE/XL system only. The Device Access Routine could not create the driver for LANDAD, reason unknown.
ACTION	Reboot the system and try the test again. If the same results occur, contact your HP Service Representative.

5048	***** ERROR -- DELETE_MANAGER CALL FAILED. (LANDADERR 5048)**
CAUSE	This can occur on an MPE/XL system only. The Device Access Routine could not delete the driver for LANDAD, reason unknown.
ACTION	Reboot the system and try the test again. If the same results occur, contact your HP Service Representative.

6000	***** WARNING -- DESTRUCTIVE SECTIONS CAN NOT BE RUN FROM YOUR TERMINAL.** ***** DESTRUCTIVE SECTIONS MAY ONLY BE RUN FROM A TERMINAL THAT IS NOT CONNECTED** ***** THROUGH THE LANIC TO BE DIAGNOSED.** ***** THE FOLLOWING SECTIONS CAN BE PERFORMED FROM YOUR TERMINAL:** **1,3,4,6,7,9,10.** ***** NO OTHER SECTIONS CAN BE SPECIFIED. (LANDADWARN 6000)**
CAUSE	Wrong terminal type.
ACTION	Change to a terminal which is directly connected to the LANIC or do not run destructive tests.

6002

```
*** WARNING -- NO SQE HEARTBEAT DETECTED WHEN USING
*** "SQE" JUMPER CONFIGURATION. (LANDADWARN 6002)
*** The No SQE Heartbeat condition was detected.
*** Run Section 7 Step 72 (Reset Link Statistics).  Then
 run
*** Section 4 (Local Loopback) with LOOPCOUNT=10.  Next run
 Section 7
*** (Link Statistics).  If six or more of the transmit attempts
 have caused
*** ths "No heartbeat detected after transmission" statistic to
 increment,
*** a hardware fault exists.
*** Possible sources are:
*** 1) If you are using an HP28664A Starlan10 transceiver, make certain
***    that the SQE enable switch in in the ENABLED position.
*** 2) An Ethernet 1.0 transceiver is connected.  This is an illegal
***    configuration.  Set the SQE option to "no SQE".
*** 3) The transceiver is broken.
*** 4) The AUI cable is broken.
*** 5) The stub connector cable is broken.
*** 6) The LANIC is broken.
```

CAUSE Possible sources are: 1) If you are using an HP28664A Starlan10 transceiver, make certain that the SQE enable switch in in the ENABLED position. 2) An Ethernet 1.0 transceiver is connected. This is an illegal configuration. Set the SQE option to "no SQE". 3) The transceiver is broken. 4) The AUI cable is broken. 5) The stub connector cable is broken. 6) The LANIC is broken.

ACTION Run Section 7 Step 72 (Reset Link Statistics). Then run Section 4 (Local Loopback) with LOOPCOUNT=10. Next run Section 7 (Link Statistics). If six or more of the transmit attempts have caused the "No heartbeat detected after transmission" statistic to increment, a hardware fault exists. Check the following:

1) If you are using an HP28664A Starlan10 transceiver, make certain that the SQE enable switch in in the ENABLED position. 2) An Ethernet 1.0 transceiver is connected. This is an illegal configuration. Set the SQE option to "no SQE". an IEEE 802.3 compatible transceiver (HP30241A,HP28641A or HP28664A). 3) Swap the transceiver with a transceiver which is known to be good. Re-run the test. If the results are correct, then replace transceiver. 4) Swap the AUI cable with an AUI cable which is known to be good. Re-run the test. If the results are correct, then replace AUI cable. 5) Swap the stub connector cable with a stub connector cable which is known to be good. Re-run the test. If the results are correct, then replace stub connector cable. 6) If all other actions fail, the LANIC is broken. Replace the LANIC.

6003	*** WARNING -- LANDAD CAN ONLY TEST HP TRANSCEIVERS. (LANDADWARN 6003) *** The diagnostic does not know how non HP transceivers react to stimuli *** and results may not be as expected. If you really want to run *** this section on a non HP transceiver, answer 1 to the question *** and proceed at your own risk.
CAUSE	The user input OTHER type transceiver. This indicates to the diagnostic that the LANIC is attached to a transceiver which has not been tested with the diagnostic.
ACTION	If you wish to proceed, answer 1 to the question. If you wish to indicate another transceiver type, then input the correct type.

6004	*** WARNING -- DIAGNOSTIC SSAP IN USE BY ANOTHER PROCESS. (LANDADWARN 6004) *** Another process has bound to the diagnostic SSAP. *** This section can only be run after the other process finishes.
CAUSE	You have exhausted all tries to obtain a SSAP.
ACTION	Try again. If it still fails, reboot the system.

6005	*** WARNING -- LANIC IS OFFLINE. (LANDADWARN 6005) *** The LANIC must be online to perform this test. *** This indicates that you may have a bad LANIC. *** Please run section 5 (selftest) to determine if your *** LANIC is defective. If selftest passes, the LANIC *** will be returned to the online state.
CAUSE	LANIC is offline. LANIC may be defective.
ACTION	This indicates that you may have a bad LANIC. Please run section 5 (selftest) to determine if your LANIC is defective. If selftest passes, the LANIC will be returned to the online state. If selftest fails, replace the LANIC.

6006	*** WARNING -- SSAP ! ALREADY IN USE BY ANOTHER PROCESS. (LANDADWARN 6006)
CAUSE	LANDAD has probably been previously aborted and the SSAP specified has not been unbound.
ACTION	Enter another SSAP.

6007	*** WARNING -- NO VALID SECTIONS WERE SPECIFIED. LANDADWARN 6007) *** Valid sections are 1-12 inclusive.
CAUSE	An invalid section was specified to be run.
ACTION	Input a valid section number.

6008	***** WARNING -- DO NOT HIT THE BREAK KEY.** **(LANDADWARN 6008)** ***** This instance of the driver was created by LANDAD.** ***** If you hit the BREAK key and abort LANDAD, system** ***** resources will not be recovered and you will have** ***** to reboot the system in order to use networking.**
CAUSE	This warning message is seen on MPE/XL only. It is a temporary solution to a driver problem which will be fixed in MPE/XL Version 4.0.
ACTION	Do not hit the BREAK key and select the ABORT option or you will have to reboot the system to use networking.

6009	***** WARNING -- ERROR IN IO PATH.** **(LANDADWARN 6009)** ***** Any subsequent errors may be associated with** ***** a malfunction in a hardware module which is in** ***** the path of the LANIC being tested.**
CAUSE	Something in the path between LANDAD and the LANIC is not present or is malfunctioning.
ACTION	Verify that you have the LANIC in the correct slot of the card cage and that you are using the correct parameters in the RUN command line.

Contents

5

SCSI CD Diagnostic

Introduction

The SCSI Compact Disk Diagnostic (SCSICD) is designed to provide a means of testing SCSI compact disks on any system that supports the online diagnostics subsystem. The disks that will be tested using this diagnostic are those that utilize the Small Computer System Interface (SCSI) message protocol for communication with the SPU. Tests provided can:

- Perform an extensive fault isolating diagnostic trouble tree on the disk and it's system interface. Defects encountered will be relayed to the user.
- Verify the integrity of the SCSI data path to the selected disk.
- Identify the product type of the selected disk.
- Perform the internal selftest on the disk.
- Obtain and decode status messages from the disk.
- Test common transactions that are used in communication between the disk and CPU to determine the integrity of these transactions.

In addition, an interactive external exerciser will be provided. This exerciser provides access to internal disk diagnostics and utilities.

Defects and Enhancements

Submit defect reports and enhancement requests for this diagnostic through the STARS database referencing Product Number 30600-10044.

Minimum Configuration

The hardware running beneath SCSICD is of no concern to the diagnostic, as long as the online diagnostics subsystem is supported and there is at least one SCSI compact disk configured on the system.

In order to run this diagnostic, the online diagnostics subsystem must be supported.

Autodiagnostics

If SCSICD is run as an autodiagnostic by the system, the default section will be run.

Operating Instructions

The SCSI Compact Disk Diagnostic is part of the total online diagnostics subsystem. It is designed to provide the user with both an on-line and off-line means of thoroughly testing any SCSI compact disk on the system. With the on-line version, the functionality of system disks is limited to Non-Exclusive/Non-Destructive commands.

SCSICD can be run in two different modes as described below:

- Non-Exclusive/Non-Destructive - indicates that the user can only run tests on the device that are non-destructive to data on that device as well as other commands that may be executing. An example of a test that would not be allowed in this mode is the Device Reset.
- Exclusive/Destructive - indicates that the user may run any test on the device. There are virtually no restrictions placed on the user in this mode and, therefore, extreme caution should be exercised by anyone running tests in Exclusive/Destructive Mode.

Default Tests

If the user does not specify any sections the following default section will be executed:

Section 10 Fault isolating diagnostic trouble tree

RUN Command

The SCSI compact disk diagnostic can be accessed by the user via the Diagnostic User Interface. It is initiated using the RUN SCSICD command. All parameters available in the run command are acceptable as parameters when running this diagnostic. Note that if the ERRONLY parameter is set "on", only error messages will be output by this diagnostic. Error messages can be distinguished from other messages by three "*"'s preceding the text of the message (i.e., '*** MESSAGE' is an error message whereas 'Message' is not). Also note that error messages are in all capital letters and other messages use some lower case. This diagnostic can also be run as an auto-diagnostic by the diagnostic system. As noted in the previous section, certain modes are required in order to do certain tests. The mode needed to execute each command will be noted in the description of that command.

Test Execution

When SCSICD is run, the following header and welcome message will be displayed:

```
**********************************************************************
******                                                          ******
******                    SCSI CD DIAGNOSTIC                    ******
******                                                          ******
******         (C) Copyright Hewlett Packard Co.  1990          ******
******                   All Rights Reserved                    ******
******                     Version  n.nn.nn                     ******
******                                                          ******
**********************************************************************

Welcome, Today is FRI, Aug 5, 1982  7:39 AM
```

At this point, the diagnostic calls IO_Path_Test which is a diagnostic procedure that tests the I/O path to the device. If the status returned from this procedure call is "fail", the following message will be output:

```
*** WARNING -- THE I/O PATH TO THE DISK MAY NOT BE FUNCTIONING
    PROPERLY (SCSIERR 100)
```

Otherwise,the diagnostic issues an inquiry to the specified device to determine whether or not it is a SCSI compact disk. If the device does not respond to the inquiry command, the following message will be output:

```
*** DEVICE FAILED TO RESPOND TO INQUIRY COMMAND IN ALLOTTED TIME
(SCSIERR 101)
```

A second inquiry command is then issued. If this command fails, the following messages are displayed:

```
*** DEVICE FAILED TO RESPOND TO INQUIRY COMMAND IN ALLOTTED TIME
(SCSIERR 101)

*** SCSI INQUIRY COMMAND FAILED.  DEVICE MAY NOT BE SCSI OR INQUIRY
INFORMATION RETURNED MAY BE ERRONEOUS.  SCSICD MAY NOT FUNCTION
PROPERLY (SCSIERR 207)

Do you with to continue (Y/N)[N]:
```

If the user answers no the diagnostic will terminate immediately.

If a response was obtained, then the returned status is examined to determine if the device is a SCSI disk. If not, the following message will be displayed:

```
*** SCSICD IS UNABLE TO DIAGNOSE THE SELECTED DEVICE (SCSIERR 102)
```

If this message is generated, the diagnostic will terminate immediately after outputting it.

At this point, the sections specified by the user will be executed and the results output. If the user did not specify any sections the default section will be executed. If at any time, the number of errors generated reaches the limit specified by the user in the ERRCOUNT parameter of the run command, the following message will be output:

```
*** THE MAXIMUM NUMBER OF ERROR MESSAGES HAS BEEN EXCEEDED (SCSIERR 110)
```

The diagnostic will then terminate execution. If the ERRPAUSE parameter of the RUN command was assigned a value of "on", then this diagnostic will stop after each error is generated and ask the user if the test should continue:

```
Do you wish to continue (Y/N)[Y]?
```

If the response is `Y` then the test will be resumed (if possible), and if the response is `N` this diagnostic will terminate. the `Y]` indicates that `Y` will be the default response if the user simply hits <CR> in response to the prompt. If the sections specified by the user were executed the number of times specified in the LOOP parameter of the run command without the number of errors exceeding the ERRNUM value, the diagnostic will terminate normally and the following message output:

```
SCSI CD Diagnostic Exiting . . .
```

Upon termination of this diagnostic, control will return to the diagnostic subsystem.

Test Section Descriptions

The remainder of this document is devoted to explaining each section in SCSICD. For each section, this explanation will consist of a description of the section, including the actions performed therein, the expected output from that section, and any error messages that may be generated that are worth noting. Please note that in regard to the error messages, all possible error messages that may be generated are not listed. The only error messages that are listed are those that are considered to be of special significance.

Section 10—DIAGNOSTIC TROUBLE TREE

This section will execute the fault isolating diagnostic trouble tree. The algorithm follows:

1.) Issue an internal power-on SELFTEST to the device.

2.) Read block twice and compare the data.

OUTPUT :

```
Section 10 -- Diagnostic Trouble Tree

Device Selftest Completed

Read Block Completed

End of Section 10 -- Diagnostic Trouble Tree
```

POSSIBLE ERROR MESSAGES :

```
*** DEVICE ENCOUNTERED AN ERROR WHILE EXECUTING THE
    SCSI INITIATE DIAGNOSTIC COMMAND (SCSIERR 109)

    SENSE = { sense }

{Note: -- This sense printout will include the failing field
          replaceable unit(s) as specified by the device}
```

Section 17—EXTERNAL EXERCISER

The SCSI External Exerciser is an interactive program which provides the user with access to the set of internal diagnostics and utilities within a SCSI compact disk. The purpose of the exerciser is to aid service-trained personnel in troubleshooting SCSI compact disks to a field replaceable unit level.

The External Exerciser, as discussed earlier, is an interactive program that provides the user with access to the set of internal diagnostics and utilities within a SCSI compact disk.

Media Testing

Media tests are powerful tools used to determine media integrity within a SCSI device. These tests can find recoverable and unrecoverable read errors and provide information concerning each error, such as the address where the error occurred and the type of error.

All media tests allow the user to input a loop count when requesting the test. Each time an error is detected during the test, the test will stop, report the error, and then resume testing until the loop count has been satisfied. Note that the loop count is not a count of the number of errors, but rather the number of passes the device will execute during the media test. The following information will be reported for each error that occurs:

- The current logical address.
- The error type.
- The loop count when the error occurred.

There is only one type of media test that can be performed on compact disk devices, the read only media test. The type of test, is a non-destructive test which reads data from the disk and attempts to detect any read errors that occur. There are two tests available in this category:

1.) Selected Area Read Only Media Test—Sequentially reads the current data on the disk in a specified area of the media.
2.) Random Read Only Media Test—Reads random sectors of random length data. This allows read errors to be detected on a large portion of the media in a minimum amount of time.

When the external exerciser is invoked, the following prompt will be displayed to the user:

```
SCSICD>
```

When the prompt appears, the exerciser is waiting for a command from the user. The available commands are listed in this section in alphabetical order accompanied by descriptions. Each command description is in the following format:

COMMAND NAME

SHORT DEFINITION

Explanation of what the command does and when it should be used.

INPUT FORMAT:

```
SCSICD> [COMMAND NAME]
```

Note that the prompt for this exerciser is SCSICD>. The appearance of this prompt indicates that the exerciser is waiting for the user to input a command.

OUTPUT FORMAT:

Information printed as a result of this command being executed

POSSIBLE ERROR MESSAGES:

Any error messages worth noting that may be generated as a result of this command. Note that this does not imply that all possible error messages that may be generated are listed here, but rather, only those that are especially worth mentioning in relation to the command.

There are several conventions used throughout this section in the command formats. They are as follows:

nnnn	- refers to a decimal number of any magnitude that is output by the diagnostic.
vvvv	- refers to a decimal number of variable magnitude that must be input by the user.
xxxx	- refers to text displayed or output by the program
H	- refers to a hexadecimal digit (0-F).
O	- refers to an octal digit (0-8).
B	- refers to a binary digit (0-1).

Any text enclosed in parenthesis indicates that the user is expected to input that text in response to a query from the diagnostic. Text enclosed in square brackets indicates the default response that will be assumed if the user simply hits <CR> in response to the query. For example, the user may be asked if the program should continue by the following question:

```
Do you wish to continue (Y/N)[N]?
```

The text enclosed in the parenthesis (i.e., **Y / N**) indicates that the user is to type either a **Y** or an **N** in response to the question. Each option is separated by a **/**. The default response in this case would be **N**, as indicated by the **[N]** prior to the question mark. If there is no default response listed (enclosed in square brackets), the user will not be allowed to "default" the input by only entering a <CR>.

Comments in the input and output sequences are enclosed in curly brackets and are not output by the program. For example, following the question in the previous example, the input sequence would probably look something like the following:

```
{if response was N this command will terminate}
```

Error messages will be preceded by three asterisks ("***") and will be in all upper case letters. Other messages will not be all upper case and will not be preceded by the asterisks.

If the user enters a command in response to a prompt and that command is not recognized as an exerciser command, the following message will be displayed:

```
*** YOUR RESPONSE WAS INVALID

SCSICD>
```

This message simply means that the command entered is not part of the command set for the external exerciser.

When the user enters a command that is recognized by the external exerciser, that command will then be processed according to the corresponding command description given in one of the following sections of this document. If an error is encountered as a result of issuing a SCSI command to the device, the hardware sense that is returned by the device will be displayed to the user. The status display will consist of the following format:

```
SENSE =

HH  HH  HH  HH
HH  HH  HH  HH
HH  HH  HH  HH
HH  HH  HH  HH
HH  HH

Sense Key = HH
Sense Code = HH

{One or more of the following sense fields may be printed}

Address of event:
   Block address = nnnn

Failed field replaceable unit: nnnn
```

Note that only the portions of the above status display that correspond to errors indicated by the hardware status variable will be output. This means, for example, that if no Drive Errors were indicated by the status, none would be output.

In order to exit the exerciser, the EXIT command should be entered (see EXIT command description).

CAPACITY

Non-Exclusive/Non-Destructive Command

This command allows the user to determine the maximum capacity, block address of the last addressable block and the current block size.

