NCR UNIX SVR4 MP-RAS

In-Service Diagnostics and Basic Software Maintenance

3.02.00

B003-0126-A000
12/97

It is the policy of NCR Corporation (NCR) to improve products as new technology, components, software, and firmware become available. NCR, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult with your NCR representative or NCR office for information that is applicable and current.

To maintain the quality of our publications, we need your comments on the accuracy, clarity, organization, and value of this book.

Address correspondence to:

Systems Information
NCR Corporation
3325 Platt Springs Road
West Columbia, SC 29170

Table of Contents

Chapter 2
Device Names

Chapter 3
Invoke Applications

Chapter 4
Error Log Services

Appendix A

Application Integration

Preface

Who Should Read This Book

This book is for the following people:

- Remote Support Analyst - the person who is trained to provide over-the-phone technical assistance and support in resolving system hardware and software problems.
- Field Engineer - the person who is trained to provide on-site system maintenance to identify and resolve system hardware problems (**odmadmin** or **odmusr** logins).
- System Administrator - the person who administers the system operation through either system administrator menus or operating system commands at the shell prompt.

Releases Covered In This Book

This book applies to Release 3.02 of the NCR UNIX SVR4 MP-RAS operating system.

What You Should Know

Before using this book, you should be familiar with the following,

- Be familiar with the NCR UNIX SVR4 MP-RAS operating system. For more information about the operating system, read the *User Guide: Introduction to UNIX SVR4.*

- The types of tasks a system administrator is required to perform. Read the *NCR UNIX SVR4 MP-RAS Administrator Guide: OA&M Menu Interface* and *Command Line Interface* manuals for a description of system administrator tasks.

How to Use This Book

This book describes the menus and operations that are part of the NCR UNIX SVR4 operating system package called **odm**.

This book tells how to use the **odm** menus to perform the following operations:

- Invoke Non-Integrated Applications - describes how to use **odm** menus to perform selected diagnostic and maintenance functions installed by OEM packages that ***do not*** interface to the Operations, Administration and Maintenance menu interface (**sysadm**) or the Operations Advantage/BASE menu interface (**osa**).
- Invoke Integrated Applications - describes how to use **odm** menus to perform selected maintenance functions available via **sysadm** or **osa**, such as Micro Channel Configuration.
- System Reporting - describes how to use **odm** menus to display and clear the system error log and how to check system base files and file boundaries.
- System Services - describes how to use **odm** menus to perform data capture and tally operations.
- Diagnostic Device Testing - describes how to use **odm** menus to perform diagnostic testing of specific types of system devices, Local Area Networks (LANs) and Wide Area Networks (WANs).

Conventions Used in This Book

The following conventions are used in this book:

- The following type identifies text that you must enter exactly as shown:
  `login su`

- The following type identifies output from a screen or command:
  `Press the Return key to continue.`

- Path names and file names appear in italics. For example:

  The */etc/profile* file defines the standard environment for all users.

- Utilities, commands, user names, package names and terminal keys appear in boldface type. For example:

  The **cpio** command backs up files.

- Menu selections appear in brackets. For example:

  Use the <Internal Options> menu selection.

- Parameters for command line options appear in italics within the text. For example:

  Specify the class (***-cclass***) for the printer.

- Keys to be pressed on the keyboard, including function keys, appear in all capital letters. For example:

  Press ENTER.

- The numbers and letters in parentheses after a command name indicate the type of command.

  (1)
  Standard command available to users at the
  $ prompt

  (1M)
  Command available only to someone logged in as **root**

  (2)
  System call

  (3)
  Library function

The tasks in this book are presented in a format that includes some or all of the following items:

- Menu Selections - the sequence of selections that provide access to the menu you use to perform the task.

- Procedure - the actions you take in performing the task.
- Display - information that is displayed on your screen.

Related Publications

For supplementary information, refer to the following books:

- *Command Line Interface*
- *OA&M Menu Interface Administrator Guide*
- *Installing NCR UNIX SVR4 MP-RAS*
- *NCR UNIX SVR4 MP-RAS Reference Manual*
- *UNIX SVR4 Programmer's Guide Character User Interface (FMLI and ETI)* for information about creating forms and menu language interface menus.
- *Integrated LAN Driver Installation and Diagnostics*
- *NCR Micro Channel Communications Interface Adapter Installation & Configuration*

Chapter 1

Using odm Menus and Function Keys

Using odm Logins

The **odm** package provides for two login names: **odmadmin** and **odmusr**. The table below lists the differences between **odmadmin** and **odmusr**.

Table 1-1. odmadmin and odmusr

odmadmin	odmusr
Must use **odm** menus	**Must** use shell commands
May perform functions that require **root** permissions	May NOT perform functions that require **root** permissions

This book does not describe command-line operations performed by **odmusr**. Refer to the documents listed in the "Preface" for information about the use of shell commands by **odmusr**.

Note: The **odm** menus provide system administration-type functions without the need for command line procedures. These menus are created using the Forms and Menu Language Interface (FMLI) as described in the *UNIX SVR4 Programmer Guide: Character Users Interface (FMLI and ETI).*

Accessing odm Menus

You can access the **odm** menus from either the login prompt or the system prompt.

Before you can access the **odm** menus, you must correctly specify your terminal type. If the terminal type is specified incorrectly, the screen display may be garbled.

Using the Login Prompt

Use the following procedure to access the **odm** menus from the login prompt:

1. Log in as **odmadmin.**
2. Enter the password when it is requested.
3. Enter the correct terminal type. You must be using the system monitor (which is terminal type **at386**) or one of the following terminals:
   - **vt100** in vt100 mode
   - **wyse85** in vt100 mode
   - **2900** in vt100 mode
   - **2920** in 7901+ mode

Note: You can also specify the terminal type by placing an entry in the */etc/ttytype* file or by setting and exporting the TERM variable.

Using the System Prompt

Use the following procedure to access the **odm** menus from the system prompt:

1. Log in as **root.**
2. Enter the **root** password**.**
3. Enter **odm** at the system prompt**.**

Using odm Function Keys

Function Key Locations

The function keys on the system console are located at the top of the keyboard and are labeled F1 through F8. Function keys on a 2920 terminal are also located at the top of the keyboard.

The table below shows the function key locations for all terminals emulating a vt100:

Table 1-2. vt100 Function Key Locations

Function Key	vt100 Location
F1	PF1
F2	PF2
F3	PF3
F4	PF4
F5	Keypad 4
F6	Keypad 5
F7	Keypad 6
F8	Keypad ,

Function Key Definitions

Function keys may perform differently depending on the current menu frame. Each current menu frame identifies the function keys at the bottom of the screen. The eight keys defined on the screen coincide *spatially* with the eight function keys on the keyboard. For example, the leftmost function key identified at the bottom of the screen coincides with the F1 function key on the keyboard.

The possible functions are as follows:

CANCEL

Deletes the current frame and returns to the previous frame.

CHOICES

Displays possible selections (choices) that may be entered in the current field or frame.

CMD-MENU

Displays a menu with special purpose commands for controlling the **odm** session**.**

CONT

Resumes a task interrupted by a confirmation message--a query from the system asking you to verify that you want to continue the current task. When you are doing a task that may have undesirable results, you receive such a query to make sure you are aware of the possible risks.

ENTER

Selects the current menu item (the RETURN key also performs this function).

HELP

Displays text that describes what actions you may take at the current cursor position.

MARK

Lets you select multiple items within a menu. Some CHOICES menus make you use MARK even if you want to select *only one* item from that CHOICES menu.

NEXTFRM

Moves to the next frame.

PREVFRM

Moves to the previous frame.

NEXTPAGE

Pages the text up, displaying the next page of information.

PREVPAGE

Pages the text down, displaying the previous page of information.

RESET

Restores the default value to the current prompt.

SAVE

Saves the answers currently displayed, and either displays a subsequent form (if there is one) or performs the task.

Moving Through the Menus

The following sections describe how to move the cursor through the menus and how to select items from the menus.

Moving the Cursor

To move the cursor to the item you wish to select, do one of the following:

- Enter the first letter or letters (must be a unique identification) of the desired selection.

  For example, enter **ru** for the Run-To-Run flex disk test selection.
- Use the arrow keys.
- Use the space bar.

Selecting an Item

To select the item pointed to by the cursor, do one of the following:

- Press the RETURN key.
- Use the function key for ENTER.
- Use the function key for SAVE.

Exiting odm Menus

There are two ways to exit the **odm** menus**:**

- Display the Command Menu by pressing the appropriate function key, then select **exit** from the Command Menu.
- Enter **CTRL-j,** then type **exit** at the arrow prompt**.**

Note: If you enter the **odm** menus and find that your terminal type is not set correctly and the screen is garbled, you can use the **CTRL-j** method to get out of the menus.

odm Menu Examples

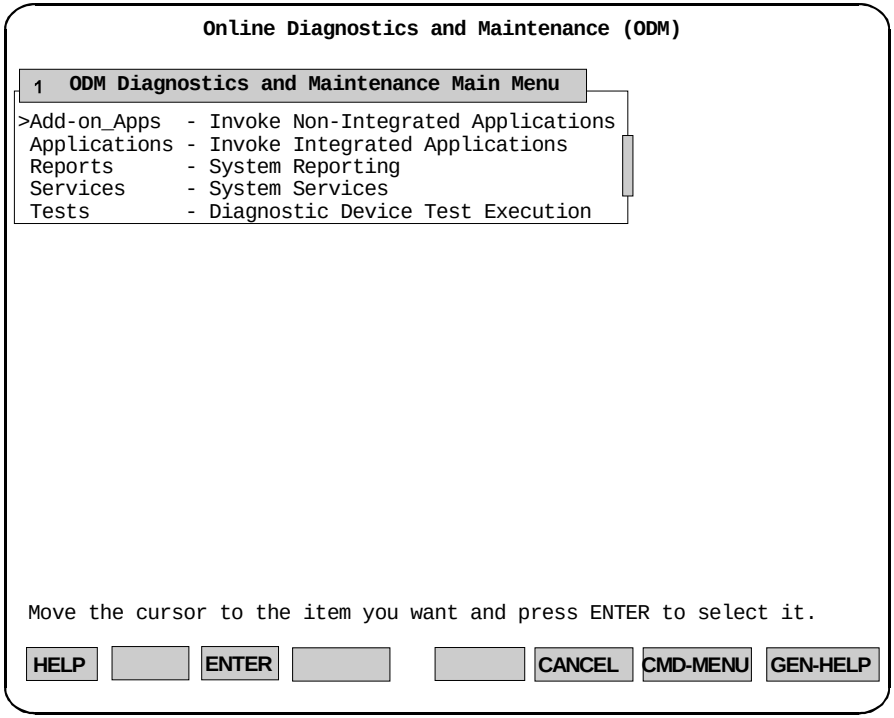
odm menu screens may display different amounts of information. This section includes examples of the following:

- The initial menu
- The command menu
- The asy device test menu (i.e., testing the built-in communications port)

The Initial Menu

Entering **odm** at either the login or system prompt takes you to the initial menu screen as shown in Figure 1-1. From this menu (or screen), you start the sequence of operations associated with Add-on Applications, Integrated Applications, System Reports, System Services, and Diagnostic Testing. Each of these menu items leads to a supporting menu from which additional selections can be made. These lower level menus are built dynamically based on installed configuration files. The menus discussed in this document are based on what is installed by default with the **odm** package. Should you have other packages installed that interface with **odm**, your menus may contain additional tasks to select. For example, if the **ild** package is installed and you select <Test>, the **ild** test function will be included in the supporting menu.

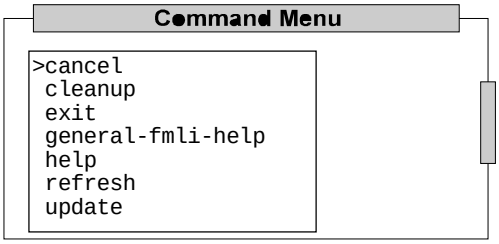
Figure 1-1. odm Initial Menu



Command Menu

Pressing the CMD-MENU function key displays the command menu as shown in the figure below.

Figure 1-2. odm Command Menu



The Command Menu selections perform the following function:

cancel

Leaves the Command Menu, closes the current frame and returns to the previous frame.

cleanup

Cancels all frames and returns to the screen at which you entered the menus.

exit

Leaves the menus and returns to the UNIX prompt.

general-fmli-help

Provides help about **fmli** itself and various **fmli** topics, such as screen layout, function keys, and **fmli** frames.

help

Provides help about the task in the frame you were using before the Command Menu.

refresh

Refreshes the screen.

update

Immediately updates the contents of the frame displayed prior to the Command Menu so that it reflects any changes made on the system since you accessed that frame.

Example Menus for Testing asy Devices

The following sequence of selections produces the "asy Device Test Execution" form shown in Figure 1-3:

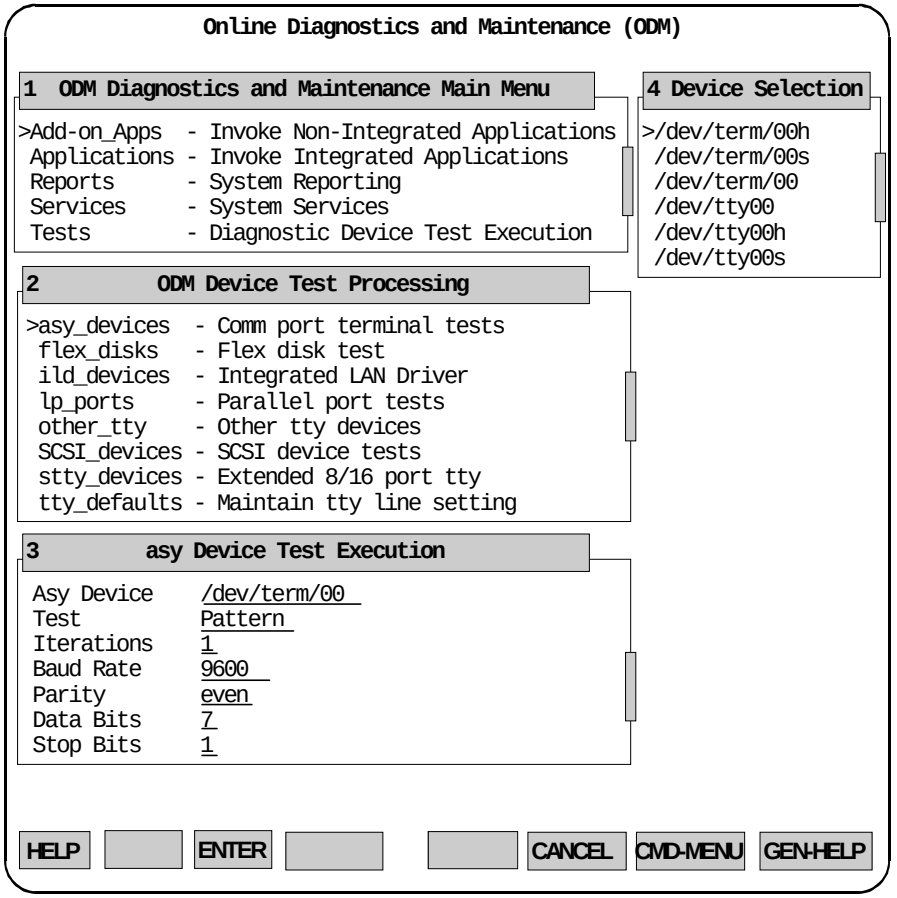
1. Select <Tests> from the initial menu display**.**
2. Select <asy_devices> from the ODM Device Test Processing menu to load the asy Device Test Execution form.

   The "asy Device Test Execution" form requires seven values to begin a test:

   a. device name
   b. test to be performed
   c. number of iterations to run the test
   d. line speed
   e. line parity
   f. number of data bits
   g. number of stop bits

   Move the cursor to the field of interest and press the CHOICES function key to display the list of choices available for each input field.
3. For the device field, pressing the CHOICES function key results in the display of a Device Selection menu. Use the arrow key(s) to move the > to the name of the **asy** device you want to test and use the MARK function key to select that asy device.
4. Use the ENTER function key (or the keyboard RETURN key) to put the asy device name into the **asy** Device field.
5. When you select the **asy** device (*/dev/term/00* in this example), the screen enables the SAVE function key. Use SAVE to activate the TTY Device Loop 3 Test form to continue the test. The "Testing Terminals" chapter (Chapter 11) describes the **asy** Diagnostics test sequences in more detail.

Figure 1-3. asy Device Test Menu Example



odm Menu Selections

This section contains tables that describe the selections available through the **odm** menus. Each table contains information about the menu selections that appear on the initial main menu: Invoke Non-Integrated Applications, Invoke Integrated Applications, System Reporting, System Services, and Diagnostic Device Test Execution.

For each task you can perform through the menus, the tables describe the sequence of menu selections and the task(s) associated with that sequence of selections.

The tables in the following sections summarize the **odm** menu selections.

Invoke Non-Integrated Applications

The **odm** package, by default, does not contain any Non-Integrated Applications. The <Add-on_Apps> menu choice, when selected, builds a submenu of diagnostic and maintenance tasks based on information found in the directory */opt/odm/add-on*. The Add-on_Apps task is provided as an access point to maintenance and diagnostics functions installed by other packages. These functions can be any executable file (i.e., shell scripts, perl scripts, executable files) that will perform the diagnostic or maintenance task selected from the submenu.

The submenu selections typically are not an integral part of either **osa** (Operations Advantage/Base), or the **sysadm** (Operations, Administration and Maintenance) administration tool. Selecting tasks from the Add-on_Apps submenu invokes and passes control to the executable file installed by the package owning the diagnostic or maintenance function.

Invoke Integrated Applications

The **odm** package, by default, does not contain any Integrated Applications. The Applications menu choice, when selected, builds a submenu of links to other diagnostic and maintenance tasks found in **sysadm** or as stand-alone functions. Two such links that might be found on this menu are the MicroChannel Configuration and Firmware Download tasks. These menu links are available via **odm** if the supporting package is installed (**madf** for the MicroChannel Configuration tools, and **fidl** for the Firmware Download). When tasks are selected via the Integrated Applications submenu, the associated form or menu is opened and control is passed to that function.

System Reporting Menu Selections

The table below summarizes the System Reporting menu selections.

Table 1-3. System Reporting Menu Selections

Select . . .	If you want to . . .
reports error report display all log date error type	Display complete records from the error log, for selected dates (all or within the last week) and selected error types (user level, kernel level, or both).
reports error report display headers log date error type	Display only the error record headers from the error log, for selected dates (all or within the last week) and selected error types (user level, kernel level, or both).
reports error report clear error log log date error type	Clear error log entries for selected dates (all or within the last week) and selected error types (user level, kernel level, or both).
reports system check check files	Check permissions and sizes of installed directories and/or files and replace the original directory/file attributes.

Select . . .	If you want to . . .
reports system check check boundary	Check for files having zero length and check for files larger than a user specified size (default size is five MB).

System Services Menu Selections

The table below summarizes the System Services menu selections.

Table 1-4. System Services Menu Selections

Select . . .	If you want to . . .
services data capture data capture driver name (i.e., asy_devices)	Perform one of the following data capture operations on the selected device(s): 1. Start or stop data capture for the device 2. Check active data capture processes and data files 3. Process a previously captured data file 4. Remove existing data capture files.
services tallies tally driver name (i.e., asy_devices)	Perform one of the following tally operations for the selected device: 1. Display or clear device tallies 2. Display current tally threshold percentage 3. Set a new tally threshold percentage
services tallies Set Monitor Interval	Set a new threshold monitor interval for all tallies.

Diagnostic Device Test Execution Menu Selections

The following table summarizes the Diagnostic Device Test Execution menu selections.

Table 1-5. Diagnostic Device Test Execution Menu Selections

Select . . .	If you want to . . .
tests driver name device name test name iterations	Perform a specific test on a specific device (flex device, parallel port, comm port, etc.)

Chapter 2

Device Names

SCSI Device Names

SCSI device names have the format **c###t#d#s#.** These devices are supported by the SCSI driver. The following explanations describe each part of the name:

c### is the controller number

may be one, two, or three digits. The most significant # identifies the Micro Channel (0 for Primary Micro Channel, 1 for Secondary Micro Channel, and so on). The center # identifies the controller to which the device is connected (0 for the controller with the lowest address, 1 for the controller with the next higher address). The least significant # identifies the channel (or bus) on the controller (0 for the bottom channel, 1 for the top channel).

Note: When device nodes are created by **mktable**, the leading zero(s) are dropped from the device name. For example, a device name **c000t0d0s0** is displayed as **c0t0d0s0**.

t# is the target device SCSI ID

= 0 through 6 (on each SCSI backplane in the system).

d# is the device Logical Unit Number (LUN)

is usually 0, but may be used differently in disk arrays.

s# is the UNIX partition or slice number for disk devices.

Each disk in the system may be divided into partitions for UNIX and/or other operating systems. Each partition in the UNIX operating system may have 16 parts, called slices. These slices are numbered 0 through f (hexadecimal numbers). Zero (0) refers to the complete UNIX partition, and 1 through f refer to individual slices (or file systems).

s# is the rewind option setting for tape devices. The following sections contain examples of SCSI device names.

SCSI Disk Device Names

The following table contains examples of SCSI fixed (hard) disk names.

Table 2-1. SCSI Fixed Disk Names (examples)

FIXED DISK NAMES	EXPLANATION OF NAME
/dev/dsk/c0t0d0s0	Disk connected to: • c0: bottom channel on the SCSI controller with the lowest address connected to the Primary Micro Channel • t0: lowest numbered device slot in the SCSI backplane (SCSI ID 0) • d0: device Logical Unit Number 0 • s0: partition 0 ../dsk indicates this is a block device
/dev/dsk/c11t5d0s1	Disk connected to: • c11: top channel on the SCSI controller with the second lowest address connected to the Primary Micro Channel • t5: sixth device slot in the SCSI backplane (SCSI ID 5) • d0: device Logical Unit Number 0 • s1: partition 1 ../dsk indicates this is a block device

SCSI Tape Device Names

The following table contains examples of SCSI tape device names.

Table 2-2. SCSI Tape Device Names (examples)

TAPE NAMES	EXPLANATION OF NAME
/dev/rmt/c0t3d0s0	Tape device connected to: • c0: bottom channel on the SCSI controller with the lowest address connected to the Primary Micro Channel • t3: fourth device slot in the SCSI backplane (SCSI ID 3) • d0: device Logical Unit Number 0 • s0: rewind tape on open and close
/dev/rmt/c10t3d0s0n	Tape device connected to: • c10: bottom channel on the SCSI controller with the second lowest address connected to the Primary Micro Channel • t3: fourth device slot in the SCSI backplane (SCSI ID 3) • d0: device Logical Unit Number 0 • s0n: do not rewind tape on open and close

Flexible (Flex) Disk Device Names

The table below contains examples of flex disk device names. These devices are supported by the **fd** driver.

Table 2-3. Flex Disk Names (examples)

FLEX NAMES	EXPLANATION OF NAME
/dev/rdsk/f03ht	Flex drive 0, 3.5”, high density, double sided, total disk, use when formatting
/dev/rdsk/f13ht	Flex drive 1, 3.5”, high density, double sided, total disk, use when formatting
/dev/rdsk/f0t	Flex drive 0, 3.5”, double sided, use for reading/writing

8-Port Serial Controller Device Names

The following table contains examples of 8-port serial controller device names. These devices are supported by the **acl** and **stty** drivers.

Table 2-4. 8-Port Serial Controller Device Names (examples)

DEVICE NAMES	EXPLANATION OF NAME
/dev/term/s00 through /dev/term/s07	The eight ports on the first 8-port serial controller
/dev/term/s08 through /dev/term/s15	The eight ports on the second 8-port serial controller
/dev/term/s16 through /dev/term/s23	The eight ports on the third8-port serial controller
/dev/term/s40 through /dev/term/s47	The eight ports on the sixth 8-port serial controller

16-Port Serial Controller Device Names

The table below contains examples of 16-port serial controller device names. These devices are supported by the **acl** and **stty** drivers.

Table 2-5. 16-Port Serial Controller Device Names (examples)

DEVICE NAMES	EXPLANATION OF NAME
/dev/term/s00 through s15	The 16 ports on the first 16-port serial controller
/dev/term/s16 through s31	The 16 ports on the second 16-port serial controller
/dev/term/s32 through s47	The 16 ports on the third 16-port serial controller

Distributed Terminal Controller (DTC) Device Names

The table below contains examples of Distributed Terminal Controller (DTC) device names. These devices are supported by the **hty** and **dtc** drivers.

Table 2-6. DTC Device Names (examples)

DEVICE NAMES	EXPLANATION OF NAME
/dev/term/000 through /dev/term/xxx	Serial ports (terminals) connected to Cluster Controller Units (CCUs) on the DTC
/dev/term/lp0	Parallel printer port on 9-port CCU
/dev/dtc/0x	The Distributed Terminal Controller (DTC) board x (device node name used for diagnostics, downloading, and so on)

Console Terminal Name

The following table contains the console device name.

Table 2-7. Console Device Name

CONSOLE NAME	EXPLANATION OF NAME
/dev/console	VGA monitor
/dev/vt00	first virtual terminal
/dev/vt01	VGA monitor - second virtual terminal
/dev/vt02	VGA monitor - third virtual terminal
/dev/vt03	VGA monitor - fourth virtual terminal
/dev/vt04	VGA monitor - fifth virtual terminal
/dev/vt05	VGA monitor - sixth virtual terminal
/dev/vt06	VGA monitor - seventh virtual terminal
/dev/vt07	VGA monitor - eighth virtual terminal
/dev/vt08	VGA monitor - ninth virtual terminal
/dev/vt09	VGA monitor - tenth virtual terminal

Serial I/O Port Names

The table below contains the serial I/O port device names. These devices are supported by the **asy** driver.

Table 2-8. Serial I/O Port Device Names (examples)

PORT NAME	EXPLANATION OF NAME
/dev/tty00s, /dev/tty00, /dev/term/00, or /dev/term/00s	First serial I/O port on processor, software controlled (XON/XOFF)
/dev/tty00h or /dev/term/00h	First serial I/O port on processor, hardware controlled
/dev/tty01s, /dev/tty01, /dev/term/01, or /dev/term/01s	Second serial I/O port on processor, software controlled (XON/XOFF)
/dev/tty01h or/dev/term/01h	Second serial I/O port on processor, hardware controlled

Parallel Printer Names

The table below contains the parallel printer device names. These devices are supported by the **lp** or **SCSI** driver.

Table 2-9. Parallel Printer Device Names (examples)

PRINTER NAME	EXPLANATION OF NAME
/dev/lp or /dev/lp0	Parallel printer on System 3000 parallel connector
/dev/rdsk/**c#t#d#s#**	SCSI parallel printer

Serial Printer Names

The following table contains the serial printer device names. These devices are supported by the **acl** and **stty** drivers.

Table 2-10. Serial Printer Device Names

PRINTER NAME	EXPLANATION OF NAME
/dev/term/**s**##	Serial printer on 8-port or 16-port serial controller

Pseudo Device Names

The table below contains the **pseudo** device names.

Table 2-11. Pseudo Device Names

PSEUDO DEV NAME	EXPLANATION OF NAME
/dev/null	Programmer's bit bucket. Used to discard unwanted data.
/dev/zero	Used to zero fill
/dev/mem	Main memory
/dev/kmem	Kernel area of memory

Chapter 3

Invoke Applications

Invoke Non-Integrated Applications

Non-Integrated Applications are diagnostic and/or maintenance programs that are *not* an integral part of **osa** (OperationsAdvantage/BASE menu interface) or **sysadm** (the Operations, Administration and Maintenance menu interface), and which are not normally accessible through the FMLI (Form and Menu Language Interpreter) interface.

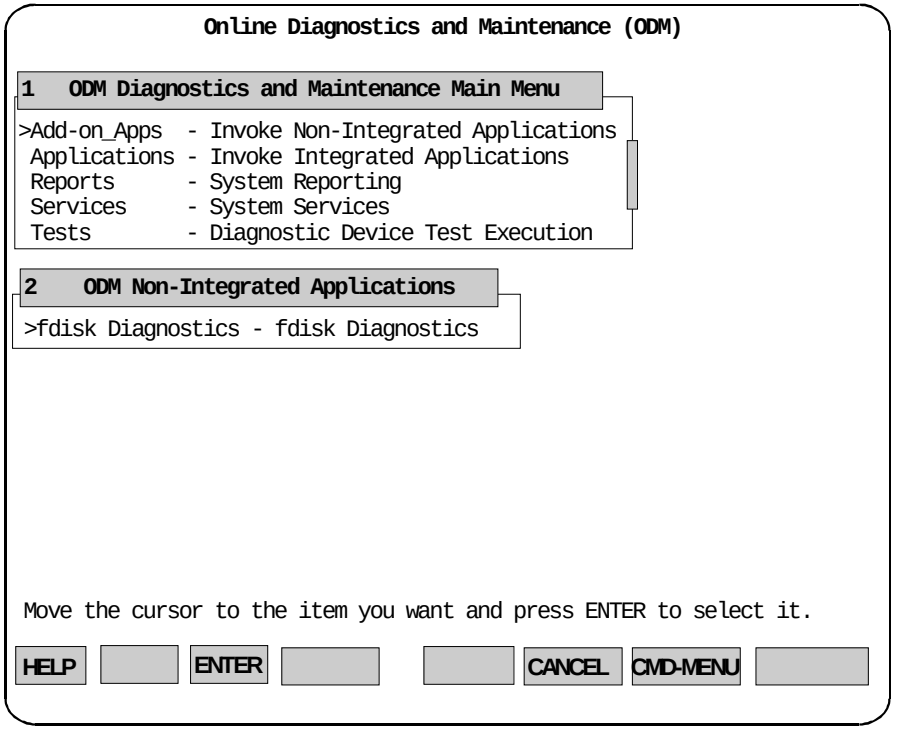
The <Add-on_Apps> menu choice of the **odm** initial menu provides an access point to maintenance and diagnostic functions installed by other packages or written by a system administrator. These functions can be any executable object (i.e., shell script, perl script, executable program) that, when selected from the ODM Non-Integrated Applications menu, will perform the selected diagnostic or maintenance task. The **odm** program invokes the executable object associated with the selected task, passing control to the executable object.

The following menu selections provide access to the diagnostic and maintenance tasks provided by non-integrated applications:

1. Select <Add-on_Apps> from the initial menu.
2. Select the diagnostic task of interest.

The **odm** menu selections required to invoke a script to display disk partitioning information (**fdisk**) are displayed in the following figure.

Figure 3-1. Invoke Non-Integrated Application



The selections in the ODM Non-Integrated Applications menu are dynamically built by the **odm** program based on information found in the directory */opt/odm/add-on*.

Note: No selections are available on this menu when the **odm** package is installed.

For further information on how to incorporate an executable object into the ODM Non-Integrated Applications menu, refer to Appendix A, “odm Interfaces”.

Invoke Integrated Applications

Integrated Applications are diagnostic and/or maintenance programs that are available through **sysadm** (the Operations, Administration and Maintenance menu interface), or which are accessible through the FMLI interface.

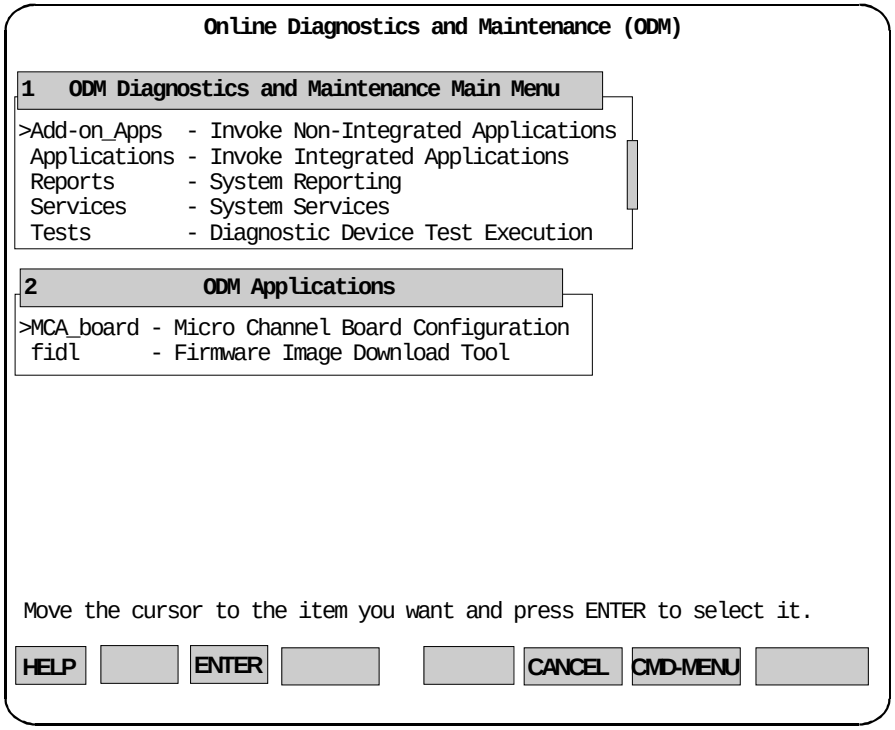
The Applications menu choice of the **odm** initial menu provides *a single point of access* to these integrated maintenance and diagnostic functions. When the Applications item is selected from the initial menu, the ODM Applications menu is dynamically built.

The following menu selections provide access to the diagnostic and maintenance tasks provided by integrated applications:

1. Select <Applications> from the initial menu.
2. Select the diagnostic task of interest.

If the **fidl** (Firmware Download) and **madf** (MicroChannel Configuration) packages are installed on the system, the ODM Applications menu would include a task for each package, as depicted in the following figure.

Figure 3-2. Invoke Integrated Application



Note: No selections are available on this menu when the **odm** package is installed.

For further information on FMLI, refer to the document *UNIX SVR4 Programmer's Guide: Character User Interface (FMLI and ETI)*.

Chapter 4

Error Log Services

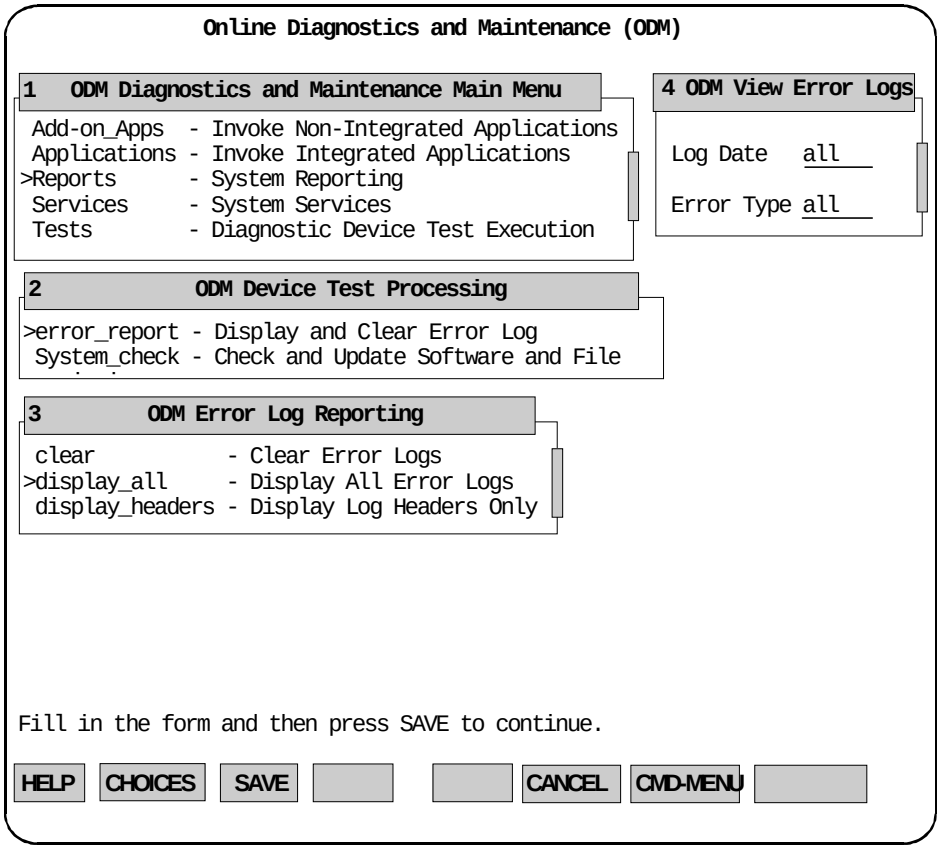
Accessing the Error Log

The menu selections below provide access to the error log. The display created by these menu selections is depicted in the following figure.

1. Select <Reports> from the initial menu**.**
2. Select <error_report> from the ODM System Reporting menu.
3. Select <display_all>, <display_headers>, or <clear> from the ODM Error Log Reporting menu.

For information about the **display_all, display_headers,** or **clear** error report operations, go to the section of this chapter that has the same name as the operation you selected.

Figure 4-1. Error Report Menu Selections



Using the display_all Selection

The **display_all** operation displays all of the error records from the system error log. Each error record displayed contains the following information:

- Header title line
- Header data line
- Error message

After selecting <display_all> from the ODM Error Log Reporting menu, you must also select the Log Date(s) and Error Type(s) on the ODM View Error Logs form.

Log Date

Use the CHOICES function key to display the available Log Dates. The choices in this field may be specific dates (such as **04-25)** or **all.**

Specific dates such as **04-25** indicate that the error log contains one or more errors that occurred on April 25.

Selecting **all** displays all errors in the log, regardless of the date when they occurred. Errors are kept in the log for one week (Monday to Monday). After selecting a specific date, or **all**, use the arrow key to move the cursor to the *Error Type* field.

Error Type

Use the CHOICES function key to display the available Error Types. CHOICES are **kernel, user,** and **all.** Use the SAVE key to store both the Log Date and Error Type selections.