INPUT FORMAT:

```
SCSICD> CAPACITY
```

OUTPUT FORMAT:

```
CAPACITY UTILITY
PDEV XXXX is an HPXXXX disk drive

   Capacity Information
   ====================
   Max Block Address = nnnn
   Current Block Size = nnnn
   Drive Capacity = nnnn bytes

CAPACITY UTILITY COMPLETED
```

DEVICE RESET

Exclusive/Non-Destructive Command

This command is used to reset the device. The following will be performed as a result of this command:

- Finish any logical block write in progress
- Abort any command in progress
- Controller initialization
- Initialize spare table
- Initialize saved pages information
- Initialize logs

INPUT FORMAT:

```
SCSICD> DEVICE RESET
```

OUTPUT FORMAT:

```
DEVICE RESET UTILITY
PDEV XXXX is an HPXXXX disk drive

DEVICE RESET UTILITY COMPLETED
```

DIAG

Non-Exclusive/Non-Destructive Command

This command will initiate the internal power-on self-test. A loop option allows the diagnostic to be repeated a specified number of times. Tests are device dependent, and are full described in the support documentation for each drive.

INPUT FORMAT:

```
SCSICD> DIAG

   * * * * * * * * * * * * * * * * * * * * * * * * *
   *                  CAUTION                      *
   *   This command MAY tie up the system for      *
   *   as long as it takes to finish the test.     *
   * * * * * * * * * * * * * * * * * * * * * * * * *

   Do you wish to continue (Y/N)[Y]?

   Input the loop count (nnnn <= count <= nnnn)[nnnn]?
```

OUTPUT FORMAT:

```
   INITIATE DIAGNOSTIC UTILITY
   PDEV XXXX is an HPXXXX disk drive


   INITIATE DIAGNOSTIC UTILITY COMPLETED
```

POSSIBLE ERROR MESSAGES:

```
*** DEVICE ENCOUNTERED AN ERROR WHILE EXECUTING THE
    SCSI INITIATE DIAGNOSTIC COMMAND

    SENSE = { sense }
```

EXIT

Non-Exclusive/Non-Destructive Command

This command terminates execution of the External Exerciser. It may be entered any time the SCSICD> prompt appears.

INPUT FORMAT:

```
SCSICD> EXIT
```

OUTPUT FORMAT:

```
End of Section 17 - External Exerciser
```

HELP

Non-Exclusive/Non-Destructive Command

This command provides the user with access to information concerning the commands that are available in the external exerciser. The user may request a list of the available commands accompanied by a brief description of each, or individual command descriptions. For individual commands, the user can request a description and syntax or just the syntax.

INPUT FORMAT:

```
SCSICD> HELP [command name or <cr>]
```

OUTPUT FORMAT:

```
{If no command name was given (i.e. <cr>)}
   The following commands are available:


   CAPACITY     - Displays the maximum capacity and block size.
   DEVICE RESET - Issue a Bus Device Reset
      .
      .
      .
   SUSPEND      - Suspends SCSICD and returns to the DUI

{If a command was given}
   Do you want a description or just syntax (D/S)[D]?

{If response was D}
   COMMAND DESCRIPTION:
   Description of the command
   Syntax of the command

{If response was S}
   COMMAND SYNTAX:
   Syntax of the command
```

INQUIRY

Non-Exclusive/Non-Destructive Command

This command allows the user to obtain the drive type, media type and firmware revision.

INPUT FORMAT:

```
SCSICD> INQUIRY
```

OUTPUT FORMAT:

```
INQUIRY UTILITY
PDEV XXXX is an HPXXXX disk drive

Inquiry Information
===================

  Product Revision    = xxxx
  Firmware Revision   = xxxx

INQUIRY UTILITY COMPLETED
```

LDEV

Non-Exclusive/Non-Destructive Command

The LDEV command allows the user to select a new device to be tested. The user can select the new device by LDEV or PDEV.

INPUT FORMAT:

```
CS80DIAG> LDEV
```

OUTPUT FORMAT:

```
   LDEV UTILITY
   PDEV XXXX is an HPXXXX disk drive

   Input LDEV or PDEV (L/P)[P]?

{If response was L}
   New LDEV>

{If response was P}
   New PDEV>

   PDEV XXXX is an HPXXXX disk drive

   LDEV UTILITY COMPLETED
```

READ

Non-Exclusive/Non-Destructive Command

This command allows the user to access any data block on the selected device. Due to potential security compromise, the user will need to possess Level 0 security to use this command.

INPUT FORMAT:

```
SCSICD> READ

   Enter new address (<cr> to keep current value) --
      Block address (nnnn - nnnn) = nnnn?
```

OUTPUT FORMAT:

```
   READ UTILITY
   PDEV XXXX is an HPXXXX disk drive

   The data in hex follows:

         0   1   2   3   4   5   6   7   8   9

     0:  HH  HH  HH  HH  HH  HH  HH  HH  HH  HH   . . . . . .
    10:  HH  HH  HH  HH  HH  HH  HH  HH  HH  HH   . . . . . .
         .
         .
         .
  2040:  HH  HH  HH  HH  HH  HH  HH  HH          . . . . . .

{Note: -- the dots in the preceding table represent non-alpha-numeric
 ASCII characters. Alpha-numeric characters will be printed}

   READ UTILITY COMPLETED
```

RO MT

Non-Exclusive/Non-Destructive Command

This command is used to initiate a read only media test. Two types of tests are available through this command. The first read only media test allows the user to specify the address at which the test is to start. This test will sequentially read data starting at this address in an attempt to detect any read errors. The second type of test is a random read only media test which uses random addresses and lengths of reads in attempt to detect any read errors.

INPUT FORMAT:

```
SCSICD> RO MT

* * * * * * * * * * * * * * * * * * * * * * * * *
*                  CAUTION                        *
*   This command MAY tie up the system for        *
*   as long as it takes to finish the test.       *
* * * * * * * * * * * * * * * * * * * * * * * * *

Do you wish to continue (Y/N)[Y]?

Clear ALL logs (Y/N)[N]?

Types of RO MT's:
  S = selected area
  R = random area
Enter the test type (S/R)[S]?

Enter new address (<cr> to keep current value) --
   Block address (nnnn - nnnn) = nnnn?

Test Area:
  V = volume
  S = sector
Enter the test area (V/S)[S]?

Input the loop count (nnnn<= count <=nnnn)[nnnn]?
```

OUTPUT FORMAT:

```
          RO MT UTILITY
          PDEV XXXX is an HPXXXX disk drive

          Test Area = Volume starting at Cyl = nnnn, Head = nnnn, Sec = nnnn
                              {or}
          Test Area = Sec starting at Cyl = nnnn, Head = nnnn, Sec = nnnn

       {If no errors detected}
          No errors were detected by the media test
                    {or}

       {If errors were detected}
       *** ERROR IN TRANSMISSION DETECTED DURING READ
           MEDIA TEST: (SCSIERR 124)

                                    Bit
                          Hex   Hex  Positions
                          Value Value In Error
 Cyl   Hd  Sect Byte      Trns  Recd 01234567      Time Error Occurred
==================== ==== ==== ========  ==========================
nnnn  nnnn nnnn nnnn      HH    HH   BBBBBBBB   FRI, DEC 16, 1988  9:03 PM
nnnn  nnnn nnnn nnnn      HH    HH   BBBBBBBB   FRI, DEC 16, 1988  9:03 PM
.
.  Repeat for each occurrence
.
nnnn  nnnn nnnn nnnn      HH    HH   BBBBBBBB   FRI, DEC 16, 1988  9:03 PM

          RO MT UTILITY COMPLETED
```

SEEK

Non-Exclusive/Non-Destructive Command

This command causes the drive to seek to a specified address or series of addresses. The command is useful for testing the servo circuitry. Three types of seeks may be performed: Random Seek seeks to a random address; Alternate Seek seeks between two specified addresses; and Butterfly Seek performs a series of seeks over the entire disk surface.

INPUT FORMAT:

```
SCSICD> SEEK

   * * * * * * * * * * * * * * * * * * * * * * * * *
   *                  CAUTION                      *
   *   This command MAY tie up the system for      *
   *   as long as it takes to finish the test.     *
   * * * * * * * * * * * * * * * * * * * * * * * * *

   Do you wish to continue (Y/N)[Y]?

   Types of seeks:
     A = alternate seek
     B = butterfly seek
     R = random seek
   Select the seek type (A/B/R)[R]?

{If response was Alternate}
   Input Address #1:

   Enter new address (<cr> to keep current value) --
      Block address (nnnn - nnnn) = nnnn?

   Input Address #2:

   Enter new address (<cr> to keep current value) --
      Block address (nnnn - nnnn) = nnnn?

{If response was Butterfly}
   Test all heads (Y/N)[N]?
{If response was N}
   Input head number (nnnn <= head <= nnnn)[nnnn] ?

   Input the loop count (nnnn <= count <= nnnn)[nnnn]?^
```

OUTPUT FORMAT:

```
SERVO TEST UTILITY
PDEV XXXX is an HPXXXX disk drive

Pass nnnn started
Pass nnnn completed

SERVO TEST UTILITY COMPLETED
```

SUSPEND

Non-Exclusive/Non-Destructive Command

The SUSPEND command allows the user to suspend SCSICD and return to the DUI. The user enters **RESUME** at the DUI prompt to return to SCSICD.

INPUT FORMAT:

```
SCSICD> SUSPEND
```

OUTPUT FORMAT:

```
DUI>
```

Error Messages

This section gives a complete list of the error messages that may be generated by SCSICD along with brief explanations of the meaning of the messages. The messages will be listed in numerical order and are exactly as they appear in the message catalog. Thus, a "!" indicates that a parameter of some sort will be placed in the location marked by the exclamation point. A "&" indicates that the line below will be concatenated onto the end of the line with the ampersand. Finally, a "%" means that the line below is a continuation of the message, but not to be concatenated as with the ampersand.

100	***** WARNING -- THE I/O PATH MAY NOT BE FUNCTIONING PROPERLY (SCSIERR 100)**
CAUSE	An error was detected by the Io_Path_Test service while testing the modules on the i/o path preceding the selected device.
ACTION	Execute the appropriate diagnostics on the modules preceding the selected device on the i/o path, especially on those that may have been reported as faulty in error messages immediately preceding this message. Note that the results of the execution of this instance of SCSICD may be invalid.

101	***** DEVICE FAILED TO RESPOND TO ! COMMAND IN ALLOTTED TIME (SCSIERR 101)**
CAUSE	No response to an i/o was received prior to the expiration of the allotted time.
ACTION	Verify that the selected disk drive is actually connected to the system. Run SYSMAP, if available, to confirm the presence of the device.

102	***** SCSICD IS UNABLE TO DIAGNOSE THE SELECTED DEVICE (SCSIERR 102)**
CAUSE	The selected device identified itself as something other than a SCSI drive.
ACTION	Determine type of selected device and run the appropriate diagnostic on it.

103	***** SCSI CD DIAGNOSTIC TERMINATING (SCSIERR 103)**
CAUSE	A fatal error has been encountered.
ACTION	The specific error that was encountered should have been reported immediately prior to this message. Follow the action instructions for that error message.

104	***** A RESERVE AREA IS NOT AVAILABLE ON THIS DRIVE -- NO OPERATION WAS PERFORMED. (SCSIWARN 104)**
CAUSE	There is no available area on the disk that can be written to without corrupting user data. This can happen if no such area is provided by the system, or if the diagnostic is running in destructive mode, in which the location of the reserve area is unknown to the system since the drive is locked for diagnostics.
ACTION	For systems which do not support a reserve area (e.g. HP-UX), reserve area operations are not supported.

105	***** EXCLUSIVE MODE REQUIRED TO EXECUTE THIS COMMAND (SCSIERR 105)**
CAUSE	The diagnostic does not have access to the drive in the mode necessary to execute the selected command.
ACTION	The mode granted to a diagnostic for a selected device is system dependent. Use the HELP LDEV command at the DUI for more information.

106	***** DESTRUCTIVE MODE REQUIRED TO EXECUTE THIS COMMAND (SCSIERR 106)**
CAUSE	The diagnostic does not have access to the drive in the mode necessary to execute the selected command.
ACTION	The mode granted to a diagnostic for a selected device is system dependent. Use the HELP LDEV command at the DUI for more information.

107	***** INVALID DEVICE WAS SELECTED (SCSIWARN 107)**
CAUSE	The device selected with the LDEV command is not a valid device.
ACTION	Verify the PDEV or LDEV input and retry the command.

108	***** ! COMMAND IS NOT IMPLEMENTED ON THIS DRIVE/SYSTEM (SCSIERR 108)**
CAUSE	The selected operation is either not implemented on the selected drive or the system does not provide access to it.
ACTION	This operation is unavailable.

109	***** DEVICE ENCOUNTERED AN ERROR WHILE EXECUTING THE SCSI ! COMMAND (SCSIERR 109)**
CAUSE	The drive reported an error as a result of executing the selected operation.
ACTION	The hardware status that is displayed immediately following this message should indicate what sort of problem occurred, including a failing FRU if one is reported. If an FRU is reported, replace it and re-execute this diagnostic.

110	*** THE MAXIMUM NUMBER OF ERRORS HAS BEEN EXCEEDED (SCSIERR 110)
CAUSE	The user specified error limit has been reached.
ACTION	If more errors are desired, re-run the diagnostic assigning a larger value to the ERRCOUNT parameter of the run command.

111	*** UNRECOGNIZED COMMAND -- TYPE "HELP" FOR A LIST OF VALID COMMANDS (SCSIERR 111)
CAUSE	The specified command is not a valid command.
ACTION	Use the help facility to obtain a list of the commands that are valid and enter the desired command.

112	*** UNRECOGNIZED REPLY WAS FOUND (SCSIERR 112)
CAUSE	The reply that was entered in response to a prompt by the diagnostic is not valid.
ACTION	Refer to the prompt that was displayed and enter a response that is within the specified list of valid responses.

113	*** A NUMERICAL INPUT WAS EXPECTED BUT NOT RECEIVED (SCSIERR 113)
CAUSE	The reply that was entered in response to a prompt by the diagnostic is not a valid number.
ACTION	Re-enter number using only numeric characters and valid special characters (e.g. +, -, , etc.).

114	*** AN UNEXPECTED ERROR OCCURRED IN THE IO_SCSI DAR (SCSIERR 114)
CAUSE	A call to the SCSI device access routine resulted in an unexpected status return.
ACTION	The specific status generated by Io_Cs80 should have been displayed immediately prior to this error message. Report this set of error messages to support personnel.

115 *** ERROR IN TRANSMISSION DETECTED DURING READ LOOPBACK TEST: (SCSIERR 115)

Byte #	Octal Value Transmitted	Octal Value Received	Bit Positions In Error 01234567
======	==========	==========	==========

CAUSE One or more bytes of data that were received from the disk as a result of a loopback operation did not contain the expected value(s).

ACTION Data is most likely being corrupted along the data path between the host and the drive. Check all cable connections along the path and re-execute the diagnostic. If errors persist, execute appropriate diagnostics against the modules that lay in the path between the host and the device.

---

118 *** ERROR -- EXPECTED ! BYTES FROM THE DEVICE AND RECEIVED ! BYTES (SCSIERR 118)

CAUSE The number of bytes in the reply from the device was not what was expected. This is most likely a result of executing the diagnostic on a drive which is not supported by it.

ACTION Verify that the selected device is in the list of supported devices for the diagnostic (LIST ALL from the DUI). If it is, report the problem to support personnel.

---

119 *** THIS COMMAND IS NOT SUPPORTED ON THIS DEVICE (SCSIERR 119)

CAUSE The specified operation is not supported by the selected device.

ACTION While other devices supported by the diagnostic may support the specified command, the selected device does not.

---

120 *** FILE SYSTEM ERROR ENCOUNTERED WHILE RETRIEVING A MESSAGE (SCSIERR 120)

CAUSE An error was returned while attempting to obtain a message from the catalog. The actual error will have been displayed prior to this message.

ACTION This is a software error. Report to support personnel.

---

121 *** IDENTICAL READS FROM DISK RETURNED NON-IDENTICAL DATA (SCSIERR 121)

CAUSE A read operation was performed on the same sector twice and the data from both reads did not match exactly.

ACTION This is most likely a hardware problem with the drive. If loopback executes correctly, the read/write board in the drive is the probable cause of the error.

---

123 *** SELECTED DEVICE IS NOT A RECOGNIZED HP DEVICE (SCSIERR 123)

CAUSE The selected device does not identify itself with a recognized Hewlett-Packard product number or the version of diagnostic does not recognize or support the product.

ACTION Verify visually the type of product being diagnosed. If the product is a valid Hewlett-Packard product, update the diagnostic to a version that supports that product.

---

124 *** ERROR IN TRANSMISSION DETECTED DURING READ/WRITE CHANNEL TEST: (SCSIERR 124)

Cyl	Hd	Sect	Byte	Hex Value Trns	Hex Value Recd	Bit Positions In Error 01234567	Time Error Occurred
====	==	====	====	====	====	========	==================

CAUSE Data written to the disk does not agree with the data read.

ACTION Verify the data path to the device using the LOOPBACK command and WTR ERT on the target sector.

---

140 *** NO OPERATION WAS PERFORMED (SCSIERR 140)

CAUSE Due to a previous error, which has already been reported, no operation was performed.

ACTION Refer to actiou instructions for previously reported error.

---

200 *** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO SEND/RECEIVE INFORMATION FROM THE USER (SCSIERR 200)

CAUSE Due to a previous error, which has already been reported, the diagnostic was unable to communicate with the user interface process.

ACTION Refer to action instructions for previously reported error.

---

201 *** AN ERROR WAS ENCOUNTERED IN ATTEMPTING TO RETRIEVE A MESSAGE FROM THE CATALOG (SCSIERR 201)

CAUSE Due to a previous error, which has already been reported, the diagnostic was unable to extract a message from its message catalog.

ACTION Refer to action instructions for previously reported error.

202	***** AN ERROR WAS ENCOUNTERED IN ATTEMPTING TO CONVERT A NUMBER TO A STRING (SCSIERR 202)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to convert a number to a string.
ACTION	Refer to action instructions for previously reported error.

203	***** AN ERROR WAS ENCOUNTERED IN ATTEMPTING A BIT EXTRACTION OPERATION (SCSIERR 203)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to extract one or more bits from a number.
ACTION	Refer to action instructions for previously reported error.