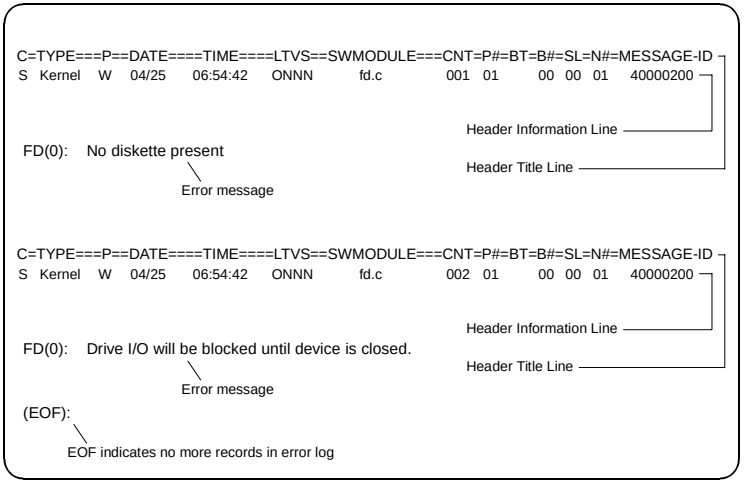
Note: If **all** is selected for both date and type, the time required to display all of the error log records could be as long as 15 minutes, possibly longer. If you select a specific date and error type, instead of taking the default of **all**, the data is displayed faster and the amount of data is more manageable. The time required to display the information is dependent on the number of log entries. The more entries in the log file, the longer it will take to display the information.

display_all Example

The following figure is an example of an error log display created by selecting the following:

- <Reports>
- <error_report>
- <display_all>
- Log Date: all
- Error Type: all

Figure 4-2. Error Log Records Display



Using the display_headers Selection

The **display_headers** operation displays only the header part of each error record in the system error log. Each error record display contains the following information:

- Header title line
- Header data line

The display does not include the ASCII message that describes the error.

After selecting <display_headers> from the ODM Error Log Reporting menu, you must select the Log Date(s) and Error Type(s) on the ODM View Error Logs form.

Log Date

Use the CHOICES function key to display the available Log Dates. The choices in this field may be specific dates (such as **04-25**) or **all.**

Specific dates such as **04-25** indicate that the error log contains one or more errors that occurred on April 25.

Selecting **all** displays all errors in the log, regardless of the date when they occurred. Errors are kept in the log for one week (Monday to Monday). After selecting a specific date, or **all**, use the arrow key to move the cursor to the *Error Type* field.

Error Type

Use the CHOICES function key to display the available Error Types. CHOICES are **kernel, user,** and **all.** Use the SAVE key to store both the Log Date and Error Type selections.

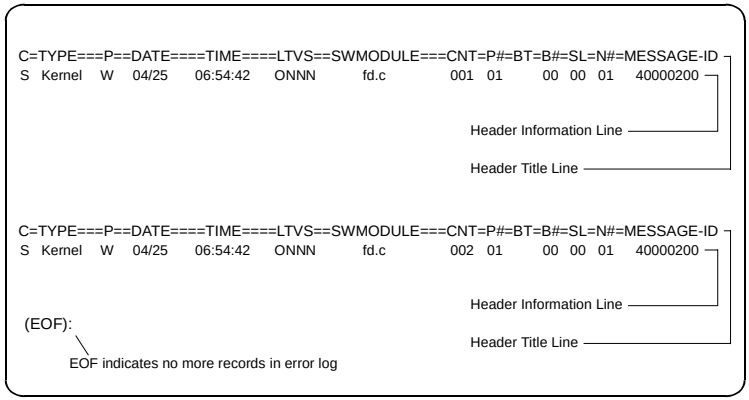
Note: If **all** is selected for both date and type, the time required to display all of the error log records could be as long as 15 minutes, possibly longer. If you select a specific date and error type, instead of taking the default of **all**, the data is displayed faster and the amount of data is more manageable. The time required to display the information is dependent on the number of log entries. The more entries in the log file, the longer it will take to display the information.

display_headers Example

The next figure is an example of an error log display created by selecting the following:

- <Reports>
- <error_report>
- <display_headers>
- Log Date: all
- Error Type: all

Figure 4-3. Error Log Headers Display



Using the clear Selection

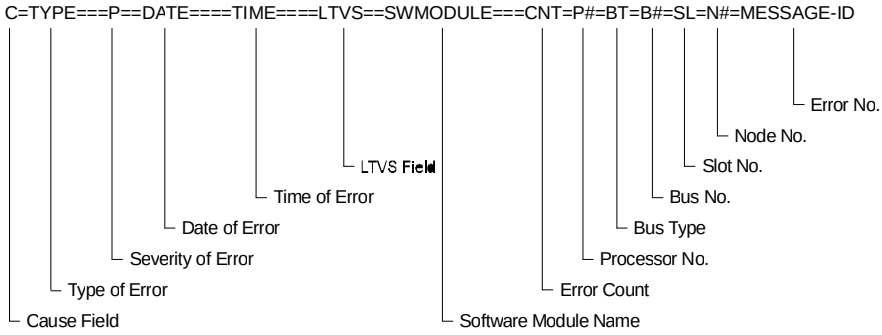
The **clear** operation clears all error records currently present in the streams error directory (*/var/adm/streams*) and/or the user error directory (*/var/adm/usererr*). The error records cleared depend on the specification of the *Error Type* field.

Error files in these directories are available for a maximum of two weeks because they are automatically cleared weekly (on Mondays). This week's (current) error files in either directory are named *error.mm-dd.* Last week's (old) error files are saved for one week in files named *oerror.mm-dd.*

Header Field Descriptions

Each error log record contains a common error header. The header is made up of two lines: The first line is the header title, and the second line is the header information. The fields in the header title line are defined in the figure below.

Figure 4-4. Error Record Header Fields



The following sections describe the type of information in each field of header information.

C Field

The C (*cause*) field contains a letter that identifies the cause of the error. The letters displayed in the *cause* field are defined in the following table.

Table 4-1. *Cause* Field in Error Record

Cause field contains . .	If the error was caused by . .
E	Environment
H	Hardware
M	Media
N	Non-error event
O	Operator
S	Software
U	Undetermined cause

Type Field

The values displayed in the *type* of error field are defined in the following table.

Table 4-2. *Type* Field in Error Record

Type field contains . .	When the type of error is . .
appl	An application error
batt	A battery error
bootup	A system start-up error
comm	A communication error
flawbk	A software flawed defective block error
kernel	A kernel error
mperip	A miscellaneous peripheral error
ovrtmp	An over-temperature error
pwrerr	A power error
scdrom	A SCSI CDROM device error

Type field contains . .	When the type of error is . . .
sdisk	A SCSI disk error
shutdn	A system shutdown error
sproc	A SCSI host processor error
sprtr	A SCSI printer error
stape	A SCSI tape error
sworm	A SCSI WORM device error
sus	A start-up subsystem (sus) error
taly	A tally threshold exception event
timech	A time change error
timing	A system timing error
tty	A serial TTY error
upderr	A software update error
upserr	An uninterrupted power source error

P Field

The P (or *severity*) field identifies the severity of the error. The table below defines the values displayed in the *severity* field.

Table 4-3. *Severity* Field in Error Record

P field contains . .	If the error record identifies . . .
C	A critical error (the system is down)
D	A degraded error (a subsystem has failed, but can still perform some functions)
F	A fatal error (a subsystem has failed and can not be used)
I	Information only

P field contains . .	If the error record identifies . . .
W	A warning message

Date Field

The *date* field identifies the date on which the error occurred.

The format of the date is MM/YY. The dates on which error log entries were created may be specified by CHOICES during the **display_all** and **display_headers** operation selections**.**

Time Field

The *time* field identifies the time of day when the error occurred.

LTVS Field

The *LTVS* field contains information about the system operation and status at the time the error occurred. **T**, **V**, and **S** information may not be available for all systems. The following table defines the values displayed in the *LTVS* field.

Table 4-4. *LTVS* Field in Error Record

The value of . .	Identifies the . .
L	Level of software that is reporting the error: 0 = Operating System 1 = Level 1 diagnostics 2 = Level 2 diagnostics 3 = Level 3 diagnostics S = Start-Up Subsystem (SUS)
T	Temperature at the time the error occurred. N identifies the Normal operating range.

The value of . .	Identifies the . .
V	Voltage at the time the error occurred. N identifies the Normal operating range.
S	System clock speed at the time the error occurred. N identifies the Normal operating range.

SWMODULE Field

The *SWMODULE* field contains the function name (in ASCII) of the software module that is the source of the error.

CNT Field

The *CNT* field identifies the number of errors associated with this error record. The value is normally one, but can be greater. For example, if the error record represents the number of *tries* that failed during a **retry** operation, then the number can be greater than one.

P# Field

The *P#* field identifies the number of the processor (CPU) that is reporting the error. This value may not help identify the faulty Field Replaceable Unit (FRU).

Note: In MP-RAS error records, three fields (BT, B#, and SL) identify the FRU that caused the error. The chart that follows the SL field description shows how the combination of these three fields identifies the FRU.

BT Field

The *BT* field identifies the type of bus that is used to access the failing hardware. The values displayed in the *BT* field are defined in the following table.

Table 4-5. *Bus Type* Field in Error Record

If BT is . . .	The Bus Type is . . .
B	Bynet bus
C	CAT bus
E	Ethernet bus
I	Devices accessed through the PSI module
J	Memory SIMM
L	LPB Devices (Monitor, keyboard, or mouse)
M	Micro Channel bus
P	Parallel (or System) bus
S or s	SCSI device slot, cable, or backplane
U	Devices built into the unit (such as batteries)
Y	YNet bus
4	Flex disk media cables or slots 1-2

B# Field

The *B#* field identifies the bus number that is used to access the failing hardware.

SL Field

The *SL* field identifies the slot number of the failing hardware module (device/controller). The following table shows the FRU as identified by the values of the bus type, bus number, and slot number fields.

Table 4-6. *SL* Field in Error Record

FRU IDENTIFICATION IN SUS ERROR RECORDS

Bus Type	Bus #	Slot #	Field Replacement Unit (FRU)
P	0	0	System Bus Backplane
P	0 or 1	1	Processor Module 1 (CPUs 0 and 4)
P	0 or 1	2	Processor Module 2 (CPUs 1 and 5)
P	0 or 1	3	Processor Module 3 (CPUs 2 and 6)
P	0 or 1	4	Processor module 4 (CPUs 3 and 7)
P	0	9	Memory Module 1
P	0	10	Memory Module 2
P	0	17	Primary Micro Channel
P	0	18	Secondary Micro Channel
J	1 or 2	XX	SIMM in JXX on Memory Module 1 or 2
C	0	0x1a	Power Supply Interface (PSI) Module
M	0	9	Local Peripheral Board (LPB) Module
M	0	1 thru 8	Primary Micro Channel Bus, Module in Slot 1 thru 8
M	1	1 thru 8	Secondary Micro Channel Bus, Module in Slot 1 thru 8
S	MMMSSSBB (8-bit representation of Bus #)	PUN	SCSI Device, where MMM=Micro Channel Bus (0 or 1), SSS=SCSI Controller's Micro Channel Slot (0-7), BB=SCSI Controller's Bus (0 or 1), and PUN=Physical Unit Number's SCSI ID
s	0	1 thru 9	Internal SCSI Bus 1 thru 9
s	1	1 thru 8	SCSI Fixed Media Backplane 1 thru 8
s	1	9	Removable Media Backplane
s	2	XY	External SCSI Cable on Primary Micro Channel Adapter where X is slot number, Y is adapter bus number
s	3	XY	External SCSI Cable on Secondary Micro Channel Adapter where X is slot number, Y is adapter bus number
L	0	0	Monitor
L	0	1	Keyboard
L	0	2	Mouse

FRU IDENTIFICATION IN SUS ERROR RECORDS			
Bus Type	Bus #	Slot #	Field Replacement Unit (FRU)
L	0	3	Flex media cable
U	0	0	Auxiliary Battery (for LPB battery-supported circuitry
U	0	1	Power Backup Subsystem (PBS) battery
4	0	1 or 2	Flex media device 1 or 2

N# Field

The *N#* field is used in multi-node (multi-processor) configurations to identify the number of the processor module reporting the error. In systems with only one processor module, the value of N# is one. See the "Table 4-7. Subsystem Numbers" table in the next section for an explanation of subsystem numbers.

MESSAGE-ID Field

The *MESSAGE-ID* field contains a unique eight or nine digit ASCII number that identifies the error. The eight or nine digit number (sometimes referred to as the *tag* number) is divided into the following groups of digits.

```
seeeeecc
   or
sseeeeecc
```

The following list describes the meaning of each digit in the tag number:

- **s** or **ss**
  This field may appear as one digit or two digits. This field identifies the subsystem associated with the error. One digit identifies subsystems **1** through **9**, and two digits identify subsystems **10** through **47**. The following table identifies the values of the subsystem (**s** or **ss**) digits from the *MESSAGE-ID* field.
- **eeeee**
  These five digits identify the error within the subsystem.

- **cc**
  These two digits provide additional information about the cause of the error within the subsystem.

Table 4-7. Subsystem Numbers

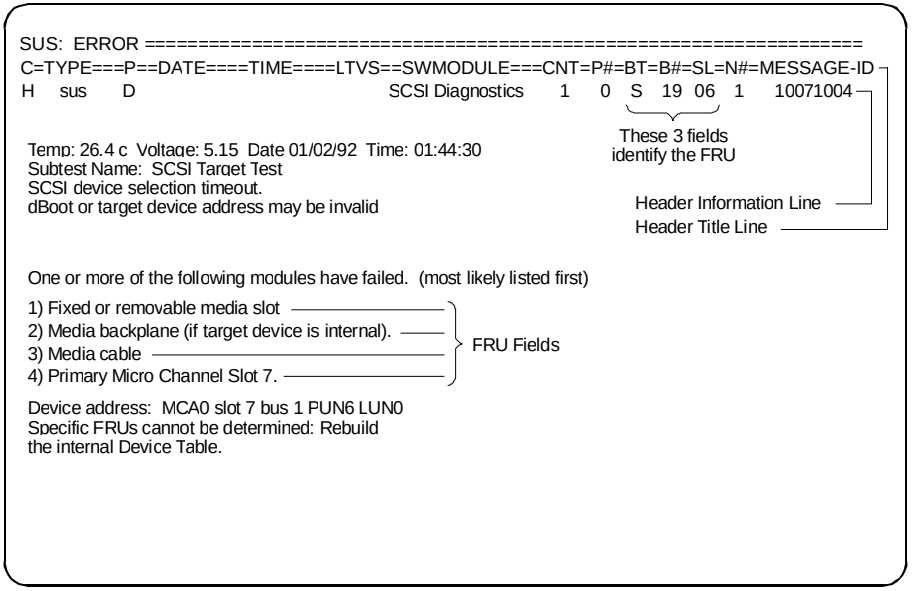
Subsystem # ..	Identifies the subsystem as . .
1	Firmware
2	The ISL
3	Reserved for future releases
4	Kernel
5	Miscellaneous peripherals
6	Ethernet LAN (ILD)
7	NetBIOS
8	WAN (HDLC, RBSL, LLD)
9	Miscellaneous applications
10	SCSI
11	TTY
12	GALAXY
13	YNet networking system
14	BYNet networking system
15	TOPEND
16	NFS
17	Stressware
18	Systems Management Applications
19	Level 2 Diagnostics
20	Not Used
21	Log File Maintenance
22	Reserved for future releases
23	POST (DOS) errors
24	Disk Array
25	AMP (Access Module Peripheral)

26	AMP ROM/Diagnostics (Level 0's)9
27	Translator Board (JBOD)
28	Environmental Board
29	APP (Application Processor)
30	Data Navigator
31-39	Reserved for future use
41	Customer Support Facility (CSF)
42	PPD Redundant Disk Array Controller
43	EaseAdvantage
44	Volume Manager (VolMgr)
45	Dump Subsystem
46	Not Used
47	Micro Channel Configuration Utilities
48	PPD RAID Manager
49	Support Sentinel
50	Tape Array Plus
51	Disk Array Plus
52	AT&T CommVault
53	AT&T NetVault
54	RBIOS POST
55	RBIOS DUMP
56	LifeKeeper
57	Deployer (aka Steamboat)
58	NM Driver (Network Manager)
59	Dual Ethernet High Performance Adapter
60	Presentation Engine Error Logging Subsystem
61	MPCM Subsystem
62	BMCA Diagnostic Software

63	Administration Workstation (AWS)
64	Server Monitor
65	Predator/S40 BIOS
66	Server Manager

The figure below is an example of a SUS error log record. Information in this error record identifies the faulty hardware component, referred to as the Field Replaceable Unit (FRU).

Figure 4-5. SUS Error Log Record



Chapter 5

System Integrity Check

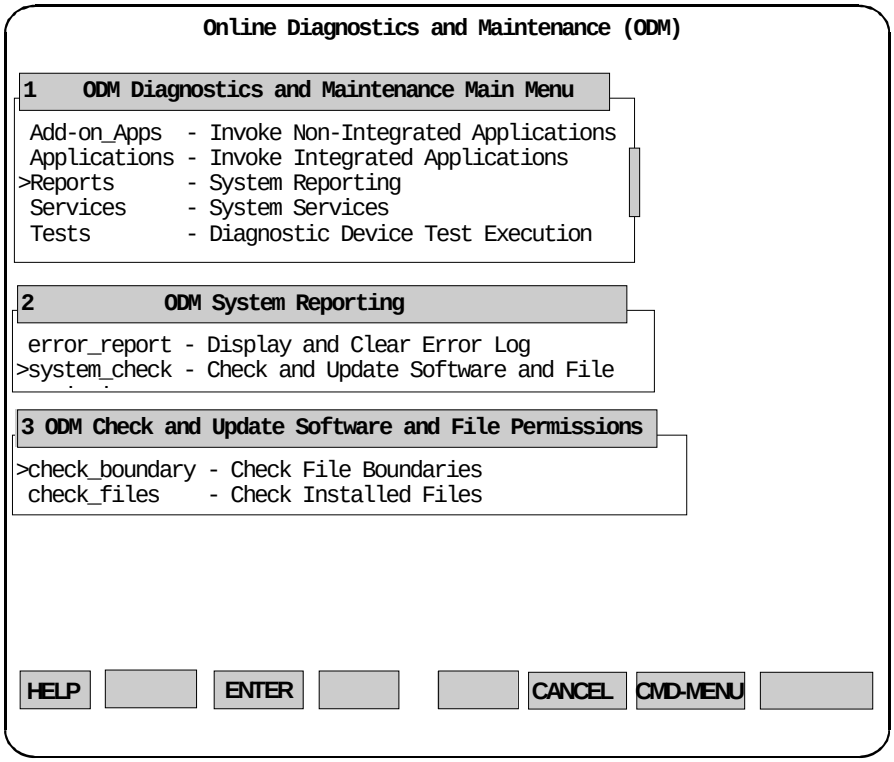
Accessing System Check Operations

The following sequence of menu selections provides access to the system check operations. The display created by these menu selections is depicted in the following figure.

1. Select <Reports> from the initial menu**.**
2. Select <system_check> from the ODM System Reporting menu.
3. Select <check_files> or <check_boundary> from the ODM Check and Update Software and File Permissions menu

For information about the **check_files** or **check_boundary** operations, go to the section of this chapter that has the same name as the operation you selected.

Figure 5-1. System Check Menu Selections



Using the check_boundary Selection

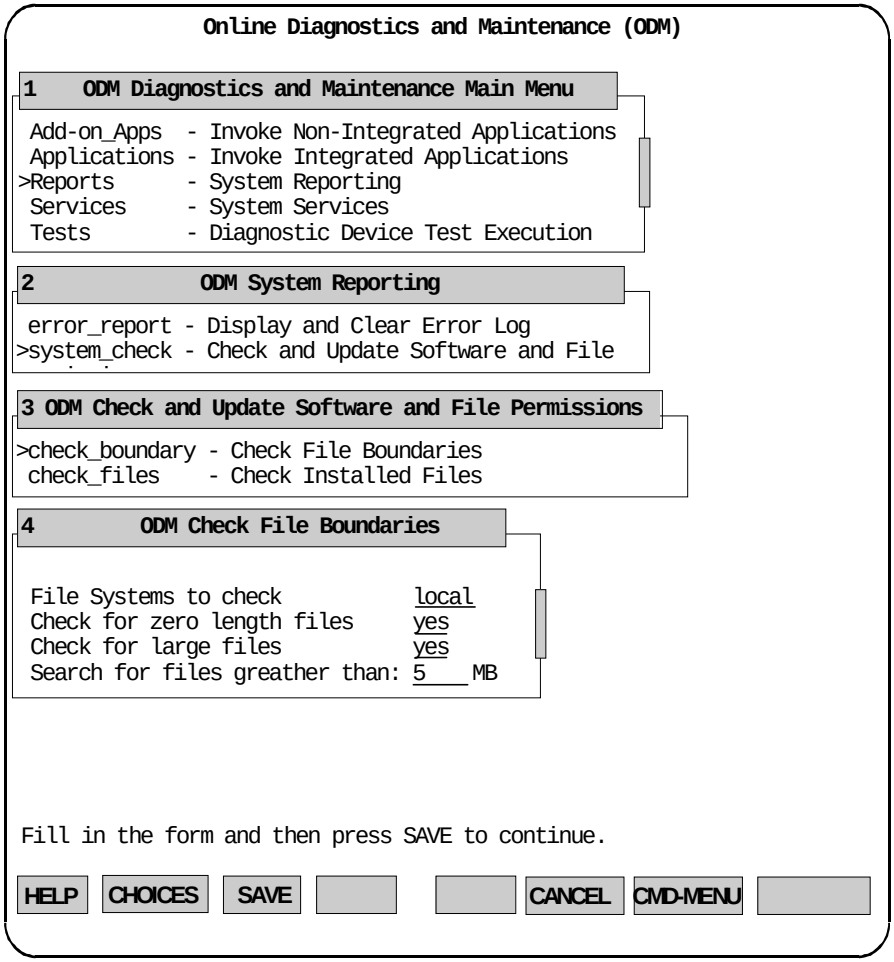
The **check_boundary** operation checks all directories for files that either have a zero length or a length greater than a user specified size.

When you select the **check_boundary** operation from the ODM Check and Update Software and File Permissions menu, the following information is required:

- File Systems to Check
- Check for Zero Length Files
- Check for Large Files
- Search For Files Greater Than XXX MB

The next figure depicts the ODM menu selections to perform the check file boundaries function, and the ODM Check File Boundaries form, which contains the options defining the file boundary check

Figure 5-2. ODM Check File Boundaries



File Systems to Check

Use the CHOICES key to display the available selections, **local** or **all**. The default value for this option is **local**.

Using the **local** option, only the local file systems will be checked for zero length or large files. Selecting CHOICES will toggle the selection to **all**, which will check all mounted file systems for zero length or large files.

Note: Selecting **all** could cause the boundary check to run for a long period of time, depending on the number of remote file systems mounted and the status of the remote host. If the remote system is not available at the time of the search, then the **check_boundary** task could appear to hang while waiting for the remote system to respond.

Check for Zero Length Files

Use the CHOICES key to display the available selections, **yes** or **no**. The default value for this option is **yes**.

Selecting **yes** will direct the search to locate and report on all zero length files. A selection of **no** will bypass this search.

Check for Large Files

Use the CHOICES key to display the available selections, **yes** or **no**. The default value for this option is **yes**.

Selecting **yes** will direct the search to locate and report on all files whose length is grater than the entered value in the Search For Files Greater Than XXX MB option. A selection of **no** will bypass this search.

Search For Files Greater Than XXX MB

Search for files greater than or equal to the value entered. The entered value is in megabytes. No choices are available.

The default for the search is to find files greater than or equal to 5 megabytes.

check_boundary Example

When you select **check_boundary** and press SAVE, the search begins. Assuming that 5 MB was the large file size selected , the following message is displayed when the search completes:

```
The following list contains files that are of zero length or
over 5MB.

Press NEWLINE to continue.
```

When you press RETURN, the utility displays a list of all files that are of zero length (if selected). If you continue, the utility displays a list of those files whose size is equal to or greater than 5MB (the selected large file size value). After displaying the list, the utility informs you that the file list has been saved as the file */tmp/listPID*, where *PID* is the process ID of the search just performed. The ODM Check and Update Software and File Permissions menu is then displayed.

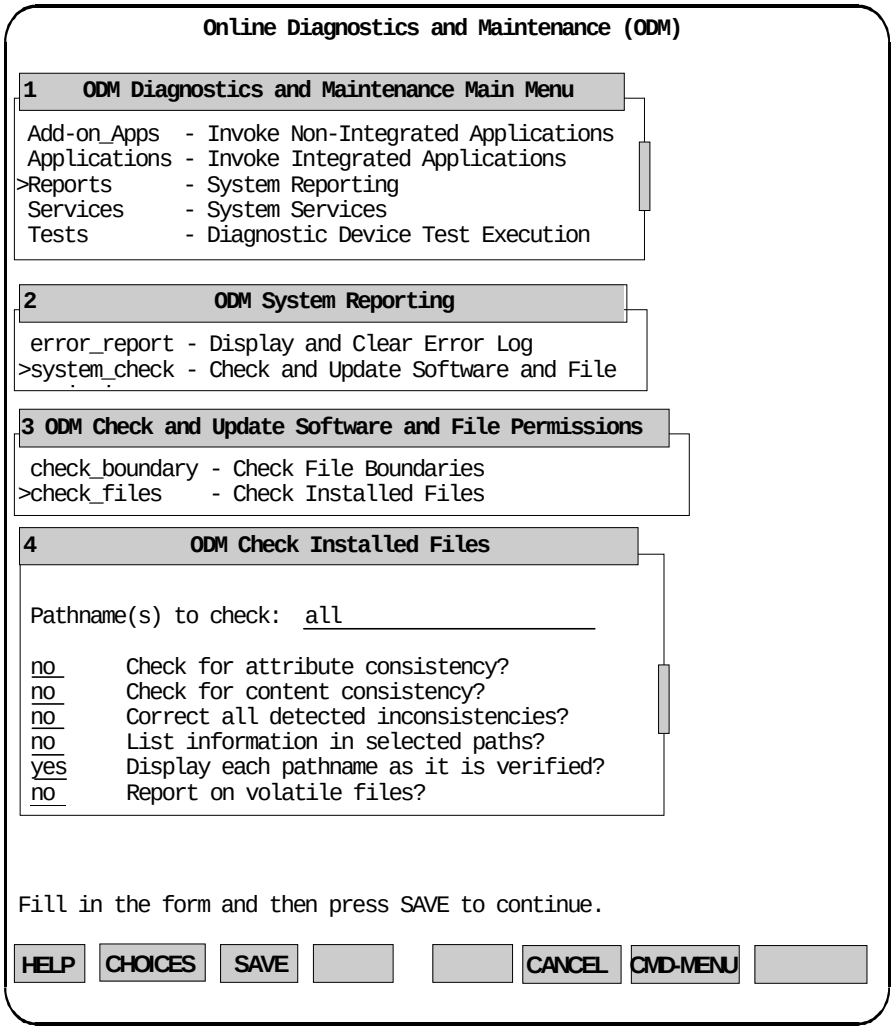
Using the check_files Selection

When you select the **check_files** operation from the ODM Check and Update Software and File Permissions menu, the following information is required:

- Pathname(s) to Check
- Check for Content Consistency
- Check for Attribute Consistency
- Correct all Detected Inconsistencies
- List Information in Selected Paths
- Display Each Pathname as it is Verified
- Report on Volatile Files

The following figure displays the ODM menu selections to perform the check installed files function, and the ODM Check Installed Files form, containing the options defining the file check.

Figure 5-3. ODM Check Installed Files



Pathname(s) to Check

The *pathname*(s) field permits you to check directories and/or files. You may enter a single file name, a directory, the pathname of a directory and its contents, or “all”.

You may enter more than one pathname. Separate each pathname with a comma.

The option settings determine how errors (inconsistencies) are displayed and corrected. If you accept the default values shown in all seven option fields, the complete list of files is displayed, and errors are shown as they are detected.

The following sections describe the options for the **check_files** files operation**.**

Check for Content Consistency

The default value for this option is **no**, which indicates that checking the sums or sizes of files is **not** performed. It does not check permissions and can not correct any errors.

Use the CHOICES key if you want to change this option setting.

Set this option to **yes** if you want to check for content consistency. Errors will be reported, but will not be corrected. The Check for Content Consistency option and/or the Check for Attribute Consistency option must be *toggled* to **yes** if the List Information in Selected Paths option is set to **no**.

If the List Information in Select Paths option is set to **yes**, the Check for Content Consistency option must be set to **no**.

Check for Attribute Consistency

The default value for this option is **no**, which indicates that checking all permissions, ownership, links, instances, and group ownership of files and/or directories is **not** performed. It does not check sums.

Use the CHOICES key if you want to change this option setting.

Set this option to **yes** if you want to check for attribute consistency. The Check for Attribute Consistency option and/or the Check for Content Consistency option must be *toggled* to **yes** if the List Information in Selected Paths option is set to **no**.

If the List Information in Select Paths option is set to **yes**, the Check for Attribute Consistency option must be set to **no**.

Correct all Detected Inconsistencies

The default value for this option is **no** because this operation may change files on the system and requires the user to possess some knowledge of the **vi** ASCII text editor. The Correct all Detected Inconsistencies option must be set to **no** if the List Information in Selected Paths option is set to **yes**.

If you move the cursor to this option, a warning message is displayed, indicating that you may change files on the system. If you use the CHOICES key to set the option to **yes**, you may press CANCEL to return to the previous menu, or you may then press SAVE to activate the change.

If you SAVE with the **yes** option setting, a list of files with errors is displayed, followed by the warning:

```
The previous screen included files or directories that were
missing or had bad sums.  They should be restored from a
backup or install tape.

WARNING:  Unless you are an experienced vi user, please do
not continue.

Would you like to continue with repairs and enter “vi”?
[y/n]
```

If you continue, the following message is displayed:

```
Pathnames in next screen are the files or directories with
errors.  Delete any pathnames which you do not want to
correct by typing ”dd” on that line. Type in ”:wq” NEWLINE
when finished.

Press NEWLINE to continue.
```

When you press NEWLINE (or ENTER), each file that you did not delete is repaired. The Check Installed Files form is displayed when the repairs are completed.

List Information in Selected Paths

The default value for this option is **no**. If you set this option to **yes**, file information is displayed, such as permissions, checksums, ownership, modifications and package dependencies as they should exist for selected files or directories.

When this option is set to **yes**, all other options must be set to **no.** If this option is set to **no**, the Check for Content Consistency option and/or the Check for Attribute Consistency option must be *toggled* to **yes**.

Display Each Pathname as it is Verified

The default value for this option is **yes**, which indicates that each file name and any error associated with the file is displayed as the check is performed.

Use the CHOICES key if you want to change this option.

Set this option to **no** if you do not want to display the file names and errors as they are being checked.

If the List Information in Selected Paths option is set to **yes**, the Display Each Pathname as it is Verified option must be set to **no**.

Report on Volatile

The default value for this option is **no**, which indicates that checking should be done on *all* files, not just volatile files (files whose contents are expected to change after being installed) or editable files (files that are edited upon installation or removal).

This option defaults to **no** if the List Information in Selected Paths option is set to **yes**.

Set this option to **yes** if you want to check volatile and editable files in addition to all other files. This is the recommended setting.

Chapter 6

Data Capture Operations

Understanding Data Capture Operations

Data capture operations (or processes) permit you to capture data being transferred to and/or from a TTY, line printer (lp), or other device, and then to process that data into an easy-to-read formatted file. Data capture files provide information about activity and performance of the selected device(s).

The following list is a typical sequence of data capture operations that you might perform to help isolate a problem that is occurring with a TTY or lp device:

- Start a data capture process for this device
- Check active data capture processes to show a list of data capture files
- Stop the data capture process
- Process the data capture file(s) to put the data in an easy-to-read format
- Remove the data capture file after examining the data

When you select <data_capture> from the ODM System Services menu, the ODM System Data Capture menu is dynamically built, based on installed configuration files (*/opt/odm/usr/intf_install/services/datacapture*) and information found in the directory */ncrm/datacapture/devices*. The items in the data capture menu are device types which coincide with particular device drivers. For example, the <asy_devices> selection coincides with devices supported by the **asy** driver.

After selecting a device type, a “Device” Data Capture Operations form is displayed, where “Device” is the name of the device selected in the ODM System Data Capture menu. For example, if you select <asy_devices> from the ODM System Data Capture menu, the name of the form displayed is “asy Data Capture Operations”. It is in this form that you identify the target device and data capture operation to be performed on the target device.

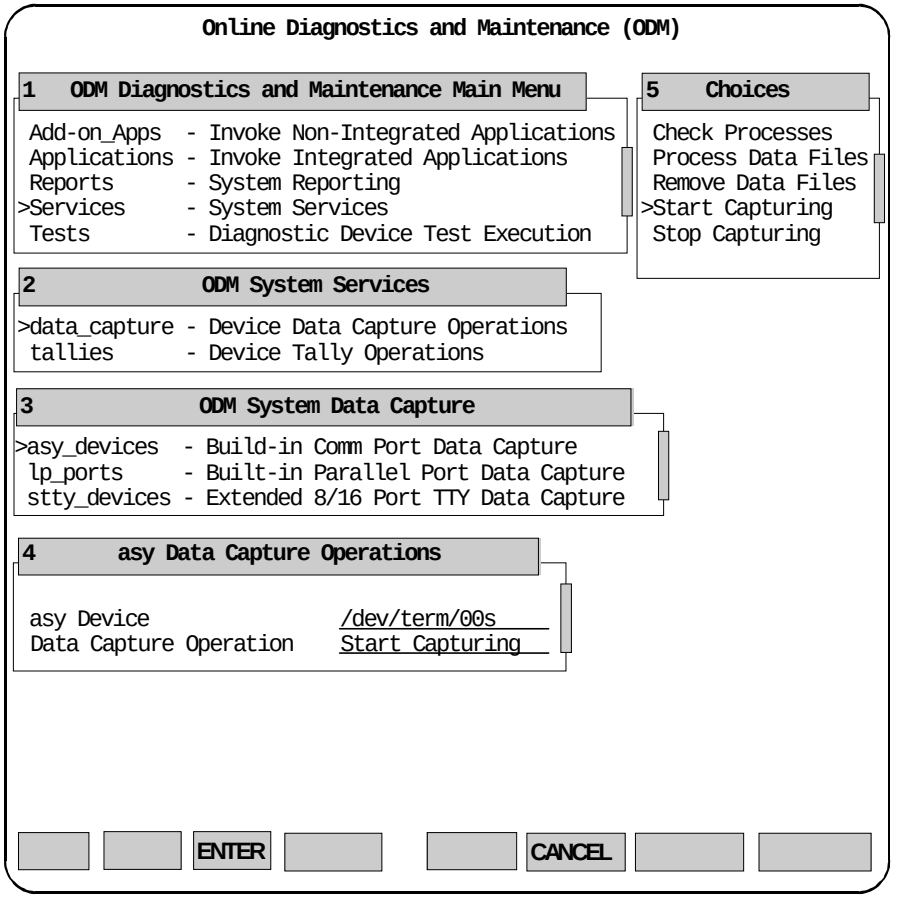
Selecting Data Capture Operations

The following sequence of menu selections provides access to the data capture operations:

1. Select <Services> from the initial menu**.**
2. Select <data_capture> from the ODM Systems Services selection menu.
3. Select the data capture device type from the ODM System Data Capture menu. The list of device types includes, but is not limited to, the following:
   a. **asy_devices**: communication port terminal devices controlled by the **asy** driver (i.e., /dev/term/00h, /dev/term/00s)
   b. **lp_ports**: parallel port devices controlled by the **lp** driver
   c. **stty_devices**: extended 8/16 port TTY devices controlled by the **acl** and **stty** drivers.
4. The "Device" Data Capture Operations form appears, where "Device" is the name of the device type selected in the ODM System Data Capture menu. A device name and data capture operation must be entered. If you know the device name, enter the desired device name and go to step 6. If you want to see the available device names, press CHOICES.
5. If you pressed CHOICES in step 4, move the cursor to the desired device name, press MARK and then press ENTER.
6. After you enter the device name, move the cursor to the Data Capture Operation field to enter the desired data capture task. If you know the operation name, enter the task name and press SAVE to perform that task. If you want to see the available data capture operations, press CHOICES.

7. If you pressed CHOICES in step 6, move the cursor to the desired operation and press ENTER to select the data capture operation. Press SAVE when your selections are complete, and the task will be performed. The following figure depicts the display created by the menu selections described in steps 1 through 7 above.

Figure 6-1. Data Capture Menu Selections



The next five sections of this chapter describe the available data capture operations, which are:

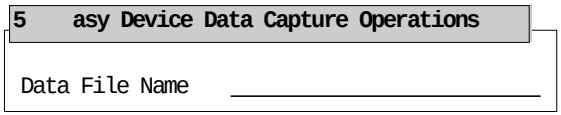
- Start Capturing
- Stop Capturing
- Check Processes
- Process Data Files
- Remove Data Files

Start Capturing

The **Start Capturing** operation captures and stores data that is transmitted and/or received on the specified device line. This operation stores the raw captured data in a disk file that can be used later to help determine the accuracy level of the link (the communication path to/from the device).

Using the input sequence noted in the section "Selecting Data Capture Operations" above, select Start Capturing for the operation and press SAVE. This results in the display of another form, depicted in the next figure, in which you specify the name of the file where the raw data will be stored.

Figure 6-2. Data Capture File Name Form



The file can be located in any directory, but the following directory is recommended:

/var/odm/datacapture/filename

The data capture utility rejects *filename* if it already exists.

After starting data capture, you may use the following data capture operations to affect the data captured in this file.

- Check Processes. Check the status of data capture processes to see which files and devices are associated with data capture and to see if they are currently capturing data.
- Stop Capturing. Stop currently active data capture processes.
- Process Data Files. Access the data captured in the data capture files.