204	***** THE SELECTED DEVICE COULD NOT BE OBTAINED FOR TESTING (SCSIERR 204)**
CAUSE	The DUI was unable to obtain access to the device. Device may be held exclusively by another process.
ACTION	Determine the access status of the device and make the corrective requirements.

207	***** SCSI CD INITIALIZATION FAILED. DEVICE MAY NOT BE SCSI OR DESCRIBE INFORMATION RETURNED MAY BE ERRONEOUS. SCSICD MAY NOT FUNCTION PROPERLY (SCSIERR 207)**
CAUSE	The diagnostic was unable to successfully obtain describe data from the drive. Since the diagnostic needs this information to function correctly, the user must make the determination to continue or terminate.
ACTION	Verify that the disk is infact a SCSI device and that the hardware path specified is valid.

208	***** AN ERROR WAS ENCOUNTERED IN ATTEMPTING TO COMMUNICATE WITH THE DIAGNOSTIC SYSTEM (SCSIERR 208)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to successfully perform a function which requires communication with the diagnostic system.
ACTION	Refer to action instructions for previously reported error.

209	***** YOUR RESPONSE WAS INVALID (SCSIERR 209)**
CAUSE	The data entered in response to a prompt was not valid.
ACTION	Refer to the prompt to determine the valid responses for the particular situation and enter one of the specified valid responses.

210	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO OBTAIN DATA FROM AN I/O BUFFER (SCSIERR 210)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to get data from its i/o buffer and, therefore cannot obtain data from the device.
ACTION	Refer to action instructions for previously reported error.

211	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO PLACE DATA INTO AN I/O BUFFER (SCSIERR 211)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable place data into its i/o buffer and, therefore, cannot send data to the device.
ACTION	Refer to action instructions for previously reported error.

212	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO OBTAIN AN I/O BUFFER (SCSIERR 212)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable obtain an i/o buffer and therefore, cannot send/receive data to/from the device.
ACTION	Refer to action instructions for previously reported error.

213	***** SECTION NUMBER ! IS NOT A VALID SECTION (SCSIERR 213)**
CAUSE	The section number input was not a valid section.
ACTION	Verify the valid sections and input your selection.

214	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO INITIALIZE THE PROGRAM (SCSIERR 214)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to start.
ACTION	Refer to action instructions for previously reported error.

215	***** HIGHER SECURITY IS NEEDED TO PERFORM THIS OPERATION (SCSIERR 215)**
CAUSE	The user requested an operation which is restricted to users with higher security than the user possesses.
ACTION	Contact system administrator if higher security level is desired.

216	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO RELINQUISH ACCESS TO THE CURRENT DEVICE. PROGRAM TERMINATING. (SCSIERR 216)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to relinquish access to the current device.
ACTION	Refer to action instructions for previously reported error.

217	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO OBTAIN ACCESS TO THE NEW DEVICE. ACCESS WAS NOT OBTAINED. (SCSIERR 217)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to obtain access to the current device.
ACTION	Refer to action instructions for previously reported error.

218	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO OBTAIN ACCESS TO THE OLD DEVICE. PROGRAM TERMINATING. (SCSIWARN 218)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to regain access to the previous device under test. Access to this device was attempted after access to the new device was not obtained.
ACTION	The device is currently being accessed exclusively by another process.

219	***** THE OPTION TO OBTAIN DISK SPACE IS NOT IMPLEMENTED FOR THIS OPERATING SYSTEM. (SCSIWARN 219)**
CAUSE	The operating system does not currently provide the functionality necessary to execute this command.
ACTION	Communicate the need for this functionality to the appropriate division.

220	***** AN ERROR OCCURRED WHILE ATTEMPTING TO OBTAIN DISK SPACE. (SCSIERR 220)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to obtain disk space.
ACTION	Refer to action instructions for previously reported error.

221	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO OBTAIN DISK SPACE. DISK SPACE RETURNED IS NOT ADEQUATE TO PERFORM TESTS. (SCSIERR 221)**
CAUSE	The area obtained from the disk device for test was not adequate to perform any substantial test. (SCSIWARN 221)
ACTION	Contiguous free space on the disk device is fragmented or there does not exist any free space large enough to test.

222	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO RELINQUISH DISK SPACE. (SCSIERR 222)**
CAUSE	Due to a previous error, which has already been reported, the diagnostic was unable to relinquish disk space.
ACTION	Refer to action instructions for previously reported error.

Contents

6

SCSI Digital Data Storage Tape Drive Diagnostic

The SCSIDDS diagnostic tests the DDS (Digital Data Storage) drive with the SCSI interface. This diagnostic detects failures down to a field replaceable unit (FRU) and is only available on operating systems which support the SHERLOCK diagnostic system.

Defects and Enhancements

Submit defect reports and enhancement requests for this diagnostic through the STARS database referencing Product Number 30600-10036.

Minimum Configuration

This product is available on any HP Precision Architecture (HPPA) system which supports the online diagnostics subsystem.

Intermediate hardware for communication with the DDS drive must be available on the host system.

The online diagnostics subsystem must be supported with DAR (device access routine) to access DDS, and I/O driver with access to diagnostic functions for DDS through the host operation system.

Default Tests

The program will allow the user to select which testing functions are to be executed, but a default set running only the trouble trees can also be invoked by not making any section or step specification. For security reasons, only non-destructive trouble trees will be run as defaults. If the user did not specify sections and steps, the following default sections and steps will be executed:

```
Section 10 : Non-exclusive and Non-disruptive hardware trouble tree
Section 12 : Non-destructive media trouble tree
```

These default sections and steps will be performed whether the drive is on-line or off-line.

User Environment

The online diagnostics subsystem has two access modes (non-exclusive/exclusive) and two test modes (non-destructive/destructive). Refer to the online Diagnostic User Interface (DUI) chapter of this manual for detailed definitions of these modes. The diagnostic system itself will determine the mode in which the test will be performed.

User Interface

The following subsections discuss SCSIDDS error and warning messages, prompts, and normal path flow.

Error Messages

Error messages will be displayed in the upper case preceded by three asterisks with a trailer containing the error number. The absence of a trailer or message other than (SCSIDDSERR *zzzz*) at the end indicates that the message originated from outside of this program. Such messages will generally provide detailed system dependent information related to the message issued by this diagnostic.

```
Example:  *** THIS IS A SAMPLE ERROR.  (SCSIDDSERR 32)
```

Warnings

Warning messages will be displayed in mixed case preceded by "* WARNING" with a trailer containing the warning number. Warnings may be accompanied by other system messages.

```
Example:  * WARNING - This is a sample warning. (SCSIDDSWARN zzzz)
```

Prompts

Prompts for data will be displayed in mixed case indented with respect to the error and warning messages and ended with the prompt symbol ">". It will include a range or selection of choices enclosed in parentheses. If appropriate, a default selection, enclosed by square brackets, supplied by the program will be used upon entry of a carriage return.

```
Example:    Do you wish to continue? (yes|no) [NO] >
```

Normal Path Flow

The SCSIDDS diagnostic can be accessed by the user through the Diagnostic User Interface(DUI) by the RUN SCSIDDS command. All parameters associated with the DUI's RUN command will be accepted by this diagnostic.

A description of SCSIDDS and all the sections are available through the DUI's Help facility.

When executing SCSIDDS, the following welcome message will be displayed:

```
**********************************************************************
******                                                          ******
******                    SCSI DDS DIAGNOSTIC                   ******
******                                                          ******
******           (C) Copyright Hewlett Packard Co. 1987         ******
******                    All Rights Reserved.                  ******
******                      Version  n.nn.nn                    ******
******                                                          ******
**********************************************************************
Welcome, Today is TUE, Aug 15, 1989 at 9:00 AM

        {Output from sections and steps executed}

SCSIDDS EXITING ...
```

When all sections and steps selected by the user are completed, or when a fatal error condition occurs, control will be returned to DUI.

Unless SCSIDDS is being called just to decode status, this diagnostic will first request access to the DDS drive from the diagnostic subsystem.

If access to the drive is denied, error messages from both the diagnostic and the subsystem will be displayed, and the program will terminate. If access to the device is obtained, however, the diagnostic will invoke a diagnostic subsystem routine to test the I/O path between the intermediate hardware and the DDS drive.

If the status of the I/O path test is not successful, a warning message will be displayed, and the diagnostic will continue. If the status is successful, the diagnostic will ask the device to identify itself. If the device fails to respond, the program will clear the device and ask again for the device to identify itself.

The SCSIDDS is CLEARing the drive and IDENTIFY will be performed again after clear.

If the device still fails to respond, the program will terminate.

```
*** DUE TO PROBLEMS WITH IDENTIFY, SCSIDDS CANNOT FUNCTION
    PROPERLY AND WILL THEREFORE TERMINATE (SCSIDDSERR 207)
```

If an unexpected identification code is received, an appropriate warning message will be displayed and the execution will continue to allow development devices to be diagnosed. In this case, the user will be asked whether to continue or not, for example:

```
*** THE SELECTED DEVICE IS NOT DIAGNOSABLE BY SCSIDDS(SCSIDDSERR 102)

                Do you wish to continue (Y/N)[N]?
```

If the device identifies itself as a valid candidate, the sections and steps specified by the user in the DUI run command will be executed. If the user did not specify sections and steps, the default sections and steps will be executed.

These default sections and steps will be performed whether the drive is on-line or off-line.

If the diagnostic request fails during the execution of any program section or step, the appropriate error message will be displayed. When the number of errors generated reaches the limit specified by the user in the ERRCOUNT parameter of the RUN command, an error will be displayed and the diagnostic will terminate.

If the ERRPAUSE parameter of the RUN command was assigned a value of "on", the diagnostic will pause after each error and will ask the user whether or not to continue the test.

Section Functional Descriptions

The following are the detailed descriptions of each section and step in SCSIDDS. Each section will be presented in the following format:

- An overview of the purpose of the section.
- How the section is divided into functional steps.
- An example of the output of the particular section.
- Examples of messages which will be displayed to the user during normal execution.

Output variables in the output examples are represented by one of the following codes (note the italics font):

n - decimal digit
(a 4 digit number would be represented by *nnnn*).
c - ascii character.
h - hexadecimal digit (0-F).
o - octal digit (0-7).
b - binary digit (0-1).
text - a description of the message.

An exclamation point (!) in the Error and Warning Messages section is normally a symbol representing the point at which certain sensitive words, phrases, or numbers should be inserted.

The value of some states and counters in the drive must be initialized. Otherwise, they will all contain Fs (hex) to indicate their unset state. This value will be displayed as "nv" (no value) by SCSIDDS. These are not error conditions unless other values are expected.

In compliance with the Online Diagnostic standard, the sections (and steps within sections) are executed in ascending numeric order. The order of each section and step function is determined by two factors: 1) customary online diagnostics subsystem order, so that it resembles other diagnostics to the extent practicable and 2) minimization of user intervention. An exception is in the selftest section (Section 5), because it was deemed desirable to allow the device to check itself before attempting less automatic and time consuming verification tests. The drive may be placed online at the beginning of the diagnostic, and is not required to reset online again until the next diagnostic.

The following table specifies what sections and steps will be available for execution in this diagnostic program:

Section 10	Non-Exclusive and Non-Disruptive Hardware Trouble Tree
Section 11	Disruptive Hardware Trouble Tree
Section 12	Non-destructive Media Trouble Tree
Section 13	Destructive Media Trouble Tree
Section 50	External Exerciser

Section 10—NON-EXCLUSIVE AND NON-DISRUPTIVE TROUBLE TREE

This section will perform a series of non-destructive tests to a DDS drive to detect and isolate the FRU. This diagnostic trouble tree will determine that the device is up and responding and that the SCSI channel appears to be free of problems.

OUTPUT:

```
Section 10 -- NON-EXCLUSIVE AND NON-DISRUPTIVE TROUBLE TREE
```

{Scenario 1:}

```
    REJECTED - Device identification
       Explanation: (nnn) hardware status error if available.

    SUSPECT:  (in descending order of probability)
        1. FRU    : SCSI cable
        2. Device powerfail
```

{Scenario 2:}

```
    FAILED   - Device identification
       SCSI code  = hhhh - WARNING: Device is not recognized
                                    as a DDS tape drive.
    REJECTED - SCSI Loopback test
       Explanation: (nnn) hardware status error if available.

    SUSPECT:  (in descending order of probability)
        1. FRU    : SCSI cable
        2. FRU nn: HP - SCSI Interface
```

{Scenario 3:}

```
    FAILED   - Device identification
       SCSI code  = hhhh - WARNING: Device is not recognized
                                    as a DDS tape drive.
    FAILED   - SCSI Loopback test
```

Byte #	Hex Value Transmitted	Hex Value Received	Erroneous bits 01234567
nn	hh	hh	bbbbbbbb
12	56	54	00000010
33	7F	3D	01000010
.	.	.	.
.	.	.	.

```
SUSPECT:  (in descending order of probability)
    1. FRU   : SCSI cable
    2. More than 1 device answering to the same address
    3. FRU nn: HP - SCSI Interface
```

{Scenario 4:}

```
FAILED   - Device identification
  SCSI code  = hhhh - WARNING: Device is not recognized
                                as a DDS tape drive.
PASSED   - SCSI Loopback test

SUSPECT:  (in descending order of probability)
    1. Invalid configuration
    2. Wrong device at specified address
    3. More than 1 device answering to the same address
    4. FRU nn: HP - SCSI Interface
    5. FRU   : SCSI cable
```

{Scenario 5:}

```
PASSED   - Device identification
REJECTED - SCSI Loopback test
   Explanation: (nnn) hardware status error if available.

SUSPECT:  (in descending order of probability)
    1. FRU   : SCSI cable
    2. FRU nn: HP - SCSI Interface
```

{Scenario 6:}

```
PASSED   - Device identification
FAILED   - SCSI Loopback test
```

Byte #	Hex Value Transmitted	Hex Value Received	Erroneous bits 01234567
nn	hh	hh	bbbbbbbb
12	56	54	00000010
33	7F	3D	01000010
.	.	.	.
.	.	.	.

```
SUSPECT:  (in descending order of probability)
    1. FRU   : SCSI cable
    2. More than 1 device answering to the same address
    3. FRU nn: HP - SCSI Interface
```

{Scenario 7:}

```
    PASSED   - Device identification
    PASSED   - SCSI Loopback test

    NO PROBLEMS DETECTED.
       Suggestions for further action:
          1. Run destructive trouble tree (section 11)
          2. Run operator assisted trouble tree (section 12)

End of Section 10 -- NON-EXCLUSIVE AND NON-DISRUPTIVE TROUBLE TREE
```

Section 11—HARDWARE TROUBLE TREE (Disruptive)

This section will check out the device as thoroughly as possible without on-site assistance (tape handling) and without taking excessively long to complete. This section is disruptive since the tests may clear data buffers and otherwise reset the device or even declare it unuseable by processes other than diagnostics. This section will attempt to clear the device to a known state and cause a series of device internal selftests to be run.

OUTPUT:

```
Section 11 -- HARDWARE TROUBLE TREE (Disruptive)

{Scenario 1:}
   [PASSED    - test description]
   [PASSED    - test description]
   [FAILED    - test description]
   [REJECTED - test description
       Explanation: [(nnn) hardware status error if available.]

    SUSPECT:  (in descending order of probability)
        1. name of suspected failing component/FRU
        2. name of suspected failing component/FRU


{Scenario 2:}

    PASSED    - Device clear
    PASSED    - Device identification
    PASSED    - SCSI Loopback test
    PASSED    - Selftest nnn

    NO PROBLEMS DETECTED.
       Suggestions for further action:
          1. Run operator assisted trouble tree (section 12)

End of Section 11 -- HARDWARE TROUBLE TREE (Disruptive)
```

Section 12—MEDIA TROUBLE TREE (Non-Destructive)

This section will perform a series of non-destructive tests to the media of the DDS tape drive to verify the integrity of the media. Several possible output scenarios from this section are given below. Note that this is not an exhaustive set of possibilities but is provided to give the user an idea of the type of output this section will produce.

OUTPUT:

```
Section 12 -- MEDIA TROUBLE TREE (Non-Destructive)

{Scenario 1:}

{Scenario 2:}

End of Section 12 -- MEDIA TROUBLE TREE (Non-Destructive)
```

Section 13—MEDIA TROUBLE TREE (Destructive)

This section will perform a series of destructive tests to the media of the DDS tape drive to verify the integrity of the media. Several possible output scenarios from this section are given below. Note that this is not an exhaustive set of possibilities but is provided to give the user an idea of the type of output this section will produce.

OUTPUT:

```
Section 13 -- MEDIA TROUBLE TREE (Destructive)

{Scenario 1:}


{Scenario 2:}


End of Section 13 -- MEDIA TROUBLE TREE (Destructive)
```

Section 50—INTERACTIVE EXTERNAL EXERCISER

The SCSIDDS External Exerciser is an interactive section which allows the user to access to a set of internal diagnostics and utilities within the DDS tape drive. The purpose of the exerciser is to aid service-trained personnel in troubleshooting DDS tape drives to a replaceable assembly level.