Note: If the data capture process is performed on a high-speed line or on a high-volume line, the process creates a very large data capture file in a very short time. To prevent the data capture files from using excessive operating system file space, use the stop data capture selection and remove the data capture files as soon as possible.

Stop Capturing

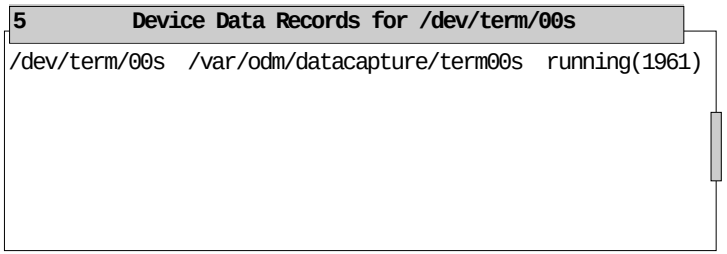
Using the input sequence noted in the section “Selecting Data Capture Operations” above, select Stop Capturing for the operation and press SAVE. This operation terminates any active data capture processes for the selected device.

Check Processes

The **Check Processes** operation displays a list of data capture files associated with data capture processes for the selected device.

Using the input sequence noted in the section “Selecting Data Capture Operations” above, select Check Processes for the operation and press SAVE. This action opens a window, similar to that depicted in the following figure, which displays a list of data capture files associated with the selected device.

Figure 6-3. Data Capture Files



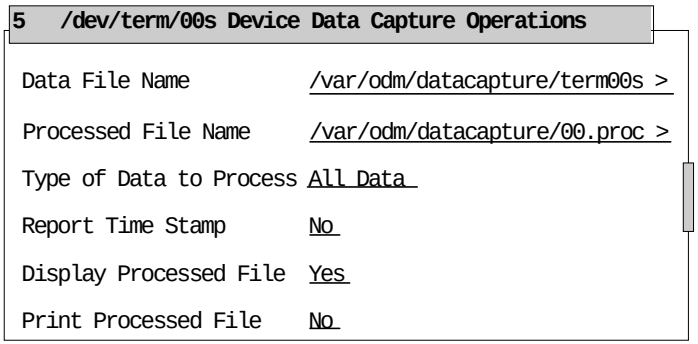
If a data capture file shown in this list is part of a currently active data capture process, the display identifies the process number in parentheses.

Process Data File

The **Process Data File** operation permits you to format an active or inactive raw data file that has been captured for the selected device and permits you to display and/or print the resulting easy-to-read formatted file.

Using the input sequence noted in the section “Selecting Data Capture Operations” above, select Process Data Files for the operation and press SAVE. A “/dev Device Data Capture Operations” form is displayed, as depicted in the figure below.

Figure 6-4. Process Data Capture File Form



Use the procedure below to fill out the form to process the data capture file.

1. Enter the Data File Name. If you do not know the complete file name, press CHOICES to display the existing data capture file names for the selected device. Select a name from the displayed list.
2. Use the arrow keys to move the cursor to the *Processed* File Name field. Enter a name for a file you want to create (not an existing file).
3. Notice that the next four fields already contain default values. If you want to accept the default values for all four fields, press SAVE to complete the selections and activate the process. If you want to change any of the last four fields, continue with the following step.

4. Use the arrow keys to move the cursor to the *Type of Data to Process* field. Select the Type of Data to Process. The default value is All Data. Additional CHOICES are Input Data or Output Data.
5. Use the arrow keys to move the cursor to the Display Processed File field. Select Yes or No in the Display Processed File field.
6. Use the arrow keys to move the cursor to the Print Processed File field. Select Yes or No in the Print Processed File field. If Yes is selected, the processed file is printed to the default printer setup for the lp subsystem.
7. Press SAVE to complete the selection and activate the file processing. The Process Data Files operation creates the formatted output file and provides you with the option to display and/or print the formatted data for the selected device.

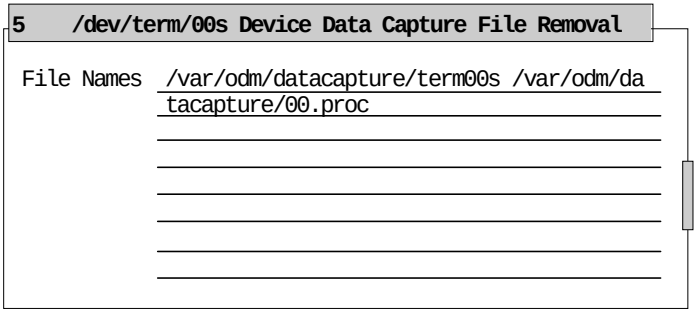
Remove Data Files

The **Remove Data Files** operation provides you with the capability to remove inactive or processed data files associated with the selected device. Active files can not be removed because a data capture operation is in process for those files. You may MARK up to 10 file names for removal during this selection.

You must specify the names of files to be removed, or use the CHOICES key, and MARK the file names.

Using the input sequence noted in the section “Selecting Data Capture Operations” above, select Remove Data Files for the data capture operation and press SAVE. A “/dev Device Data Capture File Removal” form is displayed, as depicted in the following figure.

Figure 6-5. Remove Data Capture File Form



Use the procedure below to remove previously processed data capture file(s) associated with a specific TTY device.

1. Specify the name(s) of the files to be removed. Use the CHOICES key if you need to see a list of processed files.
2. MARK the name(s) of the file(s) to be removed.
3. Press ENTER to remove the selected files.

Chapter 7

Tallies

Understanding Tallies

Tallies count specific system operations and events for specified types of devices. Peripheral device drivers maintain tally structures and have memory areas dedicated to tally activity.

Each time a tallied event occurs, the tally associated with that event is incremented. At selected time intervals, a tally daemon utility determines if any tallies have exceeded their specified error threshold levels and updates system files that contain tally information.

odm menus permit you to monitor the tallies associated with the following types of devices:

- TTY devices
- Flexible (flex) disk
- Parallel printers
- SCSI devices

These device tallies are provided by default on the MP-RAS 3.02.00 system. Other packages that interface to **odm** may provide tally monitoring for their devices.

Selecting Tally Operations

The following sequence of menu selections provides access to the tally operations for devices:

1. Select <Services> from the initial ODM menu.
2. Select <tallies> from the ODM System Services menu.
3. The ODM System Tally Processing menu appears. Select the class of devices on which to perform tally operations and press ENTER. The list of device classes includes, but is not limited to, the following:

   a. **asy_devices**: communication port terminal devices controlled by the **asy** driver (for example, /dev/term/00h, /dev/term/00s)

   b. **flex devices**: flex drive devices controlled by the **fd** driver

   c. **lp_ports**: parallel port devices controlled by the **lp** driver

   d. **SCSI_devices**: SCSI devices controlled by the SCSI driver (for example, /dev/rdsk/c*, /dev/rmt/c*)

   The “Device” Tally Operations form is displayed.

4. If you know the device name, enter the desired name and press ENTER. If you want to see the available device names, press CHOICES, select the desired device, and press ENTER to proceed to the tally operation selection.
5. If you know the tally operation, enter the desired operation and press SAVE. If you want to see the available tally operations, press CHOICES, select the operation, and press SAVE to begin the tally operation for the selected device. Available tally operations are:

   a. Clear tallies

   b. Display tallies

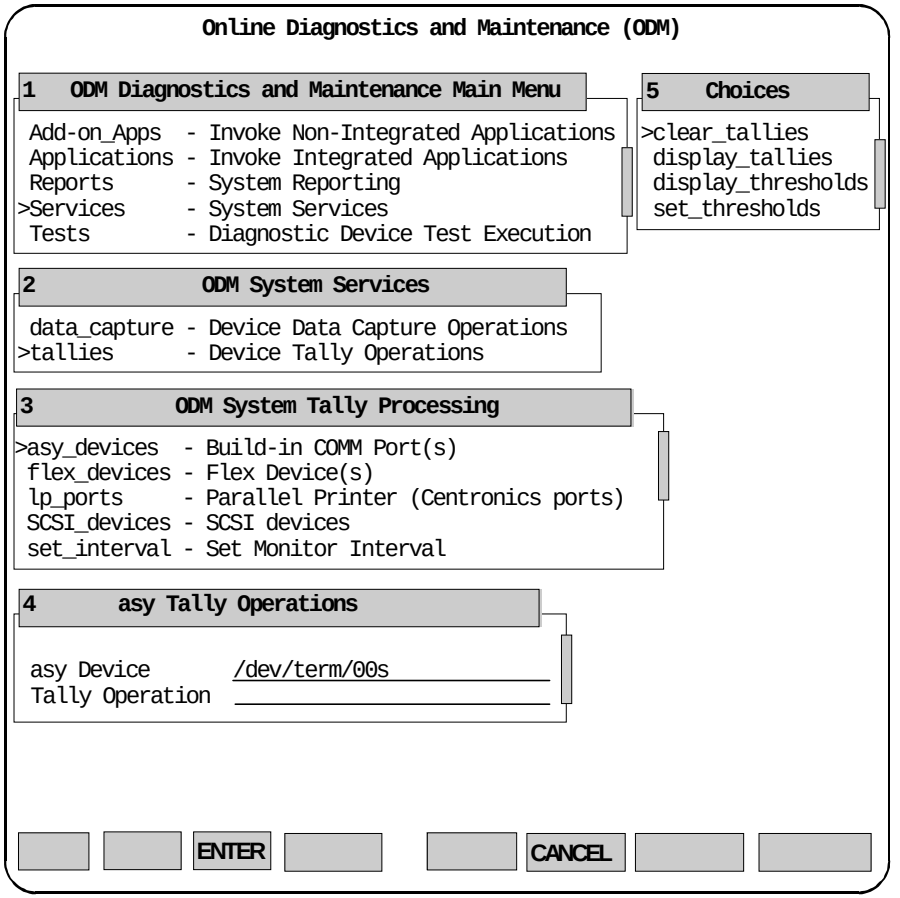
   c. Display thresholds

   d. Set thresholds

For information about the selected tally operation, go to the section of this chapter that has the same name as the operation you selected.

The following figure depicts the display created by the menu selections described in steps 1 through 5 above.

Figure 7-1. Tally Operations Menu Selections



The next five sections of this chapter describe the available tally operations, which are:

- Set Interval
- Clear Tallies
- Display Tallies
- Display Threshold
- Set Threshold

Set Interval

The **set_interval** operation displays the current number of minutes between times when the tally daemon checks to see if any tally thresholds have been exceeded. If a tally threshold has been exceeded, the daemon generates a threshold exception error record for that tally.

Note: The default interval value is 60, which causes the daemon to check for tally threshold exceptions every 60 minutes. If threshold exception errors are occurring on your system, it may be helpful to set the interval value to a smaller number so that the daemon checks tally thresholds more frequently while you are trying to find the cause of the problem.

Use the following procedure to set the tally daemon monitor time interval.

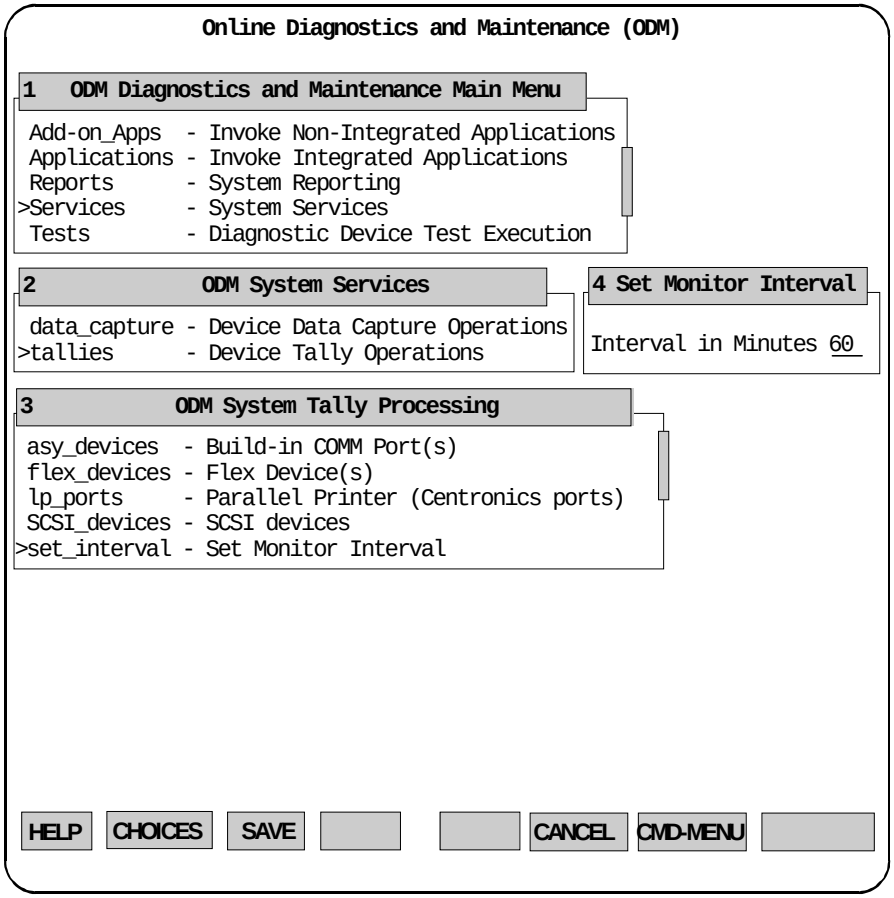
1. Select <Services> from the initial ODM menu**.**
2. Select <tallies> from the ODM System Services menu.
3. Select <set_interval> from the ODM System Tally Processing menu.
4. The **set_interval** operation displays the Set Monitor Interval form, which shows the current interval (in minutes).
5. To set the interval to a different value, enter one of the valid values shown in the next table, and press SAVE. If you want to keep the existing value, press SAVE without entering a value.
6. When you press SAVE, the system accepts the interval value, closes the Set Monitor Interval menu and returns to the “device” Tally Operations form.

Table 7-1. Tally Threshold Interval Values

INTERVAL VALUE	COMMENTS ABOUT VALUE
0	Deactivates tally daemon (performs no threshold checks)
1	Tally daemon checks threshold exceptions every minute
2	. . . . every 2 minutes
5	. . . . every 5 minutes
10	. . . . every 10 minutes
15	. . . . every 15 minutes
20	. . . . every 20 minutes
30	. . . . every 30 minutes
60	. . . . . each hour (this is the default value)
120	. . . . . every two hours
240	. . . . . every four hours
360	. . . . . every six hours
480	. . . . every 8 hours
720	. . . . every 12 hours
1440	. . . . every 24 hours

The following figure depicts the display created by the menu selections described in steps 1 through 4 above.

Figure 7-2. Set Tally Monitor Interval



Clear Tallies

The **clear_tallies** operation displays the complete list of tallies for the selected device and clears the current values for these tallies.

Using the input sequence noted in the section “Selecting Tally Operations” above, select <clear_tallies> for the operation and press SAVE to begin the clear tallies task.

1. The **clear_tallies** operation displays a table containing the following information about the tallies associated with the selected device:
   - Name of the device associated with the tallies
   - Tally IDs (a number assigned to each tally)
   - Previous value (the number of times that this event has occurred)
   - New value (reset to zero by this clear operation)
   - Description (a name that describes the type of event being tallied)

   Note: Depending on the number of tallies maintained for the selected device, the display may contain more tallies than can be displayed on the list (page) that you currently see. Use the PREVPAGE, NEXTPAGE function keys, or the arrow keys to see all of the tallies associated with the selected device.
2. Press F6 (CANCEL function key) to close the display and return to the “Device” Tally Operations form.

Display Tallies

The **display_tallies** operation displays the complete list of tallies that have accumulated for the selected device.

Using the input sequence noted in the section “Selecting Tally Operations” above, select <display _tallies> for the operation and press SAVE to begin the display tallies task.1. The **display_tallies** operation displays a table containing the following information about the tallies associated with the selected device:

- Name of the device associated with the tallies
- Tally IDs (a number assigned to each tally)
- Current value (the number of times that this event has occurred)
- Description (a name that describes the type of event being tallied)

Note: Depending on the number of tallies maintained for the selected device, the display may contain more tallies than can be displayed on the list (page) that you currently see. Use the PREVPAGE, NEXTPAGE function keys, or the arrow keys to see all of the tallies associated with the selected device.

2. Press F8 (REFRESH function key) to update the current list of tallies displayed. This can be useful for monitoring events when device problems occur.
3. Press F6 (CANCEL function key) to close the display and return to the "driver" Tally Operations form.

Display Threshold

The **display_threshold** operation displays the complete list of tallies maintained for the selected device, and shows the existing threshold percentage that has been set for each tally. The **display_threshold** operation does not permit you to change the current threshold values.

Using the input sequence noted in the section "Selecting Tally Operations" above, select <display _threshold > for the operation and press SAVE to begin the display threshold task.

1. The **display_threshold** operation displays a table containing the following information about the tallies associated with the selected device:
   - Name of the device associated with the tallies
   - Tally IDs (a number assigned to each tally)
   - Threshold (a tally value/total operation ratio) percentage
   - Description (a name that describes the type of event being tallied)

   Note: Depending on the number of tallies maintained for the selected device, the display may contain more tallies than can be displayed on the list (page) that you currently see. Use the PREVPAGE, NEXTPAGE function keys, or the arrow keys to see all of the tallies associated with the selected device.

2. Press F6 (CANCEL function key) to close the display and return to the "device" Tally Operations form.

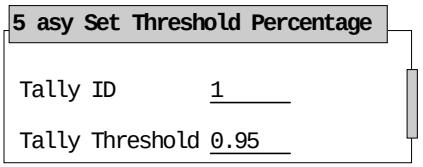
Set Threshold

The **set_threshold** operation permits you to set (or change) the threshold percentage of any tally that is maintained for the selected device.

Using the input sequence noted in the section "Selecting Tally Operations" above, select <set _threshold > for the operation and press SAVE to begin the set threshold task.

1. The **set_threshold** operation displays the Set Threshold Percentage form, depicted in the following figure. In this form, you must specify the ID of the tally whose threshold you want to change and the tally threshold that you want to establish for that tally.

Figure 7-3. Set Threshold Percentage Form



2. Enter the identification number of the tally whose threshold you wish to change. Use the CHOICES key to list the available tally IDs. You may select one ID or ALL IDs. Enter the tally ID (or ALL) and press SAVE. The cursor automatically moves to the Tally Threshold field.
3. Enter a number between 0.00 and 1.00 for the new threshold value. The value you are entering is a percent value, for example:
   - **0.01 is 1% -** Setting the threshold value(s) to 0.01 means that if the tally value for the selected tally ID reaches 1% of the total number of operations associated with this device, a *tally threshold exception* will be recorded (logged) in the error log.

- **0.50 is 50% -** Setting the threshold value(s) to 0.50 means that if the tally value for the selected tally ID reaches 50% of the total number of operations associated with this device, a *tally threshold exception* will be recorded (logged) in the error log.
- **0.00 is 0% -** Setting the threshold value to 0.00 means that NO *tally threshold exceptions* will be recorded in the error log regardless of the value of the tally ID.

4. After you enter the threshold percentage, press SAVE. The system sets the new threshold percentage and the cursor automatically returns to the "Device" Tally Operations form.

Chapter 8

Testing SCSI Devices

SCSI Device Names

SCSI device names have the format **c###t#d#s#.** The following explanations describe each part of the name:

c### is the controller number

may be one, two, or three digits. The most significant # identifies the Micro Channel (0 for Primary Micro Channel, 1 for Secondary Micro Channel, and so on). The center # identifies the controller to which the device is connected (0 for the controller with the lowest address, 1 for the controller with the next higher address. The least significant # identifies the channel (or bus) on the controller (0 for the bottom channel, 1 for the top channel).

Note: When device nodes are created by **mktable**, the leading zero(s) are dropped from the device name. For example, a device name **c000t0d0s0** is displayed as **c0t0d0s0**.

t# is the target device SCSI ID

= 0 through 6 (on each SCSI backplane in the system).

d# is the device Logical Unit Number (LUN)

is usually 0, but may be used differently in disk arrays.

s# is the UNIX partition or slice number for disk devices.

Each disk in the system may be divided into partitions for UNIX and/or other operating systems. Each partition in the UNIX operating system may have 16 parts, called slices. These slices are numbered **0** through **f** (hexadecimal numbers). Zero (**0**) refers to the complete UNIX partition, and **1** through **f** refer to individual slices (or file systems).

s# is the rewind option setting for tape devices. The following table contains examples of SCSI fixed (hard) disk names. Table 8-2 contains examples of SCSI tape device names.

Table 8-1. SCSI Fixed (Hard) Disk Device Name Example

FIXED DISK NAMES	EXPLANATION OF NAME
/dev/dsk/c0t0d0s0	First SCSI controller, SCSI device ID 0, LUN 0, partition 0 (../dsk indicates this is a block device)
/dev/dsk/c0t0d0s1	First SCSI controller, SCSI device ID 0, LUN 0, partition 1 (../dsk indicates this is a block device)
/dev/rdsk/c0t1d0s0	First SCSI controller, SCSI device ID 1, LUN 0, partition 0 (../rdsk indicates this is a character device)
/dev/rdsk/c0t5d0s1	First SCSI controller, SCSI device ID 5, LUN 0, partition 1 (../rdsk indicates this is a character device)

Table 8-2. SCSI Tape Device Name Example

TAPE NAMES	EXPLANATION OF NAME
/dev/rmt/c0t3d0s0	First SCSI controller, SCSI device ID 3, LUN 0, (s = 0 indicates rewind on open and close)
/dev/rmt/c0t3d0s0n	First SCSI controller, SCSI device ID 3, LUN 0, (s = 0n indicates do not rewind on open or close)

Selecting SCSI Diagnostics

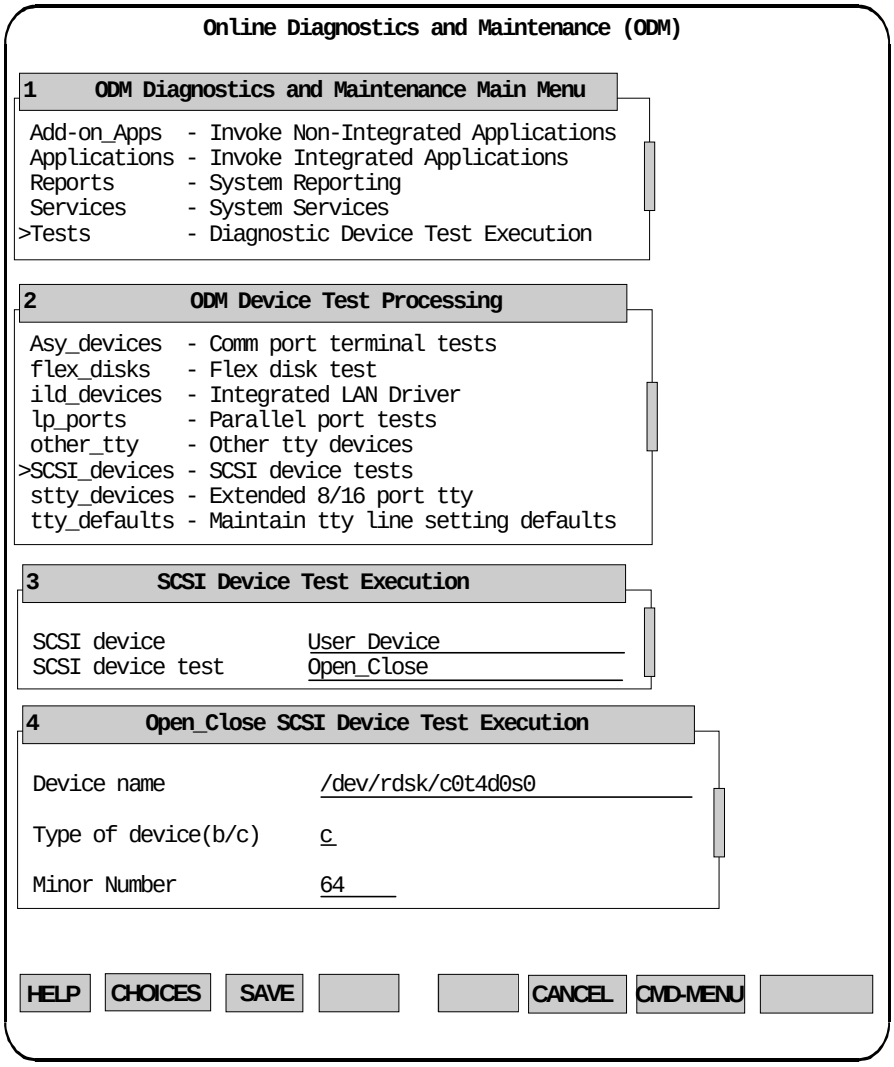
The following sections discuss the available tests for SCSI devices. Two different sets of tests are available depending on the packages installed. The **odm** package installs a basic set of SCSI tests to help determine if the device is connected correctly via an open/close test, and to identify the current firmware revision of the device to determine if the firmware needs to be updated via **fidl** (Firmware Download) utilities installed with the **fidl** package. More detailed SCSI tests are available via the **scd** (SCSI Common Diagnostics package). Both sets of tests will be discussed in this section.

Menu Selections - odm SCSI Tests

The following sequence of **odm** menu selections provides access to the SCSI device diagnostics installed by the **odm** package. If the **scd** package is installed these tests ***are not*** available.

1. Select <tests> from the initial menu.
2. Select <SCSI_devices> from the ODM Device Test Processing menu and the SCSI Device Test Execution input form is displayed.
3. If you know the name of the SCSI device, enter the SCSI device name and press ENTER to proceed to the SCSI device test input field. Otherwise press CHOICES to get a list of devices to select from, including “User Device” for newly added devices. After selecting the device, press ENTER to proceed to the SCSI device test input field.
4. If you know the name of the test to perform, enter the test name and press SAVE. Otherwise, press CHOICES to get a list of the available tests (open_close or version). Select the test and press SAVE.
5. If the device selected is an existing device, then the selected test is run. If “User Device” is selected then the SCSI Device Test Execution form is displayed as depicted in the following figure.

Figure 8-1. Open_Close SCSI Device Test Execution



6. Enter the SCSI device name following the format discussed in the section “SCSI Device Names” above and press ENTER. No choices are available.
7. Enter the Type of device(**b**/**c**) (**b** for block device, or **c** for character device) and press ENTER.
8. Enter the Minor Number for the device. This field is optional. If it is not supplied then the minor number is calculated using the device name. Press SAVE to start the test.

When the test completes, the results are displayed. If the test selected is **Open_Close**, then a status of succeeded or failed is displayed. If the test selected is Version, then the current firmware version of the device is displayed. If the test failed, possible causes for the failure are listed (things to check before attempting the test again).

Menu Selections - scd SCSI Tests

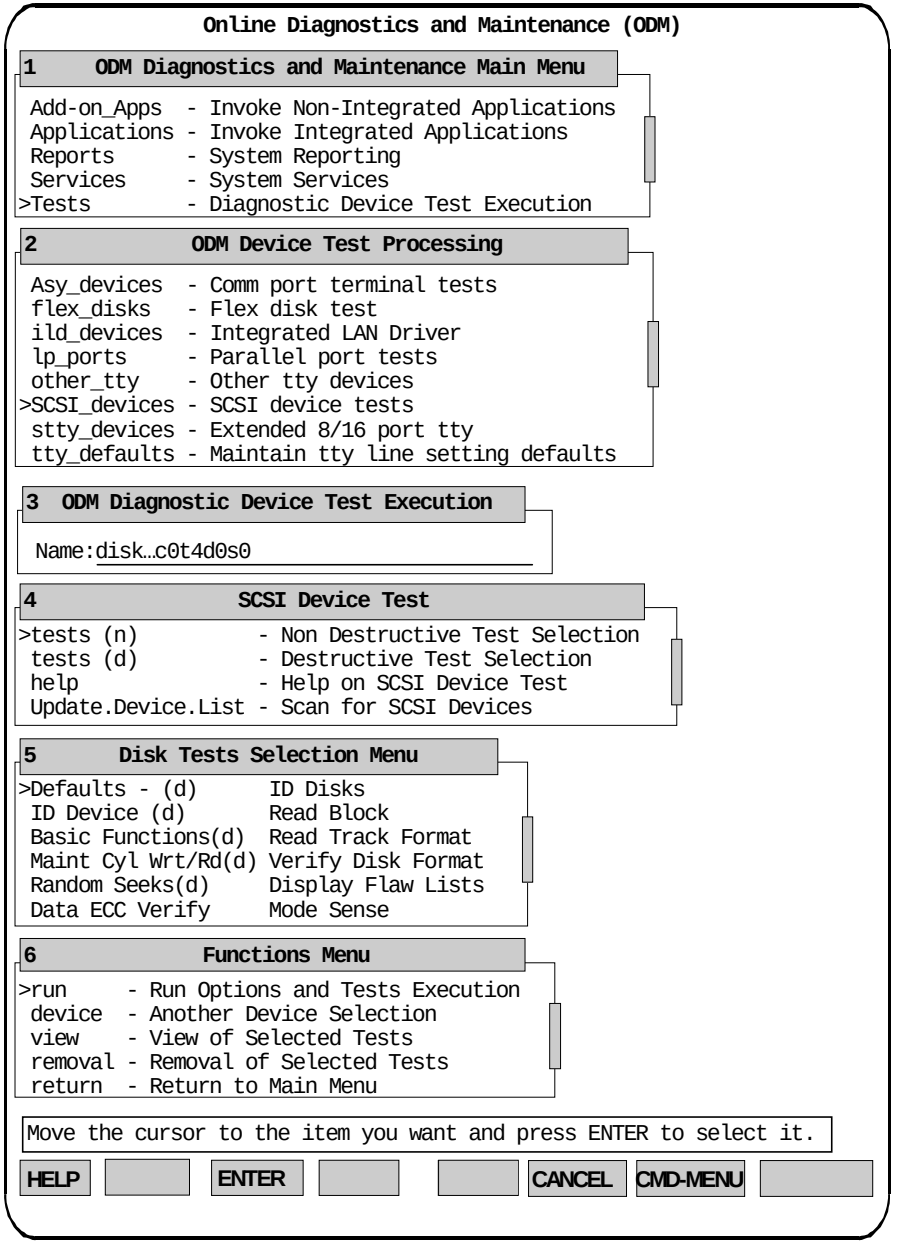
The following sequence of menu selections provides access to the SCSI device diagnostics installed by the **scd** package. An example of the display created by these menu selections is depicted in the “**scd** SCSI Device Test” figure.

1. Select <Tests> from the initial menu**.**
2. Select <SCSI_devices> from the ODM Device Test Processing menu.
3. If you know the name of the SCSI device, type that name in the ODM Diagnostic Device Test Execution form and go to step 6. If you do not know the device name, select CHOICES to display a list of available device names, and go to step 4.
4. From the CHOICES menu, use the arrow key to move the cursor to the desired SCSI device name and press MARK to select that device.
5. Press the ENTER function key (or the keyboard ENTER key) to put the selected device name into the device name field.
6. Press the SAVE function key.

7. When the program displays the SCSI Device Test menu, select the type of tests to perform (non-destructive tests or destructive tests) to display the names of available tests, or select help and press the ENTER function key.
8. From the Disk Tests Selection menu, use the MARK key to select the device test(s). When all desired tests are marked, press the ENTER key to store the list of tests and go to the Functions Menu. If the selection "Defaults=(d)" displays, marking that option automatically selects all other default selections that have a (d) beside them. Any other selections that exist without a (d) beside them must be marked to be selected.
9. In the Functions menu, select one of the following functions:
   - Run. Enter options and perform tests.
   - Device. Select an additional device.
   - View. Display a list of selected devices and tests.
   - Removal. Delete all currently selected tests.
   - Return. Close all open menus and return to the initial menu.
10. If <run> was selected in the Functions menu, a form will be displayed to enter the test options. Enter a loop number in the Total loops field to define the number of times to perform the test. Enter Y or N in the following fields:
    - Stop on error
    - Pause between tests
    - View session error log
    - Disable system error log
    - Total loops (number of times to test)

    After setting the options, press SAVE to start the testing.

Figure 8-2. scd SCSI Device Test



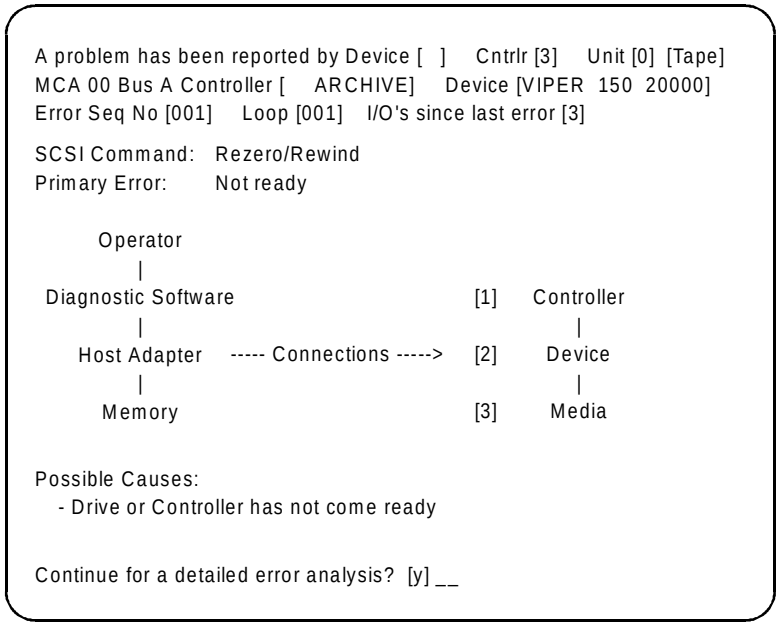
SCSI Diagnostic Error Reports

SCSI diagnostics may display multiple levels of error report screens. These screens always occur in the same sequence, and are referred to as the possible cause screen, and the detailed error analysis screen.

Possible Cause Screen

The possible cause screen is the first of two possible error report screens. The next figure depicts an example of a possible cause screen for a SCSI tape device error.

Figure 8-3. Possible Cause Screen

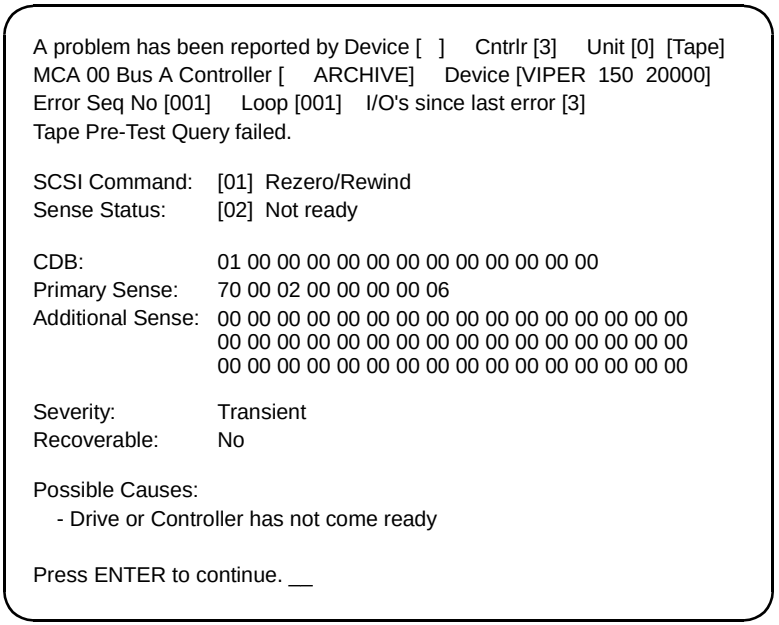


If you enter **y** to “Continue for a detailed error analysis”, the diagnostic program displays a detailed error analysis screen like the example depicted in the figure “Detailed Error Analysis Screen”.

Detailed Error Analysis Screen

In the detailed error analysis screen, the sense and status fields provide detailed information about controller and drive hardware errors. The next figure depicts a detailed error analysis screen for the same error as the possible cause screen.

Figure 8-4. Detailed Error Analysis Screen



SCSI Device Test Descriptions

This section contains information about the following:

- SCSI Disk Tests
- Disk Utilities
- CD-ROM Device Tests
- Write Once Read Many (WORM) Tests
- SCSI Tape Tests
- SCSI Printer Tests
- SCSI Flexible Disk Tests

SCSI Disk Tests

This section describes the SCSI disk tests that may be performed after you select a SCSI disk device from the SCSI menu.

ID (Identify) Device

The ID Device test reads and displays the following characteristics of the selected device:

- Disk Type
- Controller ID or vendor ID
- Hardware Revision Level
- Firmware Revision Level
- Cylinders, Heads, and Sectors (decimal)
- Number of user addressable blocks (hex and decimal)

Basic Functions

The Basic Functions test uses the diagnostic unit self test, test unit ready, and rezero functions to verify the operation of the disk controller.

Maint Cyl Wrt/Rd

The Maintenance Cylinder Write/Read test writes data to the maintenance cylinder, reads the same data, and compares the data read to the data written. If the test is selected by the defaults (d), all heads are tested. If the test is selected individually, you must identify the starting and ending heads to be tested.

Random Seeks

The Random Seeks test performs a series of 512 seek operations to random block addresses on the selected device.

Data ECC Verify

The Data ECC Verify test uses the SCSI Verify command to check the complete user addressable area of the disk. Any flaws that are detected during the test are noted and displayed at the end of the test. This list of flaws may be used as input for the **Assign Alternate Blocks** disk utility routine.

Random Data Wrt/Rd

The Random Data Write/Read test writes random data to selected areas of the disk, reads that data, and then compares the data read to the data written. Because this is a data destructive test, it may be selected only if the **-destroy** option is selected**.**

You must indicate if you want to test the entire disk. If you enter **N,** then you must specify the first and last sectors to be tested.