The following commands will be implemented:

```
Displays:       BLOCKLIMIT
                DESCRIBE
                IDENTIFY
                INQUIRY
                LOGS
                   ALL
                   FAULT LOG
                   ERROR RATE LOG
                   TAPE LOG
                MODESENSE
                REV
                SENSE

Diagnostics:    LOOPBACK
                MOTIONCHECK
                SELFTEST
                   Power on sequence
                   0  Reset Sequence
                   2  Error Rate Test Sequence
                   3  Error Rate Test Sequence with Data Compare
                   9  Mult-Controller Sequence
                  11  Dual-Ported RAM Sequence
                  13  Drive Controller Reset Sequence
                  14  Buffer Controller Reset Sequence
                  15  Interface Reset Sequence
                  32  Microprossor Test
                  33  ROM Checksum
                  34  Destructive RAM Test
                  35  Non-Destructive RAM Test
                  36  Complete RAM Test
                  38  Destructive Dual-Port RAM Test
                  40  CXD1356 Test
                  41  Sub-Area Microprocessor Test
                  42  Mechanism Test
                  48  Front Panel Check
                  49  MD Connectivity Test
                  50  Static Test
                  51  Recognition Switch Test
                  52  Voltage Test
                  53  Thread/Unthread Test
                  54  Reel Test
                  55  Drum Test
                  56  Capstan Test
```

```
                    57  Power-On Mechanism Selftest
                    64  Randomizer Off
                    67  Read Data Exerciser
                    68  Write Data Exerciser
                    69  Reposition Exerciser
                    70  Start/Stop Exerciser
                    71  Search Exerciser
                    72  BOM/EOM Exerciser
                    80  Device Area Test
                    96  Buffer Register Test
                    97  Buffer Function Test
                    98  Buffer RAM Test
                    99  Initialize Error Rate Log
                   112  SPIFI Test
                   113  SPIFI Loopback Test
                   128  Onboard DPR Test
                   129  Offboard DPR Test
                   133  Interface Loopback Tests
                   134  Buffer Initiated Loopback Tests
                  TREES
                     VERIFICATION
                     HARDWARE
                     MEDIA (NON-DESTRUCTIVE)
                     MEDIA (DESTRUCTIVE)
                  WORKOUT

Tape commands:    LOADTAPE
                  MEDIAREMOVAL
                     PREVENT
                     ALLOW
                  REWIND
                  UNLOADTAPE

Utilities:        CLEARLOG
                     ALL
                     FAUTL LOG
                     ERROR RATE LOG
                     TAPE LOG
                  RESET (SCSI Bus Device Reset)
                  TUR (Test Unit Ready)

Exerciser
 control:         EXIT
                  HELP
                  SUSPEND
```

Commands For Interactive External Exerciser

When the external exerciser is executed, the following prompt will be displayed to the user:

```
SCSIDDS>
```

Some commands may not be available if the requested device could not be obtained with exclusive access. The **EXIT** command will terminate the program. Syntax and usage help will be available via the **HELP** command.

If the user enters an unrecognized command the following error message with one or more of the applicable additional explanations will be displayed. For unrecognized command the user will be reprompted, e.g.:

```
*** UNRECOGNIZED COMMAND --
    TYPE "HELP" FOR A LIST OF VALID COMMANDS (DDSERR 111)

    SCSIDDS>
```

If paramaters are needed, the user will then be prompted for remaining information. The data prompts will be displayed in mixed case and will end with the prompt symbol **>**. They will also include a range or selection of choices enclosed in parentheses and, where apporpriate, a default selection, enclosed by square brackets, supplied by the program upon entry of a carriage return (<CR>). For example:

```
Enter the configuration value. (off|1..50) [OFF]>
```

The text enclosed in the parentheses indicates that the user is to type either **off** or an integer value between 1 and 50, inclusive. Each option is separated by a **|**. The default response **off**, as indicated by **[OFF]**, would be used if the user simply entered a carriage return. For items like configuration values, the current value will normally be displayed as the default.

Keyword responses (e.g., command names) may be abbreviated but not below the minimum number of characters necessary to distinguish them from all other valid responses. Responses will not be case sensitive. In addition, some commands will have special 2 or 3 character abbreviations which will also be recognized. These abbreviations are listed next to the command name in help messages. Where possible, command names, abbreviations, and other characteristics will conform to similar commands in the proposed Support User Interface Standard.

The **HELP** keyword will be recognized at all prompts.

If an error is encountered as a result of issuing a command to the device, error messages and the hardware status will be displayed. Note that some exerciser commands may issue several different device commands to the tape drive. Therefore, error messages may refer to command names other than those which are part of the exerciser.

Each command description below is in the following format:

Command Name

USAGE:
Explanation of what the command does and when it should be used.

SYNTAX:
Listing of command syntax (including any special abbreviation) and any parameter descriptions.

DATA PROMPTS:
Prompts to be displayed for needed information not available from the command parameter list.

OUTPUT FORMAT:
Information printed as a result of this command being executed.

ERROR MESSAGES:
Any error messages worth noting that may be generated as a result of this command. Note that this does not imply that all possible error messages that may be generated are listed here, but rather, only those that are especially worth mentioning in relation to the command.

There are several conventions used throughout the command output descriptions. They are as follows:

n	- a decimal digit (a 4 digit number would be represented as *nnnn*).
c	- an ascii character.
h	- a hexadecimal digit (0-F).
o	- an octal digit (0-7).
b	- a binary digit (0-1).
text	- a description of what will appear at that spot.

Comments in the input and output sequences are enclosed in curly brackets {}. These comments are only part of this document and are not output by the program. For example, a command description may contain the following:

```
Do you wish to continue? (Y|N)[N]>

            {if response is "N" this command will terminate}
```

However, the actual output of the program is:

```
Do you wish to continue? (Y|N)[N]>
```

BLOCKLIMIT

USAGE:

This command tells the drive to return its limits for record length. The command does not reflect the currently selected record size, only the available limits. MODE SENSE returns the current record size.

SYNTAX:

```
BLOCKLIMIT
```

DATA PROMPTS:

None.

OUTPUT FORMAT:

```
Record Length Limits
--------------------
Maximum Record Length = 16777215
Minimum Record Length = 1
```

CLEARLOG

USAGE:

This command is used to clear (initialize) the Fault Log and the Error Rate Log. The user will be prompted to choose the logs to be cleared, if not specified with the command. This command should be used with caution since logs that can help identify problems with the device will be cleared.

SYNTAX:

CLEARLOG [*log specifier*]
CLOG

log specifier:
ALL - All of the following clears will be done.
FAULT - Selftest failures and all problems encountered during normal operation.
ERROR RATE - Tape condition during normal operation.

DATA PROMPTS:

Clear logs:
0 - Clear both the Error Rate AND the Fault Logs
1 - Clear ONLY the Fault Log
2 - Clear ONLY the Error Rate Log
3 - Cancel Clear
Which log?

OUTPUT FORMAT:

ERT and Fault logs cleared.
{or}
Fault Log cleared
{or}
ERT logs cleared.

DESCRIBE

USAGE:

This command is used to display the product id and product revision level.

SYNTAX:

DESCRIBE

DATA PROMPTS:

None.

OUTPUT FORMAT:

```
The product is an HP35450A Digital Data Storage Drive.
The Product Revision Level is hhhh.
```

EXIT

USAGE:

This command may be entered any time at the SCSIDDS> prompt to terminate the execution of the External Exerciser.

SYNTAX:

EXIT

DATA PROMPTS:

None.

OUTPUT FORMAT:

End of Section 50 -- INTERACTIVE EXTERNAL EXERCISER

HELP

USAGE:

This command provides the user a list of all commands available in the External Exerciser. If the user types help, followed by the command name, a brief description of usage and command syntax will be provided.

SYNTAX:

HELP [*command name*]
?

OUTPUT FORMAT:

```
BLOCKLIMIT  -- Displays the Maximum and Minimum record lengths
CLEARLOG    -- Clears the various error logs on the device
DESCRIBE    -- Obtains describe information from the device
EXIT        -- Terminates execution of the External Exerciser
HELP        -- Provides this list of commands as well as more
               detailed descriptions and syntax of each command
IDENTIFY    -- Obtains identification information from the device
INQUIRY     -- Obtains 43 bytes of inquiry data from the device
LOADTAPE    -- Performs a full load of the tape in the drive
LOGS        -- Provides access to the device's error rate log,
               fault log, and tape log
LOOPBACK    -- Performs a write and read loopback of 256 bytes
               of data
MEDIAREMOVAL-- Capability to prevent or allow the tape to eject.
MODESENSE   -- Obtains mode sense data from the device
MOTIONCHECK -- Performs basic tape movement functions of the drive
RESET       -- Performs a HARD reset on the drive
REWIND      -- Causes the tape mounted on the drive to be rewound
               to BOT
REV         -- Displays the firmware revision numbers
SELFTEST    -- Provides access to device's internal diagnostics
SENSE       -- Obtains current sense data from the device
SUSPEND     -- Suspends SCSIDDS and returns control to the DUI.
               Type RESUME to return into diagnostic
TREES       -- Executes one of the SCSIDDS's diagnostic trouble tree
TUR         -- Tests if the scsi tape drive is ready
UNLOADTAPE  -- Performs a full unload of the tape
WORKOUT     -- Provides the user with different ways of reading and
               writing tape
```

IDENTIFY

USAGE:

This command returns the product id.

SYNTAX:

IDENTIFY

DATA PROMPTS:

None.

OUTPUT FORMAT:

```
The selected device has been identified as an HP35450A
     Digital Data Storage Drive
```

INQUIRY

USAGE:

This command requests device identification information from the drive. The SCSI dds information display is defined below.

SYNTAX:

```
INQUIRY
INQ
```

DATA PROMPTS:

None.

OUTPUT FORMAT:

```
=============================================================
                          INQUIRY DATA
=============================================================


Peripheral Qualifier = n
    The scsi dds is connected to logical unit number n.
Peripheral Device Type = n
    Sequential access device
  {or}
    The target is not capable of supporting a phsical device on this
    logical unit.
  {or}
    Unknown device type
  {or}
    fault.... peripheral device type is not supported
Removable Medium Bit = n
    The tape can be removed
  {or}
    The tape cannot be removed
Device Type Qualifier = n
ISO Version = n
    The drive does not necessarily comply with the ISO version of SCSI
  {or}
    The drive does comply with the ISO version of SCSI
ECMA Version = n
    The drive does not necessarily comply with the ECMA version of SCSI
  {or}
    The drive does comply with the ECMA version of SCSI
ANSI Approved Version = n
The drive complies with the ANSI version of SCSI-n
Asynchronous Event Notification Bit = n
    The drive does not support asynchronous event notification.
  {or}
    The drive does support asynchronous event notification.
Terminate I/O Processing Bit = n
    The drive does not support Terminate I/O Processing message.
```

```
  {or}
    The drive does support Terminate I/O Processing message.
Response Data Format = n
The Inquiry Data format complies with the ANSI version of SCSI-n
Relative Addressing Mode Bit = n
    Relative Addressing Mode is not supported.
  {or}
    Relative Addressing Mode is supported.
32-bit Wide Data Transfer Bit = n
    32-bit wide data transfers is not supported.
  {or}
    32-bit wide data transfers is supported.
16-bit Wide Data Transfer Bit = n
    16-bit wide data transfers is not supported.
  {or}
    16-bit wide data transfers is supported.
Synchronous Data Transfer Bit = n
    Synchronous data transfers is not supported.
  {or}
    Synchronous data transfers is supported.
Linked Commands Bit = n
    The drive does not support linked commands.
  {or}
    The drive does support linked commands.
Tagged Command Queuing Bit = n
    Tagged command queuing is not supported.
  {or}
    Tagged command queuing is supported.
Soft Reset Bit = n
    The drive responds to the reset condition with a HARD reset.
  {or}
    The drive responds to the reset condition with a SOFT reset.
Vendor Identification = 8 byte string
Product Identification = 16 byte string
Product Revision Level = 4 byte string
Manufacturing Date Code = 4 byte string
Power-On Reset Time = n
The drive supports HPCS version n
```

LOADTAPE

USAGE:

This command will load and rewind the tape.
The tape needs to be in the drive when the command is issued.

SYNTAX:

LOADTAPE
LT

DATA PROMPTS:

None.

OUTPUT FORMAT:

```
Load completed successfully.
```

LOGS

USAGE:

This command allows the user to read log information from the drive. The user may choose to display either all or one of the Fault, Error Rate, or Tape Logs.

SYNTAX:

LOGS [*log specifier*]

log specifier

ALL - All of the following logs will be displayed.

FAULT - The drive maintains a history of the last 30 errors which have occurred within the drive, together with a 'timestamp' of when they occurred. The information is lost when the drive is power-cycled.

ERROR RATE - Used to monitor the condition of a tape during normal operation and may also be used to monitor tape errors. The log contains information on current tape usage, the number of groups accessed, and the error counts. Entries are kept for both hard and soft errors. The data available from the log is valid until the next power-cycle, reset, CLEAR LOGS, Error Rate test, or TAPE LOAD, when it will be reset. The log is not saved to tape when the tape is unloaded.

TAPE - This information is contained within RAM in the Drive Controller. When a cassette is loaded, the contents of the System area are copied into this log. The log is updated as the tape is used and is copied back into the System area when the cassette is unloaded (the System area only contains 'previous' and 'total' information - when the cussette is unloaded, the 'current' values become the System area 'previous' values).
The term 'current' refers to information generated during the current load of the tape; 'previous' refers to information generated during the last tape load; 'total' refers to information generated during the life of the tape since it was first initialized including the current load.

DATA PROMPTS:

```
Which log is to be displayed?
    0 -- ALL
    1 -- FAULT
    2 -- ERROR RATE
    3 -- TAPE

Your selection [ALL]>
```

OUTPUT FORMAT:

```
        {For Fault Log:}

  RAW DATA :
  --------
  Byte Count = 29

                0   1   2   3   4   5   6   7   8   9
               --  --  --  --  --  --  --  --  --  --
  Byte    0    01  00  1A  02  00  00  FF  FF  FF  03
  Byte   10    46  01  00  20  00  00  06  00  FF  0C
  Byte   20    0E  01  02  80  00  00  06  FF  FF
      ================================================================
                                  FAULT LOG
      ================================================================
      Current Time          -    16777215
      ----------------------------------------------------------
      Entry Number =    1
      Error Class  -     Interface controller diagnostic errors
      Error Code   70 -- SCSI controller register error.
      Faulty FRU   - (1) Drive Mechanism & Associated Electronics
      Fault Time         -      393471


      ----------------------------------------------------------
      Entry Number =    2
      Error Class  -     Multi-processor errors
      Error Code   14 -- On board dual-port RAM test error.
      Most Probable
      Faulty FRU   - (1) Drive Mechanism & Associated Electronics
      Second Most Probable
      Faulty FRU   - (2) Buffer Controller
      Fault Time         -      458751
```

```
      {For Error Rate Log:}

RAW DATA :
--------
Byte Count = 29

              0   1   2   3   4   5   6   7   8   9
             --  --  --  --  --  --  --  --  --  --
Byte    0    02  00  1A  00  00  00  00  00  00  00
Byte   10    00  00  00  00  00  00  00  00  00  00
Byte   20    00  00  00  00  00  00  00  00  00
    ================================================================
                              ERROR RATE LOG
    ================================================================
    Number of groups written                          = 0
    Number of groups has RAW rewrites                 = 0
    Number of total RAW counts                        = 0
    Number of write hard errors                       = 0
    Number of groups read                             = 0
    Number of groups corrected with a C3-ECC pass     = 0
    Number of groups retried                          = 0
    Number of total retry counts                      = 0
    Number of read hard errors                        = 0
```

```
      {For Tape Log:}

RAW DATA :
--------
Byte Count = 39

              0   1   2   3   4   5   6   7   8   9
             --  --  --  --  --  --  --  --  --  --
Byte    0     03  00  24  00  00  00  00  00  00  00
Byte   10     00  00  00  00  00  00  00  00  00  00
Byte   20     00  00  00  00  00  00  00  00  00  00
Byte   30     00  00  00  00  00  00  00  00  13
    ================================================================
                                TAPE LOG
    ================================================================
    Total tape load          =       19

    Current groups written   =        0
    Current RAW retries      =        0
    Current groups read      =        0
    Current ECC-3 retries    =        0

    Previous groups written  =        0
    Previous RAW retries     =        0
    Previous groups read     =        0
    Previous ECC-3 retries   =        0

    Total groups written     =        0
    Total RAW retries        =        0
    Total groups read        =        0
    Total ECC-3 retries      =        0
```

LOOPBACK

USAGE:

This command tests the 512 Kbyte data buffer and the SCSI integrity of the drive. A read buffer will be performed after a write buffer and the results will be checked against the expected pattern. The tape will be rewound afterwards to allow normal operation of the drive.

SYNTAX:

WRB

DATA PROMPTS:

None.

OUTPUT FORMAT:

Loopback completed successfully.

{OR}

*** ERROR IN TRANSMISSION DETECTED DURING READ LOOPBACK TEST:(DDSERR n)

Byte #	Octal Value Transmitted	Octal Value Received	Bit Positions In Error 01234567
nn	*ooo*	*ooo*	*bbbbbbbb*
{EXAMPLE:}			
12	126	124	00000010
33	177	75	01000010
.	.	.	.
.	.	.	.
.	.	.	.

MEDIAREMOVAL

USAGE:

This command will re-enable the Eject button after a PREVENT MEDIA REMOVAL. After issuing this command, the drive will eject the DDS tape after the completion of an UNLOAD.

SYNTAX:

```
MEDIAREMOVAL

   where:
      PREVENT     - Prevents the tape from being ejected follwing an
                    UNLOADTAPE command.
      ALLOW       - Allows the tape to be ejected follwing an
                    UNLOADTAPE command.
```

DATA PROMPTS:

```
Prevent/Allow Medium Removal :
     0 - Allow the tape to be ejected following an UNLOADTAPE command.
     1 - Prevent the tape from being ejected following an UNLOADTAPE
         command.
```

OUTPUT FORMAT:

```
          Allow medium removal enabled.
          Media will be ejected following an UNLOAD command.
  {or}
          Prevent medium removal enabled.
          Media will NOT be ejected following an UNLOAD command.

NOTE : The Prevention of Medium Removal condition terminates upon
       receipt of a BUS DEVICE RESET command or by a hard reset.
```

MODESENSE

USAGE:

This command tells the drive to report the current configuration values of the media to the host.