Pattern Data Wrt/Rd

The Pattern Data Write/Read test writes specific patterns of data to the disk, reads the data, and compares the data read to the data written. Because this is a data destructive test, it may be selected only if the **-destroy** option is selected**.**

The test repeats the process eight times using the following data patterns:

- AAAAAAAA
- 00000000

- 8406036E
- 1 01248100
- 55555555
- FFFFFFFF
- C5C5C5C5
- F0F0F0F0

You must indicate if you want to test the entire disk. If you respond **N**, you must specify the first sector and last sector to be tested.

Disk Utilities

Growth flaws can be identified by examining the system error log for recoverable and unrecoverable errors. If you select Disk Utilities, you may perform the utility functions described in this section.

Read Block

The **Read Block** utility reads one or more blocks of data from the disk. You must enter the first and last blocks to read. You may enter the block numbers by logical block address or by cylinder/head/sector numbers.

Read Track Format

The **Read Track** Format utility reads and displays selected track formats. This utility is not available on all drives.

You must select the track to be read. You may select the block numbers by logical block address, physical block address, or by cylinder and head numbers.

Assign Alternate Block

The **Assign Alternate Block** utility permits new flaws (sometimes called growth flaws) to be assigned alternate blocks (or tracks) as required by the controller. This utility destroys data in the blocks for which alternates are being assigned.

For each logical bad block address, the utility reads the data from the block, and writes the data to the alternate block. If an unrecoverable error occurs on the read, a zero-filled block is written to the assigned alternate.

Verify Disk Format

The **Verify Disk Format** utility checks the complete disk for valid data format and displays a list of assigned alternate sectors (or tracks). This function typically requires up to 20 minutes to complete.

Display Factory Flaws

The **Display Factory Flaws** utility displays a list of factory flaws written in the locked factory flaw track of the disk. These flaws are always assigned alternates during a Format Unit operation. It is not necessary to manually enter these flaws.

This utility is not permitted to access some factory flaw lists. However, this does not prevent the controller from reading the flaw list and assigning alternates to those flaws during the **Format Unit** utility.

Mode Sense

The **Mode Sense** utility works with SCSI Common Command Set (CCS) controllers. The utility permits you to examine the CCS controller parameters, but does not permit you to change them. CCS controller parameters are divided into pages.

The **Mode Sense** utility displays those parameter pages that are available from the selected controller.

Mode Select

The **Mode Select** utility activates **Mode Sense** utility, and permits you to change the parameters that the Mode Sense utility displays.

Because this utility is data destructive, it can not be selected unless the **-destroy** option is selected**.**

When you change a selected parameter, **Mode Select** copies the changed data to an output buffer and then writes it to the area of the disk where these CCS parameters are saved.

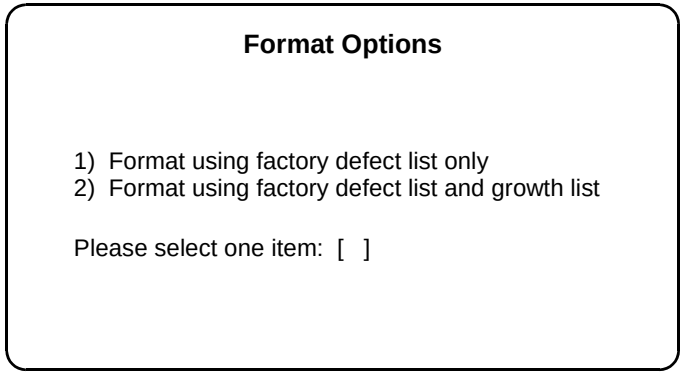
WARNING: Use this utility with caution. Know and understand CCS *controller/drive* parameters and functions before you use **Mode Select**.

Format Hard Disk

The **Format Hard Disk** utility does not perform any Operating System (OS) formatting functions such as creating boot blocks or logical file systems. This utility physically formats the disk so that the OS may use the disk to perform those functions. Because this utility is data destructive, it can not be selected unless the **-destroy** option is selected**.**

The **Format Hard Disk** utility displays the selection menu shown in the following figure, where you select the desired option for formatting the disk.

Figure 8-5. Format Hard Disk Options



Option 1

The utility formats the drive using only the primary (factory) defect list which identifies defects in the media found during manufacture. This list is stored on a protected area of the disk, and is used to assign alternate tracks/sectors whenever these defective addresses are selected by software.

Option 1 does not save any existing growth defects that have been identified. Growth defects are those that have been detected since the disk has been in use (since manufacture). You cannot enter any additional growth defects.

Option 2

The utility formats the drive using both the primary (factory) defect list and a manually entered growth defect list to reassign defective tracks/sectors. This option lets you enter additional growth defects, and saves all existing growth defects. You may also specify zero (0) additional defects.

Growth defects are entered by logical block, by cylinder/head/sector, or by cylinder/head/bytes (offset) from index.

Because NCR SCSI controllers automatically certify the media after the format utility is complete, the utility does not automatically provide additional certification for NCR controllers unless you select the **Additional Certification?** option which is displayed following the format operation.

Disk Firmware Download Utility

The Disk Firmware Download utility is intended to download microcode firmware to selected drives attached to an ADP-92, ADP-93, or Series 3 disk array, or stand-alone SCSI drives such as the 6091 disk units. The following drives are supported:

- ADP92-02:
  - SEAGATE ST1480
  - HP C2247
- ADP93-02:
  - DEC DSP3210
  - HP C2247
  - HP C3323
  - IBM DFHSS2F
  - IBM DFHSS4F
  - SEAGATE ST32550
  - SEAGATE ST15150
- ADP93-08:
  - DEC DSP3210
  - HP C2247

  - HP C3323
- Series 3:
  - IBM DFHSS2F
  - IBM DFHSS4F
  - SEAGATE ST32550
  - SEAGATE ST15150
- 6091:
  - CONNER CP30540
  - DEC DSP3105
  - DEC DSP3107
  - DEC DSP3210
  - HP C2247
  - HP C3325

You must enter the path and file name of the firmware and the verification list. Each disk drive connected to the Array Controller is sent an INQUIRY command, to verify its firmware version number to the part numbers in the verification list. If the numbers match, you may download the firmware to the Array Controller.

CD-ROM Device Tests

This section identifies the tests available to verify the operation of CD-ROM devices.

ID Device Test

The ID Device test provides information about the controller and the drive being tested. The test identifies the disk type, controller firmware and hardware revision levels, and the user addressable capacity.

Controller Self-Test Test

The Controller Self-Test test activates the controller self-test function to test the dynamic RAM controller chip.

Rezero Test

The Rezero test uses the Rezero Unit command to recalibrate the drive to the start of the disk. The drive spins up and the head moves to the start of the disk.

Random Seek Test

The Random Seek test randomly accesses 256 blocks on the disk. For each of these blocks, the controller verifies that the device accessed the correct block and read the block address field successfully.

Oscillating Seek Test

The Oscillating Seek test identifies and seeks to the middle block (m) of the media. Seeks are then sent to (m+500), (m-500), (mm+1000), (m-1000), (m+1500), and so on until the media limit is reached.

Read/Verify NCR Type Data Test

The Read/Verify NCR Type Data test requires an NCR Diagnostic CD-ROM disk to be inserted in the drive. Known data patterns on the outer, middle, and inner tracks of the media are read and verified. The data patterns are consecutive blocks of data with the first block being all 0x00, the second block being all 0x01, and so on up to the 256th block which contains all 0xff.

Read/Verify Sony Type Data Test

The Read/Verify Sony Type Data test requires an NCR Diagnostic CD-ROM disk to be inserted in the drive. Known data patterns on the outer, middle, and inner tracks of the media are read and verified. The data patterns are consecutive blocks of Sony type B data. The expected data is constructed according to the block number, then the block is read and compared (verified) to the expected data.

Write Once Read Many (WORM) Tests

This section identifies the tests available to verify the operation of WORM devices.

ID Device Test

The ID Device test uses the **Diagnostic Inquiry** and **Read Capacity** commands to identify and report the disk drive characteristics.

Basic Functions Test

The Basic Functions test uses the **Unit Self-Test**, **Test Unit Ready**, and **-Rezero** commands to verify the operation of the disk controller and disk drive.

Random Seeks Test

The Random Seeks test sends a series of 512 random seek operations to verify the seek capability of the disk drive.

Data ECC Verify Test

The Data ECC Verify test uses the SCSI verify command to verify the selected user-addressable disk surface. No data is transferred. The test displays a list of flawed blocks. You may use the list to analyze the disk and to provide input to the **Assign Alternate Block** utility.

SCSI Tape Tests

This section identifies the tests available to verify the operation of SCSI tape devices.

ID (Identify) Device

The ID Device test checks and displays the following characteristics of the selected tape device:

- Tape type

- Controller ID or vendor ID
- Hardware revision level
- Firmware revision level

Data Wrt/Rd

The Data Write/Read test writes a 16 block data pattern and two filemarks to the tape. The test then rewinds the tape, reads the tape, and compares the data read to the data written.

If the selected device is a 0.5 inch tape, the test verifies both low and high density modes.

Data Filemark Wrt/Rd

The Data Filemark Write/Read test writes two blocks of incrementing data pattern and a filemark to the tape. Then the test writes two more blocks of a different incrementing data pattern, followed by two filemarks. The test rewinds the tape, spaces to the first filemark, and then reads to verify the second data pattern.

If the selected device is a 0.5 inch tape, the test verifies both low and high density modes.

Write Files

The Write Files test writes 16 files to the tape. Each file contains 32 blocks of an incrementing data pattern and a single filemark.

After writing the last file, the test rewinds the tape, spaces one filemark, reads and compares the data pattern to the expected pattern. The test then spaces two filemarks, reads and compares the data to the next expected pattern. The test then spaces four filemarks, reads and compares the data to the next expected pattern. The test then spaces eight filemarks, reads and compares the data to the last expected pattern.

If the selected device is a 0.5 inch tape, the test verifies both low and high density modes.

Read Tape (or Data ECC Verify Test)

The Read Tape test verifies that the tape can be read reliably. The test does not depend on any specific previously written data pattern.

The test tries to read the tape, 50 blocks at a time. If the test reads a single filemark before it reads 50 blocks on any operation, it determines the number of blocks read, and displays the relative file number, the file size, and number of retries (if any) required to read the file. For files longer than 1000 blocks, the test also displays a running count of the number of blocks in the file. This number is displayed at the bottom of the screen.

Self-Test Controller

The Self-Test Controller test activates the **SCSI Self-Test Group 1** command built into the tape controller. The results are used to verify the proper operation of the controller. This Self-Test function is not available on all tape controllers.

Self-Test Unit

The Self-Test Unit test activates the **SCSI Self-Test Group 1** command built into the tape controller. The results are used to verify the proper operation of the controller and drive. This Self-Test function is not available on all tape drives.

Variable Block Data Wrt/Rd

The Variable Block Data Write/Read test writes and reads files that contain variable length records. The following typical block lengths are used:

- 0 bytes
- 1 byte
- 512 bytes
- 12288 bytes

This test is not available on 0.25 inch tape drives.

Recover Buffered Data

The Recovered Buffered Data test verifies that the selected controller and drive can perform the **SCSI Recover Buffered Data** command functions properly.

The test tries to write and then tries to recover two block and ten block data records.

Unbuffered Data Wrt/Rd

The Unbuffered Write/Read Data test verifies that the selected controller and drive perform properly in the unbuffered mode.

The test writes single filemarks and records of eight, 16, 24, and 32 blocks using five different data patterns.

If the selected device is a 0.5 inch tape, the test verifies both low and high density modes.

End-Of-Media (EOM)

The End Of Media (EOM) test writes sequentially numbered data blocks to the tape until the controller detects an EOM status. After this, the test writes a filemark, three more data blocks and two more filemarks.

The test then rewinds the tape, moves the tape to the first filemark, backs up one block, and then tries to read the last numbered block.

Finally, the test spaces the tape forward one filemark and tries to read and verify the three data blocks after the single filemark.

Known Data Write

The Known Data Write and Known Data Read/Compare tests verify compatibility between different tape drives. To do this, you should run Known Data Write on one drive and then use the same tape to run Known Data Read/Compare on the other drive.

The Known Data Write test rewinds the tape and writes 4096 blocks of incrementing data and two filemarks to the tape. The first four bytes of each 512 byte data block contain the block number. The test then rewinds the tape.

Known Data Rd/Compare

The Known Data Write and Known Data Read/Compare tests verify compatibility between different tape drives. To do this, you should run Known Data Write on one drive and then use the same tape to run Known Data Read/Compare on the other drive.

The Known Data Read/Compare test rewinds the tape, reads the tape until it reads two filemarks and then rewinds the tape. Each block read is compared to the data written in the Known Data Write test.

Erase Tape

The Erase Tape test writes eight data blocks and a double filemark. The test then rewinds and erases the tape.

The test then tries to read the tape. If the controller returns any status other than *blank tape*, the test reports an error.

SCSI Printer Tests

This section identifies the tests available to verify the operation of SCSI printers.

Identify Device

The Identify Device test uses the inquiry function to identify and display the characteristics of the selected printer.

Controller Self-Test

The Controller Self-Test selection uses the diagnostic self-test and test unit ready functions to verify the operation of the printer controller.

Write/Read/Compare Buffer

The Write/Read/Compare Buffer test selection writes data to the controller's internal buffer, reads the data, and compares the data read to the data written.

Ripple Pattern

The Ripple Pattern test selection prints a selected number of pages of all printable characters. Each line is shifted one character position from the previous line. The test prints sixty lines per page.

Complement Pattern

The Complement Pattern test selection prints a selected number of pages of the characters 0 (hex 30) and 0 (hex 4f). The test prints sixty lines per page.

Floating Bit Pattern

The Floating Bit Pattern test selection prints a selected number of pages of characters with a *1* bit floating through the four least significant bits (called a nibble) of each character. The test prints sixty lines per page.

Speed Test

The Speed test calculates the printer speed by measuring the time required to print 120 lines.

SCSI Flexible Disk Tests

This section identifies the tests available to verify the operation of SCSI flexible disk devices.

ID (Identify) Device

The Identify Device test uses the inquiry, read capacity, and diagnostic inquiry functions to identify and display the characteristics of the selected flex device.

Controller Self-Test

The Controller Self-Test selection uses the diagnostic self-test, test unit ready, and rezero functions to verify the flex disk controller.

Write/Read/Compare Buffer

The Write/Read/Compare Buffer test selection writes data to the controller's internal buffer, reads the data, and compares the data read to the data written.

Random Seeks

The Random Seeks test selection performs a series of 256 seek operations to random block addresses on the selected flex device.

Verify Data

The Verify Data test reads data from all blocks of the flex's user addressable data space and reports the block address of any error that occurs.

Write/Read Random Data

The Write/Read Random Data test writes random data patterns to selected areas of the flex, reads the same data, and compares the data read to the data written.

Write/Read Pattern Data

The Write/Read Pattern Data test writes a specified data pattern to the flex, reads the same data, and compares the data read to the data written. The test repeats this process with eight different data patterns.

The following list identifies the patterns used (in the order in which they are used:

- AAAAAAAA
- 00000000
- 8406036E
- 01248100
- 55555555
- FFFFFFFF
- C5C5C5C5
- F0F0F0F0

Format Unit

The Format Unit test physically formats the media in the selected flex disk drive. Before it starts to format the diskette, the routine asks the following question:

```
Interleave?
```

You may enter any interleave value that is valid for the specific flex disk controller being used to format the media in the flex.

Zero or one are valid values for WorldMark flexible disk controllers.

Chapter 9

Testing Flexible (Flex) Disk Devices

Selecting Flex Disk Diagnostics

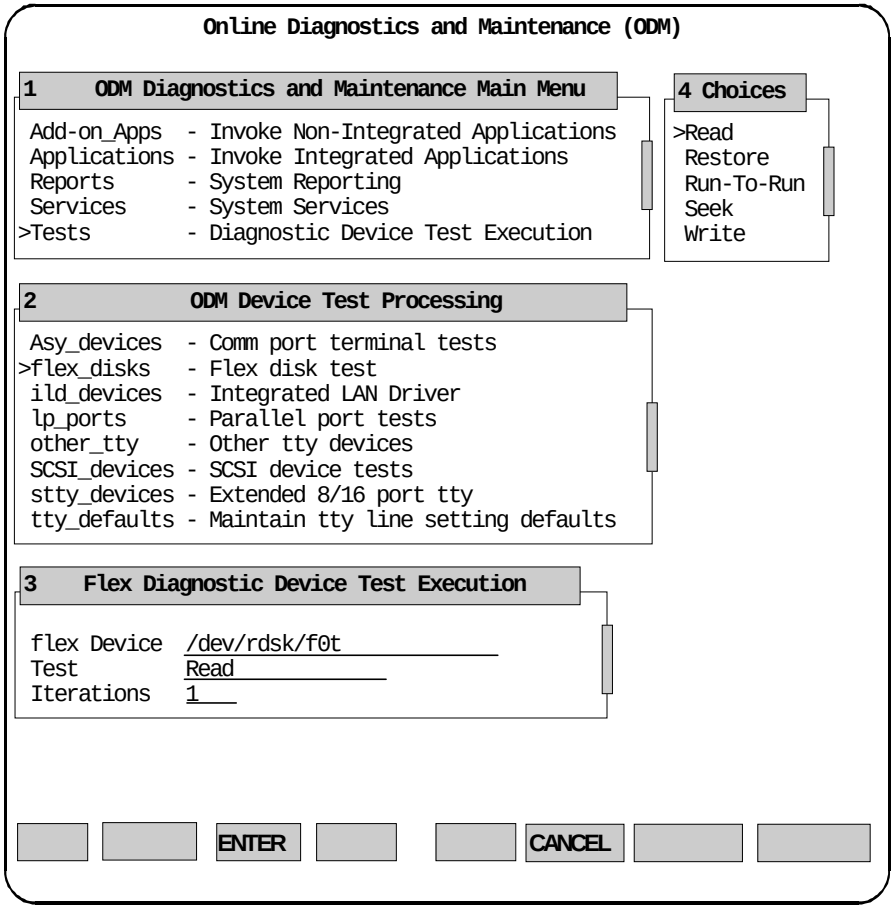
The following sequence of menu selections provides access to the flex disk diagnostics. An example of the display created by these menu selections is depicted in the next figure.

1. Select <Tests> from the initial menu.
2. Select <flex_disks> from the ODM Device Test Processing menu.
3. If you know the name of the flex disk device, type that name and press ENTER. If you do not know the device name, select CHOICES for a list of available device names, select your choice, and press ENTER to proceed to the Test identification field.
4. From the CHOICES menu, use the arrow key to move the cursor to the desired flex disk device name and press MARK to select that device.
5. If you know the test to perform, enter the desired test name and press ENTER. If you want to see the available tests for flex devices, press CHOICES, select the desired test, and press ENTER. Available flex device tests are:

   - Read
   - Restore
   - Run-to-Run
   - Seek
   - Write

   Only one test can be selected from the CHOICES menu.
6. If you want to perform the test more than one time, use the arrow keys to move the cursor to the Iterations field and enter the number of times you want to perform the test.
7. Press the SAVE function key to activate the Flex Device Test status screen.