SYNTAX:

MODESENSE

DATA PROMPTS:

```
Which page to be displayed?
    0 -- All Supported pages
    1 -- Disconnect-Reconnect Page
    2 -- Device Configuration Parameters
    3 -- Medium Partition Parameters
    4 -- Vendor-Unique Command Timers
    5 -- No Pages

Your selection [0]>
```

OUTPUT FORMAT:

example output from all pages:

```
================================================================
                          MODE SENSE DATA
================================================================

RAW DATA :
--------
Byte Count = 168

              0   1   2   3   4   5   6   7   8   9
             --  --  --  --  --  --  --  --  --  --
Byte    0    A7  00  10  08  13  00  00  00  00  00
Byte   10    00  00  02  0E  00  00  00  00  00  00
Byte   20    00  00  01  EE  00  00  00  00  10  0E
Byte   30    08  00  00  00  00  32  60  00  18  00
Byte   40    00  00  00  00  11  08  01  00  10  03
Byte   50    00  00  00  00  20  70  00  9F  FF  FF
Byte   60    01  9F  FF  FF  03  9F  FF  FF  05  9F
Byte   70    FF  FF  08  9F  FF  FF  0A  9F  FF  FF
Byte   80    10  9F  FF  FF  11  9F  FF  FF  12  9F
Byte   90    FF  FF  13  9F  FF  FF  15  9F  FF  FF
Byte  100    16  9F  FF  FF  17  9F  FF  FF  18  A0
Byte  110    00  00  19  9F  FF  FF  1A  9F  FF  FF
Byte  120    1B  9F  FF  FF  1C  9F  FF  FF  1D  9F
Byte  130    FF  FF  1E  9F  FF  FF  2B  9F  FF  FF
Byte  140    34  9F  FF  FF  3B  9F  FF  FF  3C  9F
Byte  150    FF  FF  4C  9F  FF  FF  4D  9F  FF  FF
Byte  160    E0  9F  FF  FF  E1  9F  FF  FF
```

```
INTERPRETED DATA :
---------------
Sense Data Length = 167 (number of bytes of MODE SENSE data)
Media Type = 0
    Only one media type is supported.
Write Protected Bit = 0
    The tape is write-enabled.
Buffered Mode = 1
    Buffered Mode.  The drive reports GOOD status on WRITE commands
    as soon as the data record has been transferred to the buffer.
Speed = 0
    The dds only supports one speed.  The default speed.
Block Descriptor Length = 8
Density Code = 19
    Hewlett-Packard/Sony DDS Format
Number of Records = 0
    All the remaining logical records on the tape will have the
    medium characteristics specified by the block descriptor, or until
    a subsequent MODE SELECT changes those parameters.
Record Length = 0 (length in bytes of each logical record)
   ----------------------------------------------------------------
    PAGE CODE DATA:
Page Saveable Bit = 0
    There is no non-volatile RAM on the drive into which parameter
    data may be saved.
Page Code = 2 ( Disconnect-Reconnect Page )
Additional Length = 14 (number of remaining bytes in this page)
Buffer Full Ration = 0 (not supported)
Buffer Empty Ration = 0 (not supported)
Bus Inactivity Limit = 0 (not supported)
Disconnect Time Limit = 0
    Minimum time in 100 microsecond increments that the drive will wait
    after releasing the SCSI bus before attempting reselection.
Connect Time Limit = 0 (not supported)
Maximum Burst Size = 494
    This is the maximum amount of data the drive will transfer during
    a data phase before disconnecting.
DTDC = 0 (not supported)
    ----------------------------------------------------------------
    PAGE CODE DATA:
Page Saveable Bit = 0
    There is no non-volatile RAM on the drive into which parameter
    data may be saved.
Page Code = 16 ( Device Configuration Page )
Additional Length = 14 (number of remaining bytes in this page)
Change Active Partition Bit = 0
Change Active Field = 0
Active Format Field
    Disable RAW = 0
    Read-After-Write is Enabled.
```

```
C3 ECC - Third Level Correction Code = 1
    Third Level Error Correction (C3 ECC) is Enabled.
N-Group = 0
    A group will repeatedly be written 0 times to tape.
Active Partition Field = 0
Write Buffer Full Ratio = 0 (not supported)
Read Buffer Empty Ratio = 0 (not supported)
Write Delay Time = 50
    The drive will wait 50 milliseconds with a partially full buffer
    before forcing the data to tape.
DBR - Data Buffer Recovery bit = 0
The drive does not support data buffer recovery using the 'RECOVER
    BUFFERED DATA' command.
RIS - Record Identifiers Supported bit = 1
    The tape has recorded information about the logical record ID
    relative to the partition.
RSmk - Report Setmark bit = 1
    Save-Set Marks will be reported to the drive.  (default)
AVC - Automatic Velocity Control bit = 0 (not supported)
SOCF - Stop on Consecutive Filemarks field = 0 (not supported)
RBO - Recover Buffer Order bit = 0 (not supported)
REW - Report Early-Warning End-of-Media bit = 0
    The drive will not report the early-warning condition on reads
    but will report early-warning on writes at a distance of 500mm
    before PEOT or synthetic PEOT.
Gap Size = 0 (not DDS specific)
EOD Defined = 0
    The drive will use its default EOD mark before any change of
    direction following a write-type operation.
EEG - Enable EOD Generation bit = 1
    The drive generates an EOD mark prior to a change of direction
    following a write-type operation.
SEW - Synchronize at Early Warning bit = 1
    Data written to the drive after Early-Warning End-Of-Media will be
    written to the tape as normal.
Buffer Size at Early Warning = 0 (not supported)
Select Data Compression Algorithm = 0
    The drive does not use data compression before writing data to tape
    ---------------------------------------------------------------
    PAGE CODE DATA:
Page Saveable Bit = 0
    There is no non-volatile RAM on the drive into which parameter
    data may be saved.
Page Code = 17 ( Medium Partitions Parameter Page )
Additional Length = 8 (number of remaining bytes in this page)
Maximum Additional Partitions supported by the drive = 1
Additional Partitions Defined = 0
FDP - Fixed Data Partitions bit = 0 (not supported)
SDP - Select Data Partitions bit = 0 (not supported)
IDP - Initiator Defined Partitions bit = 0
PSUM - Partition Size Unit of Measure field = 2
```

```
    Partition size = 2 megabytes.
Medium Format Recognition = 3
    The drive is capable of format and partition recognition.
Partition Size = 0
    Partition 1 eize = 0 megabytes.
    ----------------------------------------------------------------
    PAGE CODE DATA:
Page Saveable Bit = 0
    There is no non-volatile RAM on the drive into which parameter
    data may be saved.
Page Code = 32 ( Vendor-Unique Command Timers Page )
Additional Length = 112 (number of remaining bytes in this page)

Command Operation Code = 0      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 1      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 3      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 5      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 8      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 10      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 16      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 17      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 18      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 19      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 21      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 22      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 23      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).
```

```
Command Operation Code = 24      ( HP Common SCSI )
    The command will execute in 0 milliseconds (max).

Command Operation Code = 25      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 26      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 27      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 28      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 29      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 30      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 43      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 52      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 59      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 60      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 76      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 77      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 224      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).

Command Operation Code = 225      ( HP Common SCSI )
    The command will execute in 2097151 milliseconds (max).
```

MOTIONCHECK

USAGE:

This command checks out basic tape movement functions of the drive. The tape will be erased and rewound. A set of 5 files (20 records each) will be written to tape and the tape rewound. The following sequence will then be executed:

```
+ = forward space
- = backward space

     + 2 files
          read record (to verify position)
     - 1 file
          read record (to verify position)
     + 1 record
          read record (to verify position)
     - 1 record
          read record (to verify position)
     + 3 files
     - 4 files
     + 2 files
     - 1 files
          read record (to verify position)
     +11 records
     - 3 records
     +11 records
     -19 records
     + 2 records
          read record (to verify position)
     - 4 records
          verify past EOF
     +20 records
          verify past EOF
     + 4 files
          verify tape runaway
          rewind
     - 1 record
          verify past BOT
     - 1 file
          verify past BOT
```

SYNTAX:

```
MOTIONCHECK
MC
```

DATA PROMPTS:

```
This commad takes approximately xx minutes to complete...
Do you wish to continue (Y/N)[N]?
```

{*xx* will be filled in after the actual testing of the command}

OUTPUT FORMAT:

```
Motion check has completed successfully.
```

RESET

USAGE:

This command performs a reset operation on the DDS tape drive. When this is executed, the currently executing command will be aborted and the drive will be at a Bus Free state. Following that will be a hard reset leaving the drive in a power-cycled state. No data in the buffer will be saved.

SYNTAX:

RESET

DATA PROMPTS:

None.

OUTPUT FORMAT:

Reset completed successfully

REWIND

USAGE:

This command causes the tape mounted on the drive to be rewound to BOT. The user will be asked for the option of immediate bit being enabled or disabled. If enabled the drive first writes any remaining buffered data to tape followed by an EOD marker. It then returns status to the host before the actual operation has completed.

SYNTAX:

```
REWIND
RW
```

DATA PROMPTS:

```
Immediate Mode?
    0 -- Immediate bit not set.  Status will be returned after
         has completed
    1 -- Immediate bit set.  The drive first writes any remaining
         buffered data to tape followed by an EOD marker.  It then
         returns status to the host before the actual operation has
         completed.
```

OUTPUT FORMAT:

```
        REWIND of tape was successful.
```

REV

USAGE:

This command displays the current firmware revision numbers in the drive in a hex format.

SYNTAX:

REV

DATA PROMPTS:

None.

OUTPUT FORMAT:

```
============================================================
                    FIRMWARE REVISIONS (HEX)
============================================================

Controller id            = 01    ( DRIVE )
Firmware Version         = 0B
Firmware Revision        = 07
Firmware Revision Update = FF
------------------------------------------------------------
Controller id            = 02    ( BUFFER )
Firmware Version         = 0B
Firmware Revision        = 10
Firmware Revision Update = 00
------------------------------------------------------------
Controller id            = 03    ( INTERFACE )
Firmware Version         = 0B
Firmware Revision        = 0E
Firmware Revision Update = 53
------------------------------------------------------------
```

SELFTEST

USAGE:

From this command, the user may either choose to run the internal power-on test or run an internal diagnostic test. The tests which can be selected are device dependent and are fully described in the support documentation of the tape drive. The user will be prompted for all required information.

The selftest may be automatically repeated up to 1,000 times. The execution count used is implemented at the diagnostic level and NOT within the device. The selftest request will be issued until the specified execution count is reached or until the selftest fails. Once the selftest sequence has been started, the user may abort via the program interrupt (CNTL-Y on MPE XL, CNTL-C on HP-UX).

SYNTAX:

```
SELFTEST
ST
```

DATA PROMPTS:

```
Enter one of the following options:
                  0 - Run power on sequence
                  1 - Run specific internal selftest(0-134)
                  2 - Exit selftests
Your selection (0..2) [1] >

{if user selects 1 then the following prompts will appear}

The dds supports the following tests:
   0  Reset Sequence
   2  Error Rate Test Sequence
   3  Error Rate Test Sequence with Data Compare
   9  Mult-Controller Sequence
  11  Dual-Ported RAM Sequence
  13  Drive Controller Reset Sequence
  14  Buffer Controller Reset Sequence
  15  Interface Reset Sequence
  32  Microprossor Test
  33  ROM Checksum
  34  Destructive RAM Test
  35  Non-Destructive RAM Test
  36  Complete RAM Test
  38  Destructive Dual-Port RAM Test
  40  CXD1356 Test
  41  Sub-Area Microprocessor Test
  42  Mechanism Test
  48  Front Panel Check
  49  MD Connectivity Test
  50  Static Test
```

```
  51  Recognition Switch Test
  52  Voltage Test
  53  Thread/Unthread Test
  54  Reel Test
  55  Drum Test
  56  Capstan Test
  57  Power-On Mechanism Selftest
  64  Randomizer Off
  67  Read Data Exerciser
  68  Write Data Exerciser
  69  Reposition Exerciser
  70  Start/Stop Exerciser
  71  Search Exerciser
  72  BOM/EOM Exerciser
  80  Device Area Test
  96  Buffer Register Test
  97  Buffer Function Test
  98  Buffer RAM Test
  99  Initialize Error Rate Log
 112  SPIFI Test
 113  SPIFI Loopback Test
 128  Onboard DPR Test
 129  Offboard DPR Test
 133  Interface Loopback Tests
 134  Buffer Initiated Loopback Tests

Selftest number.                        [0]>

{all tests will prompt for number of times to execute}
          Number of times to execute:
                    0 - continuous
                    1 - run once
                    2 - run 10 times
                    3 - run 100 times
                    4 - run 1000 times
          Your selection (0..4) [1]>

          Do you want to break on first error?
                     0 - break on first error
                     1 - no break on error
          Your selection (0..1) [0]>

  {Parameter prompts are conditional depending on the selftest number}
          Parm A (0..255) [0]>
          Parm B (0..255) [0]>
          Parm C (0..255) [0]>
```

OUTPUT FORMAT:

```
Running selftest #n ...


 --------------------
 | Selftest Results |
 --------------------
Error Class  -     Runtime errors | Drive controller diagnostic
                   errors | Buffer controller diagnostic errors |
                   Interface controller diagnostic errors |
                   Multi-processor errors | Operational status |
                   Unknown error class.
 Error Code    n -- {description}
 Faulty FRU   - (1) Drive Mechanism & Associated Electronics
{or}
 Faulty FRU   - (2) Buffer Controller
{or}
 Faulty FRU   - (3) Interface Controller
{or}
 No FRU
```

ERROR MESSAGES:

Special warnings will be given the user upon selection of an invalid selftest code, depending on the code selected. In most cases the user will be prompted for a valid selftest.

*** SELFTEST ! IS NOT IMPLEMENTED BY THE SELECTED DEVICE.

SENSE

USAGE:

This command will obtain the current sense data from the device and display it in readable text form.

SYNTAX:

SEN

DATA PROMPTS:

None.

OUTPUT FORMAT:

```
==============================================================
                        SENSE DATA
==============================================================
```

RAW FORM:

byte#	Description	Hex Value	Bit Map 76543210	Decimal Value
0	Valid(7) Error Code(6-0)	70	01110000	112
1	Segment Number	00	00000000	0
2	Mark(7) EOM(6) ILI(5) Sense Key(3-0)	02	00000010	2
3	Information Byte(MSB)	00	00000000	0
4	Information Byte	00	00000000	0
5	Information Byte	00	00000000	0
6	Information Byte(LSB)	00	00000000	0
7	Additional Sense Length	0B	00001011	11
8	Command Specific Information (MSB)	00	00000000	0
9	Command Specific Information	00	00000000	0
10	Command Specific Information	00	00000000	0
11	Command Specific Information (LSB)	00	00000000	0
12	Additional Sense Code	3A	00111010	58
13	Additional Sense Code Qualifier	00	00000000	0
14	Field Replaceable Unit Code	00	00000000	0
15	SKSV(7) Sense Key Specific Byte(6-0)	00	00000000	0
16	Sense Key Specific Byte	00	00000000	0
17	Sense Key Specific Byte	00	00000000	0
18	Reserved	00	00000000	0


```
INTERPRETED SENSE DATA:

   Reported error is associated with the most recent command issued
   to device.
   Medium is not present.  Device is in a quiescent state with no
   media in the drive.

    Error Code  -- No error.
```

SUSPEND

USAGE:

This command temporarily suspends execution of the diagnostic, allowing the user to issue other DUI commands. This feature is handy for retaining any default values accumulated by WORKOUT or other SCSIDDS functions. To return to SCSIDDS, the user may just enter "resume" at the DUI prompt.

SYNTAX:

SUSPEND

DATA PROMPTS:

None.

OUTPUT FORMAT:

DUI>

TREES

USAGE:

This command causes one of SCSIDDS's diagnostic trouble trees to be executed.

SYNTAX:

TREES [*tree specifier*]

tree specifiers:

```
ALL                    - All of the following
VERIFICATION           - Non-destructive trouble tree.
HARDWARE               - Destructive trouble tree.
NON-DESTRUCTIVE MEDIA  - Non-destructive media trouble tree.
DESTRUCTIVE MEDIA      - Destructive media trouble tree.
```

DATA PROMPTS:

```
Which tree is to be executed?
    0 -- ALL
    1 -- VERIFICATION
    2 -- HARDWARE
    3 -- NON-DESTRUCTIVE MEDIA
    4 -- DESTRUCTIVE MEDIA

Your selection [<cr> = none]>
```

OUTPUT FORMAT:

{same as sections 10 to 13}

TUR

USAGE:

This command tests whether a tape is loaded and the drive is ready for use.

SYNTAX:

TUR

DATA PROMPTS:

None.

OUTPUT FORMAT:

DDS is ready with a tape loaded

{or drive will return a check condition and status will be decoded}

UNLOADTAPE

USAGE:

This command causes the automatic unloading to be initiated. The request will be honored even if the tape has not been rewound. The tape will be unloaded and ejected. If the PMR is set, then the tape will remain in the drive.

SYNTAX:

```
UNLOADTAPE
ULT
```

DATA PROMPTS:

None.

OUTPUT FORMAT:

```
Tape was successfully unloaded.
```

WORKOUT

USAGE:

This command provides the user with different ways of reading and writing tape. It is useful in isolating medium vs drive problems, verifying media, isolating certain classes of read/write problems, and simulating system use of the tape drive.

This command will cause records to be written to tape, the tape rewound, and those same records read back off the tape. The user will have the option of whether or not to verify the records read match the ones written earlier. The user will also be able to vary the density record lengths, record content, number of files per tape, and loop count.

The user will have an option to clear the drive's internal error logs prior to writing to the tape. The user will be warned and given an opportunity to back out of the command to look at and/or record the log before reissuing this command. This will also serve as the mechanism for correcting invalid entries.

There will be a choice of byte patterns to be written to the tape. Each record will consist of repetitions of the pattern up to the record size limit specified.

Data Verification

This command will compare each record read to what this program originally sent, displaying any discrepancies.

SYNTAX:

```
WORKOUT
WO
```

DATA PROMPTS:

```
WORKOUT

*******************************************
*                WARNING                  *
*   This command may destroy user data    *
*******************************************

Do you wish to continue (Y/N)[N]?

Reset in progress...

Reset completed successfully


Clear the Error events, W/R error counters logs (Y/N)[N]?
```

```
Is the media a Short(0.5 GBytes) or a Long(1.3 Gbytes)
media? (S/L)[S]>

Do you want to write to the end of the media? (Y/N)[N]>

Do you want:
  R = random transfer length
  U = user input transfer length

Which option would you like (R/U)[U]?

Input transfer length (1 <= bytes <= nnnn)[1]?

Input number of records to write to media(1 <= records <= n)[1]?

 The sources of the data pattern to be used are:
   sequential     [ff,00,01,02,..fe]
   alternating    [00,ff,00,ff,..]
   butterfly      [00,ff,01,fe,..]
   single         [you will be prompted]
 Which pattern source would you like [sequential]>
 {if user chooses single}
Input the pattern in hex:

 Input loop count (1 <= count <= 254)[1]?
Writing to media is in progress...

Reading from media and comparing data are in progress...
```

OUTPUT FORMAT:

```
      No errors were detected in the media test
                      or
*** ERROR IN TRANSMISSION DETECTED DURING READ/WRITE MEDIA TEST:

                                 Bit
                    Hex   Hex   Positions
                    Value Value In Error
REC #  BYTES #      TRNS  RECD  01234567     Time Error Occurred
================== ====  ====  ========  ==========================
 2       50          FF    60  11000011   Fri, Mar 17, 1989  3:30 PM
 .        .           .     .      .                    .
 .        .           .     .      .                    .
 .        .           .     .      .                    .

WORKOUT COMPLETED.
```

Error and Warning Messages

The following are general error/warning messages which may be encountered during the execution of SCSIDDS. Note, however, that system dependent error messages may be displayed by the subsystem along with any error message generated by this diagnostic. All error message without the (DDSERR #) trailer are generated by the subsystem.

Errors which have explanatory notes (preceded by "—") will normally display only one of the notes listed here. The exclamation point (!) is a symbol used to indicate the point at which context dependent information is to be placed.

100	*** WARNING -- THE I/O PATH MAY NOT BE FUNCTIONING PROPERLY (DDSERR 100)
CAUSE	The data path between the host and specified device is not functioning properly.
ACTION	If it is determined that a device has not been selected for testing, the diagnostic will be aborted. Otherwise, the warining message will be displayed and the diagnostic will continue.

101	*** DEVICE FAILED TO RESPOND TO ! COMMAND (DDSERR 101)
CAUSE	Device timed-out during execution of command.
ACTION	Be sure scsi cable is connected to drive correctly. Reset drive and try again.

102	*** SCSIDDS IS UNABLE TO DIAGNOSE THE SELECTED DEVICE (DDSERR 102)
CAUSE	The device was not identified as a HP35450A after responding from an inquiry command.
ACTION	If this occurs during the diagnostic initialization routine, the user is asked to continue. If this occurs during section 10 or 11, the trouble tree operations will not be performed.

108	*** THIS COMMAND IS NOT IMPLEMENTED ON THE SELECTED DRIVE (DDSERR 108)
CAUSE	The selected command is not implemented by the device.
ACTION	Try another command.

109	*** DEVICE ENCOUNTERED AN ERROR WHILE EXECUTING THE ! COMMAND (DDSERR 109)
CAUSE	The command did not execute successfully.
ACTION	Interpret results from sense data.

110	***** THE MAXIMUM NUMBER OF ERRORS HAS BEEN REACHED (DDSERR 110)**
CAUSE	The maximum allowed number of errors has been reached.
ACTION	The diagnostic is aborted.

111	***** UNRECOGNIZED COMMAND --** **TYPE "HELP" FOR A LIST OF VALID COMMANDS (DDSERR 111)**
CAUSE	Parser did not recognize the command.
ACTION	Enter a valid command.

112	***** UNRECOGNIZED REPLY WAS FOUND (DDSERR 112)**
CAUSE	Parser did not recognize the reply.
ACTION	Enter a valid reply.

113	***** A NUMERICAL INPUT WAS EXPECTED BUT NOT RECEIVED (DDSERR 113)**
CAUSE	Parser expected an integer.
ACTION	Enter a valid integer.

114	***** AN UNEXPECTED ERROR OCCURED IN THE IO_SCSISA DAR (DDSERR 114)**
CAUSE	The diagnostic returned an unexpected status from the device access routine.
ACTION	The user should never see this. Used for debugging diagnostic.

115 ***** ERROR IN TRANSMISSION DETECTED DURING READ LOOPBACK TEST: (DDSERR 115)**

CAUSE The buffer read does not match the buffer written to the device.

ACTION Try different media. If it fails again, contact your support engineer.

116

Byte #	Octal Value Transmitted	Octal Value Received	Bit Positions In Error 01234567
!	!	!	!

CAUSE Bytes expected do not match byte received during loopback test.

ACTION Try different media. If it fails again, contact your support engineer.

---

118 ***** ERROR -- EXPECTED ! BYTES FROM THE DEVICE AND RECEIVED ! BYTES (DDSERR 118)**

CAUSE The number of bytes recieved from the drive do not match what was expected.

ACTION Try different media. If it fails again, contact your support engineer.

---

140 ***** POWER FAIL - NO OPERATION WAS PERFORMED (DDSERR 140)**

CAUSE Power fail caused operation to fail.

ACTION Try command again.

---

143 ***** AN ERROR WAS ENCOUNTERED WHEN INITIALIZING SECTIONS AND STEPS (DDSERR)**

CAUSE An internal error was encountered.

ACTION Notify your support engineer..

200	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO SEND/RECEIVE INFORMATION FROM THE USER (DDSERR 200)**
CAUSE	An internal error was encountered.
ACTION	Notify your support engineer..

201	***** AN ERROR WAS ENCOUNTERED IN ATTEMPTING TO RETRIEVE A MESSAGE FROM THE CATALOG (DDSERR 201)**
CAUSE	A corresponding message is not included in the catalog.
ACTION	This will not effect the execution of the diagnostic. However, your support engineer should be notified.

202	***** AN ERROR WAS ENCOUNTERED IN ATTEMPTING TO CONVERT A NUMBER TO A STRING (DDSERR 202)**
CAUSE	This as internal error in the diagnostic.
ACTION	Notify support engineer so it may be resolved.

203	***** AN ERROR WAS ENCOUNTERED IN ATTEMPTING A BIT EXTRACTION OPERATION (DDSERR 203)**
CAUSE	This as internal error in the diagnostic.
ACTION	Notify support engineer so it may be resolved.

207	***** DUE TO PROBLEMS WITH INQUIRY, SCSIDDS CANNOT FUNCTION PROPERLY AND WILL THEREFORE TERMINATE (DDSERR 207)**
CAUSE	Inquiry command cannot identify device.
ACTION	Notify support engineer for help.

209	***** YOUR RESPONSE WAS INVALID (DDSERR 209)**
CAUSE	Diagnostic did not expect your response.
ACTION	Try again with valid response.

210	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO OBTAIN DATA FROM AN I/O BUFFER (DDSERR 210)**
CAUSE	This is an internal error between the diagnostic and the device access routine.
ACTION	Notify your support engineer.

211	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO PLACE DATA INTO AN I/O BUFFER (DDSERR 211)**
CAUSE	This is an internal error between the diagnostic and the device access routine.
ACTION	Notify your support engineer.

212	***** AN ERROR WAS ENCOUNTERED WHILE ATTEMPTING TO OBTAIN AN I/O BUFFER (DDSERR 212)**
CAUSE	This is an internal diagnostic error.
ACTION	Notify your support engineer.

217	***** WARNING -- EXPECTED ! BYTES OF FIRMWARE REVISION DATA, BUT RECEIVED ONLY ! BYTES. (DDSERR 217)**
CAUSE	Drive did not return the proper number of bytes for firmware revision number.
ACTION	Internal error. Notify your support engineer.

218	***** NO MORE FIRMWARE REVISION DATA**
CAUSE	Drive returned no firmware revision data.
ACTION	Internal error. Notify your support engineer.

219	***** RECEIVED CONTROLLER ID = ! (!)** **EXPECTED CONTROLLER ID = ! (!)**
CAUSE	Drive returned the incorrect controller id.
ACTION	Internal error. Notify your support engineer.

Contents

7

ISL Diagnostics Overview

Introduction

The offline diagnostics system provides a means of testing System Processor Unit (SPU) hardware Field Replaceable Units (FRUs) and interrogating low-level hardware register contents. It includes a standard operating environment complete with a library of common procedures, program macros, and command set/feature functionality. The offline diagnostics and utilities are implemented via the Support Tape. This tape is available in either open reel or cartridge format.

The purpose of the offline diagnostics and utilities is to provide intelligent computer-based troubleshooting support in the absence of a functional operating system. The capabilities of the offline diagnostics system support two kinds of users. They are:

- Field Support Personnel (IIP & OEM) who are troubleshooting SPU failures. These users will generally use the "command-message" interface.
- Expert Support Personnel engaged in low-level hardware debugging activity. These users are more likely to use the low-level "binary" command interface. More expert levels of use also require detailed knowledge of both system internals (Initial System Loader & Operating System) and Processor-I/O Dependent Hardware code for any given SPU under investigation.

All chapters provide operating instructions, detailed test/function/command descriptions and example sessions. Error message information is incorporated into test descriptions or included at the end of the chapter.

The available diagnostic test is:

Name	Systems	Description
IOMAP	all PA-RISC systems	Input/Output Map Utility

All offline diagnostics are loaded and called by ISL. They conform to the ISL subset of the Object Module specification. This subset removes portions of the Object Module header, thereby reducing the code space requirements.

Offline Diagnostics System Requirements

In order to support the offline diagnostics system, an HP Precision Architecture computer must be able to boot the Initial System Load (ISL) environment and provide all of its functionality. User access to at least one functioning terminal is also required. The following events and functions must occur to meet these requirements:

- Processor Dependent Code (PDC) must successfully execute.
- PDC must boot and transfer control to the ISL environment.
- Offline diagnostic program code must be available in Logical Interface Format (LIF) files.

Boot Files - Logical Interchange Format (LIF)

System boot files are stored on tape or disk as Logical Interchange Format (LIF) files. The purpose of the LIF scheme is to ensure and facilitate file transfer and compatability among all HP computer systems. It is a Hewlett Packard standard for file construction on any bootable media. The standardization of file format for the pre-operating system boot environment guarantees that one ISL based offline diagnostics system environment will be loadable on all HP Precision Architecture RISC computer systems.

The LIF files are stored on the lowest blocks of the boot media, in the order listed below.

LIF File Volume Label

LIF file to the Processor Dependent Code (PDC).

Nou-LIF File Label

loading and initializing the operating system.

LIF File Directory

load utilities.

Initial Program Load (IPL) code

program software. This program provides a primitive pre-operating system environment that allows specification and manipulation of the primary boot, alternate boot, and console paths. It permits system boot options such as steering the SPU into either an ISL Offline based (Single User) utility/program or an HP-UX operating system (Multiple User) environment.

One or more LIF Data/Program files

located and loaded by the ISL program as selected by the user or a LIF directory autoboot file resident in non-volatile memory.

Loading the ISL Environment

PDC implements all processor dependent functions, such as initialization and selftest. After successfully completing selftest, PDC retrieves the LIF File Directory and begins the bootstrap process. In order to load ISL, PDC must know the particular device on which ISL resides. Typically, the device occupies the primary boot path that is maintained by PDC.

After loading the (IPL) file, PDC then loads this program, which boots the system to the ISL level. At the ISL level, the boot process can be broken to run the offline diagnostics.

Purpose of ISL Environment

The Initial System Loader (ISL) implements the operating system independent portion of the bootstrap process. It provides an intermediate stopping point in the boot process where the operator can modify the boot path or run an offline diagnostic.

All PA-RISC SPUs contain special purpose memory for maintaining critical configuration related parameters (e.g., primary boot, alternate boot, and console paths). Two forms of memory are supported: Stable Storage and Non-Volatile Memory.

Typically, when control is transferred to ISL, an autoboot sequence takes place. Typically, ISL executes commands from the **autoexec** file in a script-like fashion. During autoboot, ISL displays its revision and the name of any utility it executes. However, when autoboot is disabled the user will be prompted for input after ISL displays its revision. Acceptable input is any ISL command name or the name of any utility available in the LIF directory. If a non-fatal error occurs or the executed utility returns, ISL again prompts the user for input.

Offline Diagnostics System Components

The offline diagnostics system is composed of the User Interface (UI) and diagnostic programs. Because they run from the ISL environment rather than from the operating system, the system is unavailable for normal use. The user interface (UI) operates as an integral part of the three diagnostic programs, SPU Processor (`A1002AP/A1100AP`), SPU Memory (`A1002AM/A1100AM`), and SPU I/O (`A1002AI/A1100AI`). This user interface cannot be accessed directly by the user but functions automatically whenever any of the above programs are invoked. The offline utility program `IOMAP` contains its own user interface.

User Interface

The User Interface (UI) is the communication link between the user and the various diagnostic programs. It sends messages to the user from diagnostic programs, and returning user replies. The offline diagnostics system interface provides two levels of access, Command-Message and Binary.

Diagnostic Programs

The diagnostic programs are a comprehensive set of software to test FRUs for Processor, Memory and I/O functionality on PA-RISC SPUs. These diagnostics determine which of the field replaceable units (FRUs) need replacement.

Utility Programs

Offline utility programs cannot isolate defective FRUs, but can verify which functions of a device are operating correctly. Input/Output Map (`IOMAP`) helps determine the cause of device failure by providing stress simulation and diagnostic information.

This utility can provide identification/loopback verification of the internal I/O modules and in some cases invoke Selftests for I/O and peripheral devices. System exercisers provide a means of using or maximally loading a particular part of the system.

This utility has its own user interface which is both interactive and command driven.

Program Construction

Each diagnostic consists of sections and steps. Sections are the smallest portion of a diagnostic which can be repeated and expected to return "no errors" on known good hardware. A step generally tests a specific circuit for one or more particular failures. The steps in a section are generally grouped by the hardware tested.

Support Tape Implementation

The ISL-based offline diagnostics and utilities are implemented via the Support Tape on either an open reel (P/N 92454-13503) or cartridge tape (P/N 92452-13303) format.

Defect and Enhancement Requests

Submit defect reports and enhancement requests through the STARS database. Refer to the appropriate product numbers, listed below.

Product Number	Test Name	Product Name
30344-10001	IOMAP	Input/Output Map Utility
30344-10001	ISL	Initial System Loader

Contents

8

Input/Output Map Utility (IOMAP)

Introduction

This chapter describes the IOMAP Utility and provides operating procedures for it. IOMAP displays the configuration of all devices attached to an HP Precision Architecture RISC SPU and the modules and adapters connected to it. This utility runs on both the MPE XL and HP-UX variants of these SPUs. IOMAP provides identification, selftest, and loopback tests on each component capable of such tests. IOMAP can perform the following:

- Identify the configuration of I/O components in the system.
- Quickly test all I/O components in a system.
- Test selected individual components.
- Determine the configuration of Co-processors, Analyzer Card, Cache and TLB memory sizes, ROM revisions, and switch settings.
- Display version data from board revision code EEPROMS (850/950 family only).

This chapter provides the following information about IOMAP:

- Functional Description
- User Interface Description
- Error Message Information

Defects and Enhancements

Submit defect reports and enhancement requests concerning this diagnostic through the STARS database, referencing product number 30345-10001 (IOMAP).

Minimum Configuration

The minimum hardware necessary for IOMAP to load and run consists of:

- Any HP Precision Architecture RISC computer
- Functioning boot path
- Front panel hex/LED display
- Console (required for 850/950 family SPU)

Functional Description

The IOMAP utility resides in the ISL boot directory. It can only be called from ISL, when the operating system is not running. The system is completely unavailable to the customer while IOMAP is running. IOMAP can be loaded from either a supported boot device (cold-load) tape drive, or disk drive (start device).

Most of IOMAP is written in the C programming language. A small portion of the low level code is written in assembler. Mnemonics and abbreviations used are listed in the glossary at the end of this manual.

IOMAP provides four test modes:

- **Identify** This test attempts to identify each component in every I/O path (or specified paths). The information printed includes I/O path to the component, component name, component ID number, component software model number(if applicable), firmware revision (if applicable), hardware revision (if applicable), and an indication of which test modes are available for the component. Configuration data is determined by PDC calls. Path information is obtained from PDC calls, direct I/O (DIO), and DMA transactions.
- **Loopback** This test performs component dependent loopback tests where feasible. The result of this test is reported as a pass, fail, unimplemented or untestable status.
- **Selftest** This test initiates the internal selftest of each component where feasible. The result of this test is reported as a pass, fail, unimplemented or untestable .
- **View** This test examines the version code of each board on the system (850/950 family only). The display appears on the console only, not the hex display. When running IOMAP in *silent* mode, no output appears.

Default Test Sequence

The default **IOMAP** test consists of the following:

- Display the current configuration of the processor, including the presence of Co-processor boards and Analyzer Cards, the memory sizes of Cache and TLB cards, the processor model number, and the PDC firmware revision.
- Check all possible I/O paths to determine the components present and identify them: module, bus converter, adapter, device or unit. **IOMAP** then displays a table showing all configured components.

Selftest or loopback diagnostic tests can be specified for all testable components. When specified, it is performed after mapping. Error messages are printed for any component that fails a test. The user may also limit the identify, selftest, loopback and view tests to specified path.

Limitations on Selftest and Loopback

IOMAP performs selftest and/or loopback tests on all components with those capabilities. **IOMAP** currently cannot test the following:

- Boards lacking these features (such as Channel Adapter or Parallel Card).
- Hewlett Packard Precision Bus (HP-PB) modules on 808/815 family (such as the serial controller) which are not part of the console or boot path.
- Devices attached to ports of the terminal multiplexor (such as terminals, printers and datacomm lines).

Special Test Requirements

Adhere to the HP-PB device adapter test requirements given below.

HP-PB LAN Device Adapter

Before running selftest and loopback, terminate the external link. Attach a terminated T-connector to the LAN connector of the MAU. Connect it to the currently configured MAU (internal or external).

HP-PB GPIO Device Adapter

Attach a loopback test hood before running the external loopback test.

User Interface

The user interface is divided into input and output sections. Each of these sections can operate in different modes, as specified by the user. To obtain online help for IOMAP enter "iomap help" at the ISL> prompt.

User Input

The user input interface provides a means of configuring IOMAP, allowing selective user control over IOMAP operation. Configurations and options can be specified at the IOMAP command line. This provides a fast means for an experienced user to set up IOMAP, and also allows operation to be preconfigured in an autoboot file. An interactive interface is also available which prompts the user for specific input and provides configuration instructions.

While running IOMAP, a user can enter "Break Mode", which suspends program operation. The Break Mode "debug" facility permits trained support personnel to examine and modify status and registers.

Caution

While Break Mode can allow the user to harmlessly display IOMAP internal variables, it also invokes a powerful debug facility. If inadvertently used it may hang the SPU and cause an HPMC.

Commands and Syntax

The default run command is shown below:

```
ISL>iomap <optional parameters/keywords>
```

The default test settings for the commands are:

Command	State
debug	not enabled
defaults	no (see note below)
errcount	infinite (0)
erronly	false
help	false
loop	once
noerrpause	false
path	all
silent	false
tests	identify only

Note

The defaults listed above are *only* enabled if 'defaults' is entered on the command line. If one or more commands (other than **defaults**) are input to configure specific settings, the remaining settings take on default values. If no parameter commands are entered, IOMAP automatically invokes the interactive mode.

The parameter keywords and syntax are given below. Supplying parameters causes IOMAP to bypass the interactive interface. If parameters are entered on the command line, IOMAP accepts them and sets the other parameters to the defaults.

Using the command line input to bypass the interactive dialog does not affect the printing of status or error messages on the console. To limit console output to error messages only (status and error messages appear on the hex/LED display), specify "erronly". To limit message output to the hex/LED display only, specify "silent".

IOMAP parameter keywords are listed below. The optional part of the keyword appears inside the square brackets, "[ ]," with the required part to the left. Several keywords are followed immediately by a required equals sign, "=". Optional and required parameters are as shown for each keyword. All numeric input is in decimal. All periods "." and commas "," are required if shown, unless they are contained in square brackets "[ ]" and everything within those brackets is omitted.

deb[ug]

Print a trace of program execution while **IOMAP** is running. It requires source code to understand; drives the hex/LED display and, if not in silent mode, prints on the console.

The default is debug not enabled.

def[aults]

Specifies that default values are to be used. The defaults for each keyword are shown in this section. Use of this keyword causes the interactive interface to be bypassed.

errc[ount]

Instructs **IOMAP** to abaudon execution and return to ISL after the specified number of errors are detected. The *nn* parameter is the number of errors in decimal. The '=' is required. The default is to allow infinitely many errors without returning to ISL. Note that entering the value 0 (zero) allows an infinite number of errors.

```
err[count]= nn
```

erro[nly]

Causes **IOMAP** to print only error messages. It inhibits all informational messages about running status. Erronly is overridden by silent mode, which guarantees that no error or informational messages are printed after the configuration dialog. The default is erronly false. All informational messages are normally printed.

h[elp]

Prints information about **IOMAP** on the console before returning to ISL. It gives a brief explanation of the purpose of **IOMAP**, and enough syntax to prepare the user for the non-interactive interface. The default is help false, meaning that the help messages are not normally printed.

l[oop]

Specifies the number of times that **IOMAP** should repeat the selected tests. The '=' is required; ***nn*** is decimal. The default is for **IOMAP** to perform each selected test once before returning to ISL. Note that entering the value 0 (zero) allows an infinite number of loops.

```
l[oop]= nn
```

n[oerrpause]

Instructs **IOMAP** to not suspend execution when an error occurs. A hex/LED code is displayed, a message is printed on the console (if not inhibited by silent mode), and then execution continues.

The default is noerrpause false, meaning that **IOMAP** execution normally suspends for user intervention if an error occurs.

p[ath]

Restricts **IOMAP**'s operation to a specified I/O path. **IOMAP** only performs its tests (identify, selftest, loopback and view) on components in the specified I/O path. The '=' is required.

The default is for **IOMAP** to perform specified tests on all existing I/O paths. Use the form p[ath]= n.n . . . n For example, to run tests on the HP-IB card in CIO slot 2 of module 8, the command line would be `iomap path=8.2`

r[evport]:

Instructs **IOMAP** to display the revision information of the board specified (850/950 family only). It is used in conjunction with t[est]=v[iew]. The '=' is required; *nn* is decimal. Valid revision port numbers are from 1 to 27.

Default is to display revision information of all boards (*nn* =0).

```
r[evport]=nn
```

s[ilent]

Prevents **IOMAP** from printing anything to the console after the interactive dialog. All communications from **IOMAP** to the user are with the hex/LED display.

If parameters are specified on the command line, and silent mode is used, **IOMAP** does not print anything on the console at any point in the program. No console or console path is required in this case. The default is silent false.

t[ests]

Specifies which tests are to be performed. The four tests are identify, selftest, loopback and view. View applies to the 850/950 family only. View and the other three tests are mutually exclusive. If any other test is selected, view will not be run. The "all" option may be used to specify the first three tests. The '=' is required.

The default is identify only.

```
t[ests]=[i[dentify][,s[elftest]][,l[oopback]]]
 or
t[ests]=a[ll]
 or
t[ests]=v[iew]
```

v[iew] (850/950 family only)

Displays the version code of each board on the system. This appears on the console only, not on the Hex display. When running **IOMAP** in silent mode, no output appears.

Break Mode

The user can break the program at certain points by using [Control] (C) or [Control] (Y). These user interrupts are detected after each loop completes, and after any input or error message completes.

Note

Break mode is not entered by using the console "break" key. Enter [Control] (C) several times to ensure successful execution.

Caution

The read and write commands represent a true debug facility, which can easily destroy the state of the machine and cause a High Priority Machine Check. These debug commands should only be used by someone with a detailed knowledge of IOMAP and system internals. With explicit instructions (followed exactly), these commands are useful for examining the state of the DMA data buffers.

The following message is printed when the user presses [Control] (C) or [Control] (Y) on the keyboard. It is the first indication that break mode has been entered:

```
BREAK MODE [c,d,e,h,r,s,w] (c):
```

Break mode commands are given below:

Command	Description
c	Continue - resume program execution
d	Display - displays the current configuration values.
e	Exit - terminate IOMAP and return to ISL
h	Help - print a menu of break mode commands, with a brief explanation of the purpose of each.

r	Read - reads real memory and displays the contents on the console.
	`Syntax: r <addr in hex/LED>`
	`[,<number of words to read in hex/LED>]`

s	Status - show the detailed status information for each configured device adapter.
w	Write - writes the supplied pattern to real memory at the specified address. `Syntax:w <addr in hex/LED>,<data in hex/LED>[,<number of words to write in hex/LED>`
q	same as 'e'
?	same as 'h'

Diagnostic Output

IOMAP provides status and error information to the user through the front panel hex/LED display or the console. Selecting "silent" mode routes error/status information only to the hex/LED display. Otherwise, status and error messages are output to the console in parallel with hex/LED display output.

Hex/LED Display Format (Silent Mode Only)

Hex codes for IOMAP fall within the range of CE80 - CEBF, using defined and undefined codes, as required by the HP Precision Architecture RISC Chassis I/O standard. Since this range is not sufficient to express all of the required information, the hex/LED codes are followed by descriptive "parameter" values. The "class" code (CE80 - CEBF) is displayed first, followed by the stated number of "parameter" values.

For 840 and 850/950 family SPUs, the hex display shows the "class code" for three seconds, then displays the parameter values for two seconds each, as shown below.

```
CEBA  n  n  n
 |    |  |  |
 |    |  |  type ID
 |    |  |
 |    |  module, slot or device slot number
 |    |
 |    level in the I/O tree
 |
 Initial Code
```

For any system with only four LEDs (xx2 and 825/935 families), the LED displays cycle in the following sequence:

Data	Interval
blank	3 seconds
hex digit 1	2 seconds
blank	1 second
hex digit 2	2 seconds
blank	1 second
hex digit 3	2 seconds
blank	1 second
hex digit 4	2 seconds
blank	1 second
[parameter 1]	2 seconds

(more blanks and parameters, as required)

The cycle requires at least 15 seconds, depending on the number of parameters passed by the message.

For the 808/815 SPUs, the LED display cycles in the following sequence:

Data	Interval
blank	3 seconds
hex digits 1 & 2	2 seconds
blank	1 second
hex digit 3 & 4	2 seconds
blank	1 second
[parameter 1]	2 seconds

(more blanks and parameters, as required)

The cycle requires at least 9 seconds, depending on the number of parameters passed by the message.

Note

For the 808/815, xx2, and 825/925 family SPUs, the only truly readable message information appears on the console. Disabling "silent" mode speeds up program execution by eliminating pausing. This feature speeds up access to error information whenever the console is available.

Hex error codes are defined in the "Error Message Information" listing under the "Output to Hex Display and Console" category.

Hex/LED Display Output

When `IOMAP` execution begins, the display shows the running code "CE81". As each test executes, the current loop count appears. If an error occurs, the error described in the previous section appear.

If `IOMAP` is in silent mode, the map data appears as follows:

1. The code CEBA appears in the hex/LED display.
2. A number indicating the depth of the component in the I/O tree appears:

   (up to 6 levels of bus converters are permitted),

   device adapter
3. The component module, device or unit number appears.
4. The type ID for the component appears.

Any errors are displayed when detected and then the program continues.

For example, if the following is the console display for a 840 system using default IOMAP settings,

```
I/O Configuration:

                                                        Type SW   Revisions Tests
Path         Component Name                              ID  Mod  Hdwr Firm Avail
------------ ----------------------------------  ---- ---- ---- ---- -----
8            CIO Channel Adapter                   8H  10H   0    0
8.0              HP-IB card                        2H   -    0    0 ST LB
8.0.3                7914 Disc Drive              20bH   -    0    0
8.0.3.0                 Hard Disc Unit             0H   -    0    0
8.0.3.1                 Cartridge Tape             0H   -    0    0
8.2              HP-IB card                        2H   -    0    0 ST LB
8.2.0                2608A Dot Matrix Printer     001H   -    0    0
8.2.1                7970E Mag Tape Controller    183H   -    0    0
```

then the following would be displayed in the hex display.

```
CEBA 1 8 8
CEBA 7 0 2
CEBA 8 3 20B
CEBA 9 0 0
CEBA 9 1 0
CEBA 7 2 2
CEBA 8 0 2001
CEBA 8 1 183
CEBA 7 3 3
```

If the following is a typical display for 850/950 family SPUs,

```
I/O Configuration:

                                                        Type SW   Revisions Tests
Path         Component Name                              ID  Mod  Hdwr Firm Avail
------------ ---------------------------------- ---- ---- ---- ---- -----
2            Bus Converter                         7H   CH    0    0
2/8            CIO Channel Adapter                 8H  14H    0    0
2/8.0              HP-IB card                      2H    -    0    0 ST LB
2/8.0.3              7914 Disc Drive              20bH    -    0    0
2/8.0.3.0                 Hard Disc Unit           0H    -    0    0
2/8.0.3.1                 Cartridge Tape           0H    -    0    0
2/8.2              HP-IB card                      2H    -    0    0 ST LB
2/8.2.0              2608A Dot Matrix Printer    2001H    -    0    0 ST LB
2/8.2.1              7970E Mag Tape Controller    183H    -    0    0
```

then the following would be displayed in the hex display ...

```
CEBA 1 2 7
CEBA 2 8 8
CEBA 7 0 2
CEBA 8 3 20B
CEBA 9 0 0
CEBA 9 1 0
CEBA 7 2 2
CEBA 8 0 2001
CEBA 8 1 183
```

If the following is the normal display for 808/815 family SPU,

```
I/O Configuration:
                                        Type      SW Revisions Tests
Path         Component Name               ID     Mod Hdwr Firm Avail
------------ ------------------------  ----   ---- ---- ---- -----
4            HP-PB HP-IB Card                      4H  40H    0    0 ST LB
4.1             7933H/7935H disc drive            212H    -    -    -
5            Serial Controller                     5H   EH    0    0 ST LB
                System Console is connected to this module
6            HP-HIL                                5H  14H    0    0 ST LB
8            Memory Controller                     1H   9H    0    0    LB
28           Memory Controller                     1H   8H    0    0    LB
```

then the following would be displayed in the hex/LED display ...

```
CEBA 1 4  4
CEBA 8 1  212
CEBA 1 5  5
CEBA 1 6  5
CEBA 1 8  1
CEBA 1 28 1
```

Any errors would be displayed when detected and then the I/O mapping would continue.

The four or eight LEDs will display the same hex codes in sequence.

Console Messages

The console provides information in addition to the hex/LED display. In some cases, the console messages provide more information than the hex/LED display.

IOMAP can be run without a terminal by specifying "silent mode". Disabling "silent" mode speeds up the console display. When "silent" mode is enabled, **IOMAP** always outputs error and status information to the hex/LED display.

Silent mode allows test execution without a console, or whenever console I/O may be interrupted or corrupted. Use of silent mode is discouraged, since it is slow and more difficult to use.

Example Session

This example session illustrates a typical session with console output.

Example 1. 840 Session

The following console output illustrates IOMAP functions and displays on a 840 SPU. The interface is indentical on the 825/925 family and 808/815 family SPUs. Examples 2 and 3 show portions of the display specific to the 825/925 family and 808/815 family SPU.

```
ISL> iomap

IOMAP Revision A.01.00  February 21, 1989

IOMAP Running: CE81

This program has the capability to identify the configuration of
the system and its I/O paths and devices.  Many of the components of
the I/O system can be tested with selftest and loopback diagnostics.

Without changing any parameters, this program will map all existing
I/O components in the system, but will not perform any other diagnostics.

You may enter break mode by pressing control-C or control-Y.

You may press control-X to erase what you just typed in.


Do you wish to modify any program parameters? [y,n] (n):  y

Change tests to be executed? [y,n] (n):  y

    Identify? [y,n] (y):  y

    Selftest? [y,n] (n):  y

    Loopback? [y,n] (n):  y

Test single path only? [y,n] (n):  n

Enter the number of LOOPS [<n>,0=infinite] (1):

Change miscellaneous program parameters? [y,n] (n):  y

    Pause on errors? [y,n] (y):

    Maximum error count before returning to ISL (infinite):

    Suppress all messages? [y,n] (n):

    Print error messages only? [y,n] (n):
```

```
    Display software debug messages? [y,n] (n):


Identify: Loop 1:  All I/O components are being identified...

Processor Identification:
    Hardware Model:  4H (9740A), Revision:  0
    Hardware ID:  0000102dH, Software ID:  0000102dH
    Processor Dependent Code (PDC) Revisions:
        SM - PDC Firmware:  6       IU - PDC Selftest:  1
    Processor Board Revisions:
        SM - System Monitor:    0
        RF - Register File:     0
        EU - Execution Unit:    1
        IU - Instruction Unit:  0
        TL - TLB Board:         1
        CA - Cache Unit:        6
    Cache and TLB Sizes:
        Instruction Cache:  64 K bytes, Instruction TLB:  2 K entries
        Data Cache:         64 K bytes, Data TLB:         2 K entries
    Co-processors:
        Floating Point Co-processor is installed
    Main Memory:  24 M bytes

I/O Configuration:
                                                    Type   SW Revisions Tests
Path         Component Name                          ID   Mod Hdwr Firm Avail
------------ ------------------------------------ ---- ---- ---- ---- -----
8            CIO Channel Adapter                    8H  10H    0    0
8.0              HP-IB card                         2H    -    0 2612 ST LB
8.0.0                7933/7935 disc drive         212H    -    -    -
8.0.1                7933/7935 disc drive         212H    -    -    -
8.0.2                9144 cartridge tape drive    260H    -    -    -
8.1              Console Device Adapter
8.2              HP-IB card                         2H    -    0 2612 ST LB
8.2.1                2608A dot matrix printer    2001H    -    -    - ST LB
8.2.3                7978A mag tape unit          178H    -    -    -
8.4              LAN card                           6H    -    1 2620 ST
8.6              A-LINK card                        8H    -    8 2812 ST LB
8.6.0                7937FL disc drive              -    -    -
8.6.1                7937FL disc drive              -    -    -
12           Bus Converter (Local Port)             7H   CH    0    0 ST LB
12/0             Bus Converter (Remote Port)        7H   CH    0    0 ST LB
12/24            HP-PB MUX Card                     5H   DH    0    0 ST LB
24           CIO Channel Adapter                    8H  10H    0    0
24.2             HP-IB card                         2H    -    2 2613 ST LB
24.3             HP-IB card                         2H    -    2 2613 ST LB
36           Memory Controller ( 8 M Bytes)         1H   8H    0    0    LB
44           Memory Controller ( 8 M Bytes)         1H   8H    0    0    LB
52           Memory Controller ( 8 M Bytes)         1H   8H    0    0    LB
```

```
Identify Loop 1 (1H) complete.

Selftest: Loop 1: All testable I/O components are being tested...
Processor Selftest:  Untestable
Component    Component Name                                Selftest Results
-----------  --------------------------------------------  ----------------
8            CIO Channel Adapter                           Untestable
8.0              HP-IB card                                3 Sec ST Passed
8.0.0                7933/7935 disc drive                  Unimplemented
8.0.1                7933/7935 disc drive                  Unimplemented
8.0.2                9144 cartridge tape drive             Unimplemented
8.1              Console Device Adapter
8.2              HP-IB card                                3 Sec ST Passed
8.2.1                2608A dot matrix printer              3 Sec ST Passed
8.2.3                7978A mag tape unit                   Unimplemented
8.4              LAN card                                  3 Sec ST Passed
8.6              A-LINK card                               3 Sec ST Passed
8.6.0                7937FL disc drive                     Unimplemented
8.6.1                7937FL disc drive                     Unimplemented
12           Bus Converter (Local Port)                    Passed
12/0             Bus Converter (Remote Port)               Passed
12/24            HP-PB MUX Card                            Passed
24           CIO Channel Adapter                           Untestable
24.2             HP-IB card                                3 Sec ST Passed
24.3             HP-IB card                                3 Sec ST Passed
36           Memory Controller ( 8 M Bytes)                Untestable
44           Memory Controller ( 8 M Bytes)                Untestable
52           Memory Controller ( 8 M Bytes)                Untestable

Selftest Loop 1 (1H) complete.

Loopback: Loop 1: All testable I/O components are being tested...

Component    Component Name                                Loopback Results
-----------  --------------------------------------------  ----------------
8            CIO Channel Adapter                           Unimplemented
8.0              HP-IB card                                64 Bytes Passed
8.0.0                7933/7935 disc drive                  Unimplemented
8.0.1                7933/7935 disc drive                  Unimplemented
8.0.2                9144 cartridge tape drive             Unimplemented
8.1              Console Device Adapter
8.2              HP-IB card                                64 Bytes Passed
8.2.1                2608A dot matrix printer              64 Bytes Passed
8.2.3                7978 mag tape unit                    Unimplemented
8.4              LAN card                                  Unimplemented
8.6              A-LINK card                               256 Bytes Passed
8.6.0                7937FL disc drive                     Unimplemented
8.6.1                7937FL disc drive                     Unimplemented
12           Bus Converter (Local Port)                    Passed
12/0             Bus Converter (Remote Port)               Passed
```

```
12/24           HP-PB MUX Card                              Passed
24          CIO Channel Adapter                             Unimplemented
24.2            HP-IB card                                  64 Bytes Passed
24.3            HP-IB card                                  64 Bytes Passed
36          Memory Controller ( 8 M Bytes)                  Passed
44          Memory Controller ( 8 M Bytes)                  Passed
52          Memory Controller ( 8 M Bytes)                  Passed

Loopback Loop 1 (1H) complete.

Do you want to exit this program and return to ISL? [y,n] (n):  n

Do you wish to modify any program parameters? [y,n] (n):  y

Change tests to be executed? [y,n] (n): y

    Identify? [y,n] (y): n

    Selftest? [y,n] (n): y

    Loopback? [y,n] (n): y

Test single path only? [y,n] (n):  y

    Enter I/O path <n.n...n>: 8.4

Enter the number of LOOPS [n,0=infinite] (1):

Change miscellaneous program parameters? [y,n] (n):

Selftest:  Loop 1:  Only the component at 8.4 is being tested...
Component   Component Name                                  Selftest Results
----------- ----------------------------------------------  ----------------
8           CIO Channel Adapter                             Untestable
8.4             LAN card                                    15 Sec ST Passed

Selftest Loop 1 (1H) complete.


Loopback:  Loop 1:  Only the component at 8.4 is being tested...

Component   Component Name                                  Loopback Results
----------- ----------------------------------------------  ----------------
8           CIO Channel Adapter                             Unimplemented
8.4             LAN card                                    Unimplemented

Loopback Loop 1 (1H) complete.

Do you want to exit this program and return to ISL? [y,n] (n):  y

IOMAP Exiting.
```

Example 2. Processor Identification Display (825/925 family)

The 825/925 family display resembles the 840 except for the processor identification. The display appears as follows:

```
Processor Identification:
    Hardware Model:  8H (A1002A), Revision:  0
    Hardware ID:  0000303bH, Software ID:  0000303bH
    Processor Board Revisions:
      CPU - CPU Chip:                      2
      SIU - System Interface Unit:         1
      CCU - Cache Control Unit:            2
      TCU - TLB Control Unit:              2
      MIU - Math Interface Unit:           2
      PDH - Processor Dependent Hardware:  1
      PDC - Processor Dependent Code:      1
    Cache and TLB Sizes:
        Instruction Cache:  16 K bytes, Instruction TLB:  1 K entries
        Data Cache:         16 K bytes, Data TLB:         1 K entries
    Co-processors:
        Floating Point Co-processor is installed
    Main Memory:  32 M bytes
```

Example 3. Processor Identification Display (808/815 family)

The 808/815 family display resembles 840 except for the processor identification. The display appears as follows:

```
Processor Identification:
    Hardware Model:  100H (A1408A), Revision:  0
    Hardware ID:  00000000H, Software ID:  00000000H
    Processor Board Revisions:
         ROM Component Revision:       0
    Cache and TLB Sizes:
        Instruction Cache:  256 bytes, Instruction TLB:  32 entries
        Data Cache:           0 bytes, Data TLB:         32 entries
    Co-processors:
        Floating Point Co-processor is installed
    Main Memory:  4 M bytes
```

Note

If a LAN card fails selftest or loopback test, the error code returned from the test will be displayed immediately below the "*** FAILED ***" message. To determine what the error is and what causes it, please run the Local Area Network Device Adapter Diagnostic (**LANDAD**) within the On-Line Diagnostic System.

The following message is printed after an error occurs, to prompt the user for input:

```
An error was detected; do you want to continue? [y,n] (y):
```

The program suspends until the user responds affirmatively. The *noerrpause* mode inhibits this message and the suspend condition. A response of *n* will disable prompting on errors associated with the current loop and cause the program to exit; a response of *y* will continue program execution.

Example 4. Processor and I/O Displays (850/950 family)

The 850/950 family display differs from the 840 since it shows bus converter paths and can display board revision levels. The processor identification appears as follows:

```
ISL> iomap

IOMAP Revision A.01.00  February 21, 1989

IOMAP Running: CE81

This program has the capability to identify the configuration of
the system and its I/O paths and devices.  Many of the components of
the I/O system can be tested with selftest and loopback diagnostics.

The program can also be used to examine the revision information of each
board (revision port) on the A1100A system.

Without changing any parameters, this program will map all existing
I/O components in the system, but will not perform any other diagnostics.

You may enter break mode by pressing control-C or control-Y.

You may press control-X to erase what you just typed in.


Do you wish to modify any program parameters? [y,n] (n):  y

Do you want to read revision port information? [y,n] (n): y

Which revport? [d=done,0=all,<1..27>] (0): 1

Slot: 1  Board: PDH Board

Addr   Information       Bits   Data (Hex unless otherwise indicates)
0x00   Assembly Number   00-39  3019060002 (BCD)
0x05   Slot Number       00-07  1
0x06   Date Code         00-15  2752 (BCD)
0x08   Hardware ID       00-31  0000000000 (BCD)
0x0C   SMR Serial Number 00-47  CCCCCC (ASCII)
0x12   Division Number   00-15  47 (ASCII)
0x14   Boot ID           00-31  00000000
0x18   Software ID       00-31  00000000
0x1C   Reserved for PDH  00-31  00000000

Next revport? [y,n] (y): n

Which revport? [d=done,0=all,<1..27>] (0): 9

Slot: 9  Board: Mem Array #7
```

```
Slot selected is empty.

Next revport? [y,n] (y): n

Which revport? [d=done,0=all,<1..27>] (0): 27

Slot: 27  Board: BC-Y (SMB slot #3)

Addr   Information       Bits   Data (Hex unless otherwise indicates)
0x00   Assembly Number   00-39  3019060030 (BCD)
0x05   Slot Number       00-07  1B
0x06   Date Code         00-15  2752 (BCD)
0x08   Hardware ID       00-31  0000000000 (BCD)
0x0C   SMR Serial Number 00-47  CCCCCC (ASCII)
0x12   Division Number   00-15  47 (ASCII)
0x14   Boot ID           00-31  00000000
0x18   Software ID       00-31  00000000
0x1C   Reserved for PDH  00-31  00000000

End of revports.

Which revport? [d=done,0=all,<1..27>] (0): d

Do you want to exit this program and return to ISL? [y,n] (n): y

IOMAP Exiting.
```

Hex/LED Display Output Interface

Hex codes are displayed in the front panel hex/LED display at different times during the execution of **IOMAP**. Hex code must be within the range of CE80 - CEBF, using defined and undefined codes, to meet the specifications in the Chassis I/O Architecture document. This is not sufficient to express all of the desired information, so they are followed by descriptive *parameter* values. The *class* code (CE80 - CEBF) is displayed for three seconds, followed by the stated number of *parameter* values displayed for two seconds each.

This section gives the text and meaning of each error message. Each error message is represented by a pair of hex codes. The first one is error class number and the second one is the error code within the error class. There are five classes of errors: input, Mid-bus, HP-PB, DA and other (execution) errors. Within each classes, there are different error codes. Because different machine has different hardware configuration, not all error messages are applicable to all machines. For example, the 815 does not have channel adapters. Therefore, any reference to channel adapter in the error messages will not apply to 815. In this section, "[]" is used to place machine dependent display. For example, bus converter number is placed in "[]" indicating that it is displayed on a system that has a bus converter. Each console message has a corresponding output to the hex/LED display.

Input Error CE90

HEX `Error ce90 cea0`

CONSOLE `*** Expected '=' after "kkk" keyword`

CAUSE An equals sign '=' must follow immediately after the *kkk* keyword. Keyword *kkk* represents one of the following command line keywords: *errcount*, *loop*, *path*, or *tests*.

HEX `Error ce90 cea1`

CONSOLE `*** Expected number after "kkk="`

CAUSE A decimal number must follow after the equals sign '=', following keyword *kkk* on the command line. Keyword *kkk* represents one of the following command line keywords: *errcount*, *loop*, *path*, or *tests*.

HEX `Error ce90 cea2`

CONSOLE `*** Unrecognized keyword "kkk"`

CAUSE This keyword, which was entered on the command line interface, does not match a valid keyword.

HEX `Error ce90 cea3` *<ASCII value of 'c'>?*

CONSOLE `*** Expected 'c' as delimiter`

CAUSE The command line interface expected *c* as a delimiter, but another character was supplied.

HEX `Error ce90 cea4`

CONSOLE `*** Syntax error`

CAUSE This error can occur in many places. It means that the user data entered was not in a correct or a recognizable form. Correct syntax is given in prompt.

HEX	**Error ce90 cea5**
CONSOLE	***** Number cannot be negative**
CAUSE	The loop count and error count cannot be entered as negative numbers.

HEX	**Error ce90 cea6 [bc] ca da**
CONSOLE	***** [bc/]ca.da are reserved for the console**
CAUSE	Any attempt to configure console will cause this message.

HEX	**Error ce90 cea7** *n*
CONSOLE	***** Invalid module number** *n*
CAUSE	This means that a Precision Architecture (Spectrum) module number outside the range 1 - 63 was entered. *n* is the module number supplied by the user.

HEX	**Error ce90 cea8 [bc] n**
CONSOLE	***** Module [bc/] n does not exist**
CAUSE	This means that a Spectrum (PA) module number is valid but the corresponding module is not installed in the system.

HEX	**Error ce90 cea9 [bc/] ca**
CONSOLE	***** Mid_bus module [bc/] ca is not a Channel Adapter**
CAUSE	The Mid_bus module number, *ca*, is valid but corresponds to a module other than a channel adapter.

HEX `Error ce90 ceaa n`

CONSOLE `*** Invalid Device Adapter number n`

CAUSE The Device Adapter number entered by the user was not between 0 and 15, inclusive.

HEX `Error ce90 ceab`

CONSOLE `*** Number out of range`

CAUSE The number entered is too large or too small to be accepted. The value must be within the range specified by the prompt.

HEX `Error ce90 ceac [bc] ca da`

CONSOLE `*** On [BC bc,] CA ca, Device Adapter da does not exist`

CAUSE The specified Device Adapter number does not correspond with a DA installed on the specified Channel Adapter.

HEX `Error ce90 cead [bc] ca da`

CONSOLE `*** On [BC bc], CA ca, Device Adapter da is not an HP-IB DA`

CAUSE The Device Adapter specified does not correspond to an HP-IB DA. The number may correspond to another type of device adapter.

HEX `Error ce90 ceae`

CONSOLE `*** Unrecognized TESTS keyword kkk`

CAUSE This keyword entered after *tests=* is not *all* or a comma separated list of *identify*, *selftest*, or *loopback*. This error occurs in the command line interface only.

HEX	`Error ce90 ceaf`
CONSOLE	`*** Revport information is not available on this machine.`
CAUSE	The machine does not have Revision Code EEPROMS and therefore the information is not available.

HEX	`Error ce90 ceb0`
CONSOLE	`*** No tests specified`
CAUSE	At least one test mode (Identify, Selftest, or Loopback) must be specified for `IOMAP` to perform any meaningful testing.

HEX	`Error ce90 ceb1 n[/n]/n`
CONSOLE	`*** Module n[/n]/n does not exist`
CAUSE	This means that the module number is valid but the corresponding module is not installed in the system.

HEX	`Error ce90 ceb2 n`
CONSOLE	`*** Module n does not exist`
CAUSE	This means that the SMB module number is valid but the corresponding module is not installed in the system. This error message applies to CHEETAH only.

HEX	`Error ce90 ceb3 n`
CONSOLE	`*** Module n does not exist`
CAUSE	This means that the VSC module number is valid but the corresponding module is not installed in the system.

HEX `Error ce90 ceb4 n n`

CONSOLE `*** Module n/n does not exist`

CAUSE This means that the VSC adapter slot is valid but the corresponding slot is empty.

HEX `Error ce90 ceb5 n n n`

CONSOLE `*** Module n/n/n does not exist`

CAUSE This means that the VSC adapter module is valid but the corresponding module is not installed in the system.

Mid-bus Error CE93

HEX	`Error ce93 cea0 [bc] ca stat`
CONSOLE	`*** On [BC bc], Channel Adapter ca module error: Status = stat`
CAUSE	The specified Channel Adapter has had a module error. The contents of the module's status register is displayed as *stat*.

HEX	`Error ce93 cea1 [bc] ca stat`
CONSOLE	`*** [On BC bc, ] Channel Adapter ca timeout: Status = stat`
CAUSE	The CA io_stat register ca_ready bit did not become set (ready) with a reasonable interval (currently one second). This prevented a DMA transfer from being started by software.

HEX	`Error ce93 cea2 mod stat`
CONSOLE	`*** PDC_IODC failed while reading ENTRY_TEST: Module = mod, Status = stat`
CAUSE	An error was returned from the PDC call PDC_IODC while attempting to read ENTRY_TEST IODC from the card at the specified module.

HEX	`Error ce93 cea3 mod stat`
CONSOLE	`*** PDC_IODC failed while reading ENTRY_INIT: Module = mod, Status = stat`
CAUSE	An error was returned from the PDC call PDC_IODC while attempting to read ENTRY_INIT IODC from the card at the specified module.

HEX	`Error ce93 cea4 mod stat`
CONSOLE	`*** Error executing ENTRY_INIT on Module mod: Status = stat`
CAUSE	An error occurred while attempting to execute the ENTRY_INIT IODC on the specified module.

HEX	`Error ce93 cea5 mod`
CONSOLE	`*** SCSI card not 'Ready for Command': Module = mod`
CAUSE	SCSI card at module mod is reporting that it is not ready for a command. This error is usually a result of the SCSI card not accepting the Inquiry command.

HEX	`Error ce93 cea6 mod`
CONSOLE	`*** Unexpected Bus Service Request on SCSI card at Module mod`
CAUSE	This error is a result of a SCSI device unexpectantly asserting the Bus Service Request line during the Inquiry command.

HEX	`Error ce93 cea7 mod`
CONSOLE	`*** Inquiry command to SCSI card at Module mod did not complete`
CAUSE	SCSI card at module mod was unable to complete the Inquiry command.

HEX	`Error ce93 cea8 mod stat`
CONSOLE	`*** Fatal error returned from SCSI card at Module mod: Status =` *stat*
CAUSE	SCSI card at module mod reported a fatal error. Most likely the result of being unable to execute a SCSI Inquiry command.

HEX	`Error ce93 cea9 mod stat`
CONSOLE	`*** Bad status returned from SCSI card at Module mod: Status =` *stat*
CAUSE	SCSI card at module mod returned a bad status value of stat. This most likely occurred while attempting to execute a SCSI Inquiry command.

DA Error CE95

HEX	`Error ce95 cea0 [bc] ca da`
CONSOLE	`*** Device Adapter 'SLOW' switch (S1-7) must be down`
CAUSE	The Device Adapter speed switch S1-7 must be 'down'(toward the board surface). DIP switch S1 is at the front edge of the board.

HEX	`Error ce95 cea1 [bc] ca da`
CONSOLE	`*** Device Adapter 'SCTL' switch (S1-6) must be up`
CAUSE	The 'SCTL' (system controller) switch must be set 'up' (away from the board surface). 'SCTL' is the DIP switch on the board front edge.

HEX	`Error ce95 cea2 [bc] ca da`
CONSOLE	`*** Device Adapter load resistor pack is missing`
CAUSE	the resistor pack must be in place for the HP-IB card to operate at the maximum speed.

HEX	`Error ce95 cea3 [bc] ca da`
CONSOLE	`*** On [BC bc,] CA ca, Device Adapter da failed selftest`
CAUSE	The DA Sense Register PST bit is not set, indicating that the DA selftest failed last time it was run.

HEX	`Error ce95 cea4 [bc] ca da`
CONSOLE	`*** On [BC bc,] CA ca, Device Adapter da is not 'Ready For Command'`
CAUSE	The DA Sense Register RFC bit is not set, meaning that the DA is not able to accept another command. It should be ready at this time.

HEX	**Error ce95 cea5 [bc] ca da**
CONSOLE	***** On [BC bc,] CA ca, Device Adapter da timeout error**
CAUSE	A DMA transfer initiated for this DA failed to complete within a reasonable time period.

HEX	**Error ce95 cea6 [bc] ca da stat**
CONSOLE	***** On [BC bc,] CA ca, Device Adapter da status error: Status =** *stat*
CAUSE	The Device Adapter status does not match the expected value, after a DMA transfer was performed.

HEX	**Error ce95 cea7 [bc] ca n stat**
CONSOLE	***** On [BC bc,] CA ca, subchannel n not ready: Status =** *stat*
CAUSE	On the CA, this subchannel's status register subc_ready bit did not come ready within a reasonable period of time (currently one second). This prevented a DMA transfer from being started by software.

HEX	**Error ce95 cea8 [bc] ca da stat**
CONSOLE	***** On [BC bc,] CA ca, Subchannel da ARQ timeout: Status =** *stat*
CAUSE	The one second timeout expired before ARQ was seen in the DA sense register.

HEX	**Error ce95 cea9 [bc] ca da stat**
CONSOLE	***** On [BC bc,] CA ca, Subchannel da was not destroyed: Status =** *stat*
CAUSE	After ARQ was seen in the DA sense register, the attempt to Destroy SubChannel on the Device Adapter by Direct I/O failed (meaning that no SubChannel Destroyed message was received back from the DA).

HEX `Error ce95 ceaa da`

CONSOLE `*** Unable to reset SCSI bus on Device Adapter da`

CAUSE An attempt to reset the SCSI bus on device adapter da failed. The SCSI bus is reset before the first Inquiry command is issued.

HEX `Error ce95 ceab da`

CONSOLE `*** Inquiry command to SCSI card on Device Adapter da did not complete`

CAUSE An Inquiry command to the SCSI card at device adapter da failed to complete without error.

Other Execution Error CE96

HEX `Error ce96 cea0 index stat`

CONSOLE `*** Error in call to Processor Dependent Code (PDC): Index = index , Status = stat`

CAUSE A call to PDC routine with index index returned a bad status. It is possible that printing this line may not be successful, since output to the console requires repeated PDC calls.

HEX `Error ce96 cea1`

CONSOLE `*** Fatal error--internal buffer(s) overwritten`

CAUSE **IOMAP** has suffered an internal data management problem. If it is not a bad copy of **IOMAP**, then there may be a software bug or hardware problem.

HEX `Error ce96 cea2 n`

CONSOLE `*** Maximum error count (n) exceeded; program will abort.`

CAUSE The maximum error count specified n has been exceeded. Errors associated with the current loop will be displayed and then the program will return to ISL.

HEX `Error ce96 cea7`

CONSOLE `*** Component failed selftest`

CAUSE The current component (last one displayed) failed selftest.

HEX `Error ce96 cea8`

CONSOLE `*** Component failed loopback test`

CAUSE The current component (last one displayed) failed loopback.