Figure 9-1. Flex Device Test Menu Selections



When you press SAVE in the Flex Diagnostic Device Test Execution screen, the diagnostic program displays the Flex Device Test status window which is used to start the requested test. This window is used to report the status of the selected test and any errors that occur while the test is in progress. Once a test is started, it can be interrupted by pressing HALT. The “Flex Device Test” section below provides examples of expected test output.

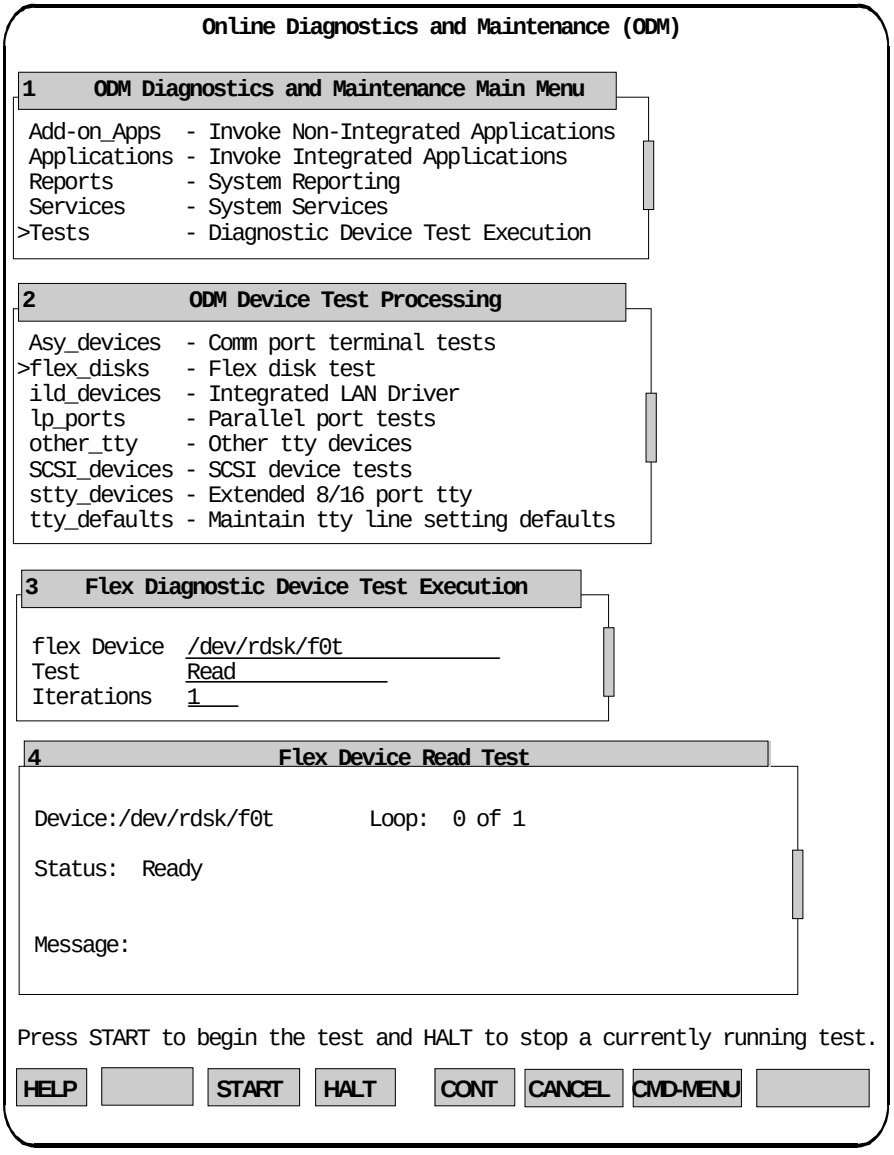
Flex Device Test

The diagnostic program uses one window to report error and status information.

The following figure contains an example of the Flex Device test window before starting the **Read** test. To control test execution, perform the following:

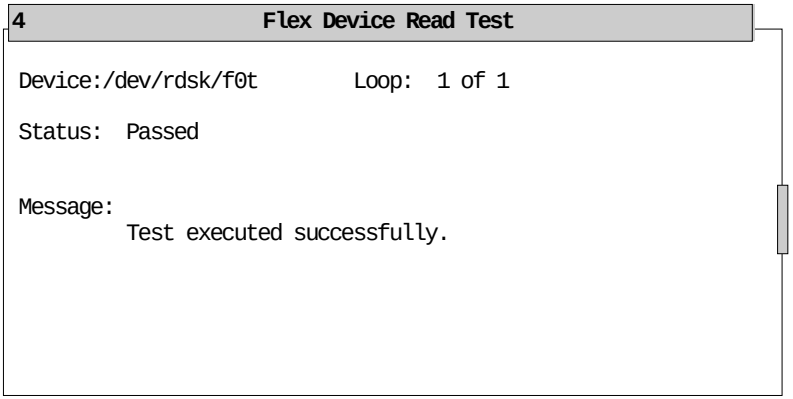
- To **begin** the test. Press START and the test will begin for the ***first*** device selected.
- To **terminate** a test in progress. Press HALT.
- To **proceed** to the next device if multiple devices were selected. The test will pause after it completes execution on the previous device. Press CONT to proceed to the next device.

Figure 9-2. Flex Device Test Status Window



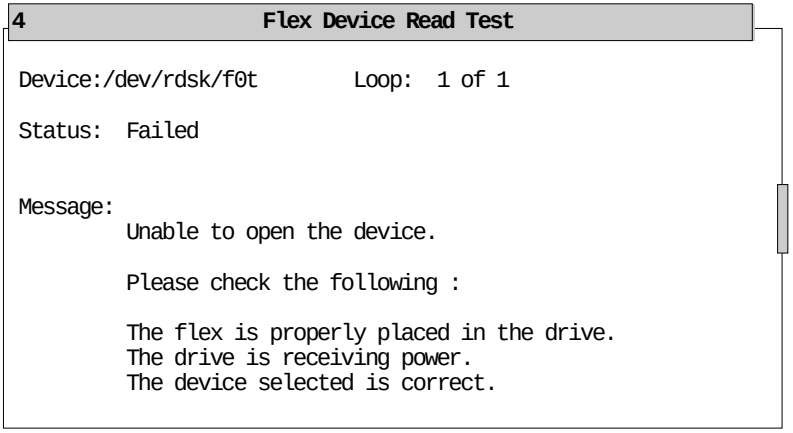
When the test completes, the status window is updated, similar to the window depicted in the next figure.

Figure 9-3. Flex Device Test: Success



If the test failed, a window similar to the window depicted in the following figure is displayed, which includes a reason for the failure.

Figure 9-4. Flex Device Test: Failure



Flex Disk Test Descriptions

After the flex disk device name is selected as shown in the figure “Flex Device Test Menu Selections”, use the CHOICES function key to display the list of available flex disk tests. The following tests may be selected from that list:

- Read
- Restore
- Seek
- Run-To-Run
- Write

Read

The Read test reads data from specific physical blocks into a buffer. The test starts at block zero and reads to the end of the disk, one block at a time. The test takes several minutes to perform.

Restore

The Restore test sends a Recalibrate function to the controller to move the drive’s read/write head assembly to cylinder 0, track 0.

The test checks only for a pass or fail status from the device.

Seek

The Seek test sequentially seeks from the lowest to the highest numbered cylinders, and then performs a set of staggered seeks from the lowest cylinder, to the highest cylinder, then from the next lowest, to the next highest, and so on until it checks all cylinders.

Run-To-Run

The Run-To-Run test automatically links the Restore, Seek, Read, and Write/Read tests sequentially, without returning to the diagnostic selection menus between each test. The sequence does stop between tests to display the status of each test.

Write

The Write test writes blocks of data to the disk. The test starts at block zero and writes to the end of the disk, one block at a time.

Chapter 10

Testing Printers

Selecting Printer Diagnostics

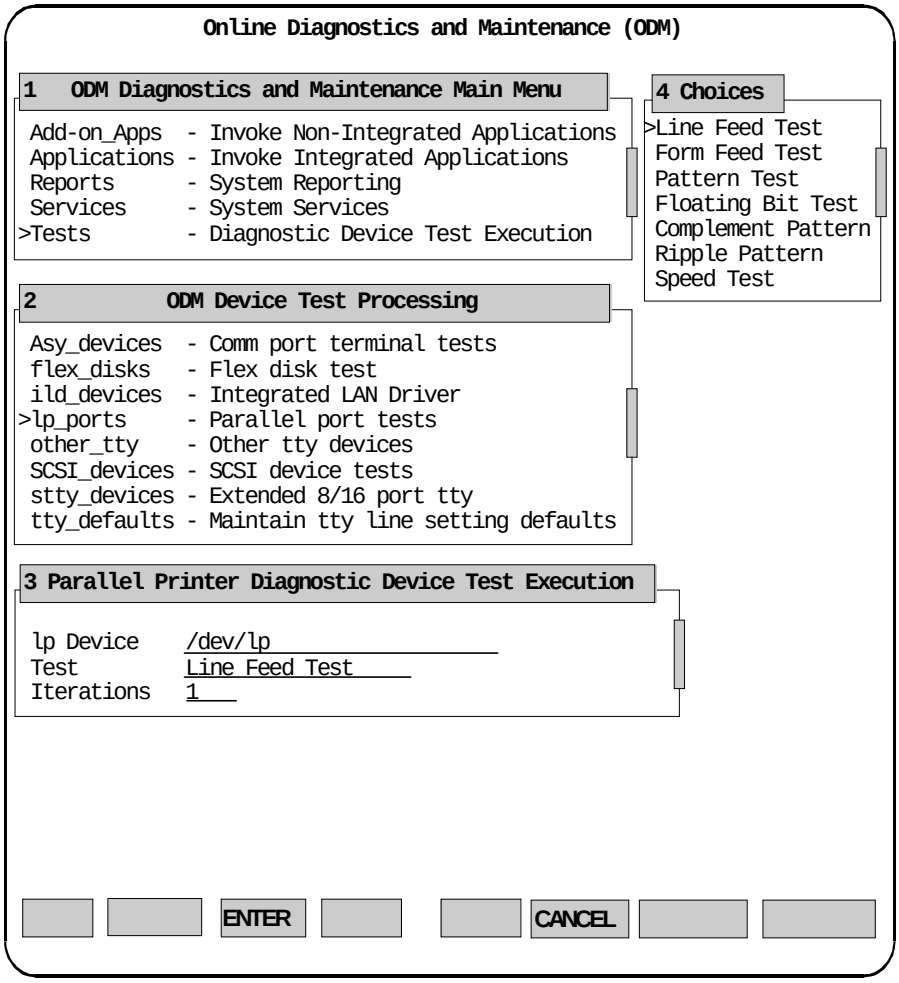
The following sequence of menu selections provides access to the printer diagnostics. An example of the display created by these menu selections is depicted in the next figure.

1. Select <Tests> from the initial menu.
2. Select <lp_ports> from the ODM Device Test Processing menu.
3. If you know the name of the printer, type that name and press ENTER. If you do not know the device name, select CHOICES for a list of available device names, select your choice, and press ENTER to proceed to the test selection.
4. If you know the test to perform, enter the desired test name and press ENTER. If you want to see the available tests for lp ports, press CHOICES, select the desired test , and press ENTER. Available lp device tests are:

   - Ripple Pattern
   - Complement Pattern
   - Floating Bit Pattern
   - Speed Test
   - Line Feed Test
   - Form Feed Test
   - Pattern Test

   Only one test can be selected from the CHOICES menu.
5. If you want to perform the test more than one time, use the arrow keys to move the cursor to the *Iterations* field and enter the number of times you want to perform the test.
6. Press the SAVE function key to activate the Parallel Port Test status window.

Figure 10-1. Parallel Printer Device Test Menu Selections



When you press SAVE in the Parallel Printer Diagnostic Device Test Execution screen, the diagnostic program activates the Parallel Port Device Test status window which is used to start the requested test. This window is used to report the status of the selected test and any errors that occur while the test is in progress. Once a test is started, it can be interrupted by pressing HALT. The “Parallel Port Device Test” section below provides examples of expected output.

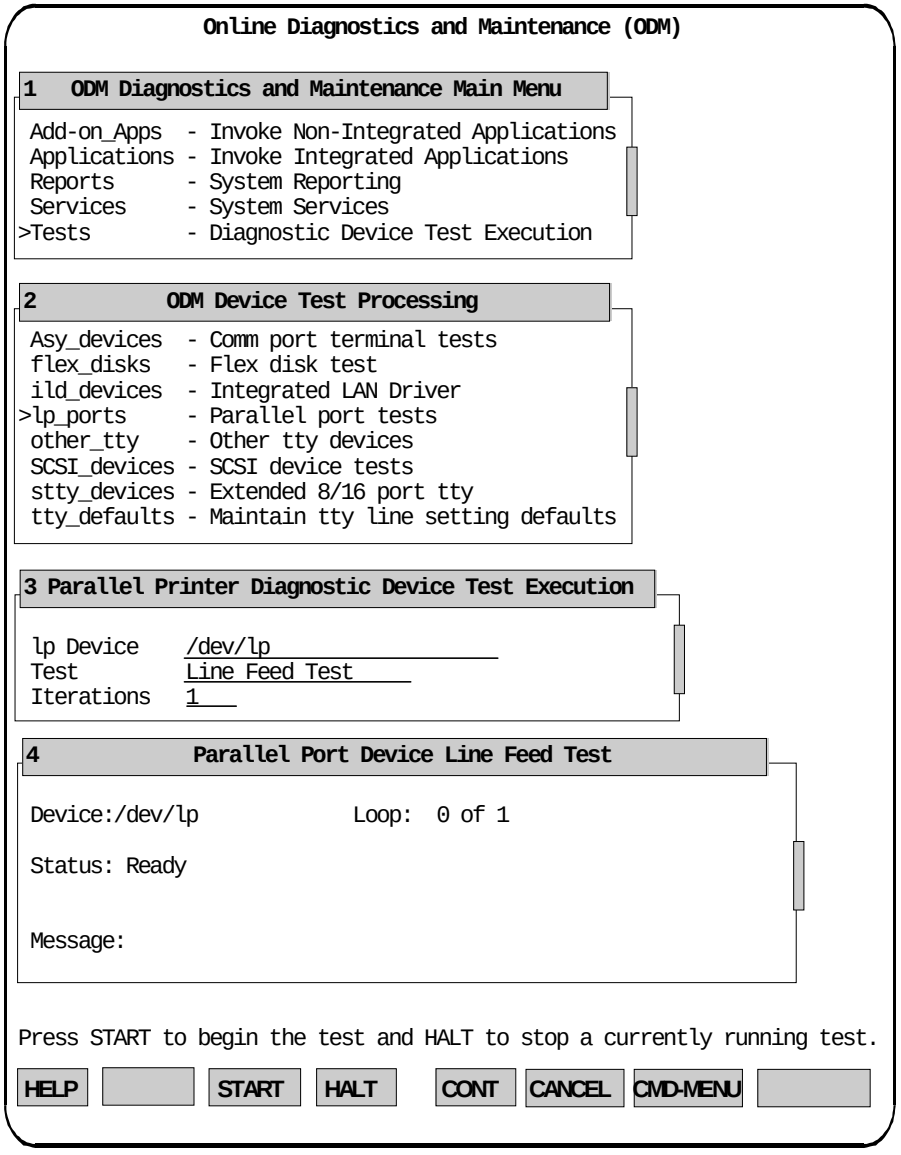
Parallel Port Device Test

The diagnostic program uses one window to report error and status information.

The next figure contains an example of the Parallel Port Device test window before starting the **Line Feed** test. To control test execution, perform the following:

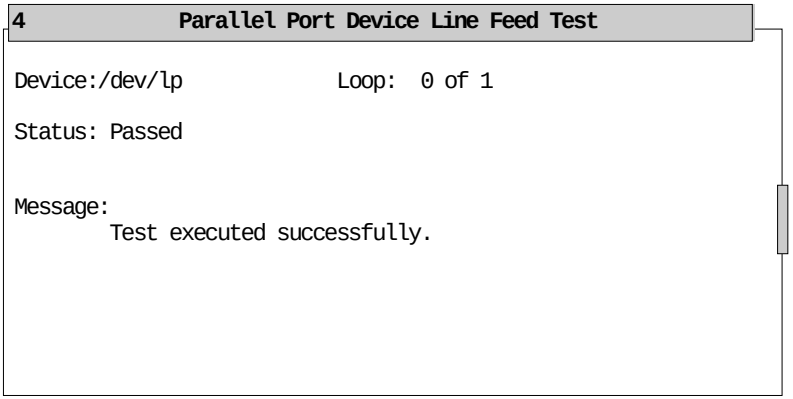
- To **begin** the test. Press START and the test will begin for the ***first*** device selected.
- To **terminate** a test in progress. Press HALT. Because of the speed of some tests (like the Line Feed test), it is not possible to interrupt the test. For these tests, it is only possible to interrupt the test if multiple iterations have been selected.
- To **proceed** to the next device if multiple devices were selected. The test will pause after it completes execution on the previous device. Press CONT to proceed to the next device.

Figure 10-2. Parallel Port Device Test Status Window



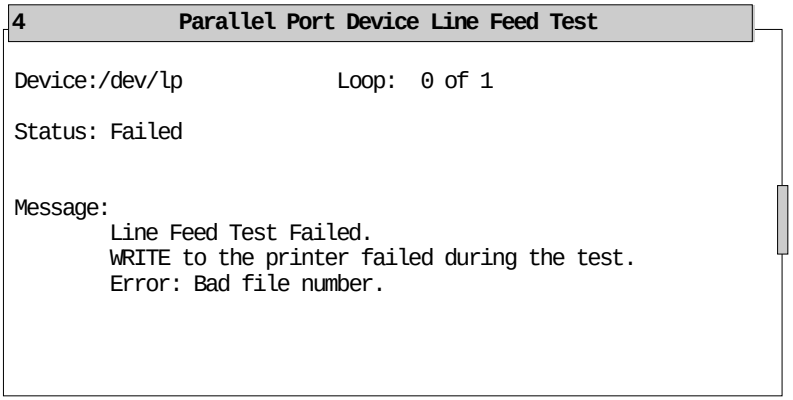
When the test completes, the status window is updated. If the test passed, the status window is similar to the window depicted in the following figure.

Figure 10-3. Parallel Port Device Test: Success



If the test failed, a window similar to the window depicted in the following figure is displayed, which includes a reason for the failure.

Figure 10-4. Parallel Port Device Test: Failure



Parallel Printer Port Test Descriptions

After you select the printer device name, use the CHOICES function key to display the list of available printer tests, as depicted in the figure “Parallel Printer Device Test Menu Selections”. The following tests may be selected from that list:

- Ripple Pattern
- Complement Pattern
- Floating Bit Pattern
- Speed Test
- Line Feed Test
- Form Feed Test
- Pattern Test

Ripple Pattern

The Ripple Pattern test selection prints a selected number of pages of all printable characters. Each line is shifted one character position from the previous line. The test prints 60 lines per page.

Complement Pattern

The Complement Pattern test selection prints a selected number of pages of the characters 0 (hex 30) and 0 (hex 4f). The test prints 60 lines per page.

Floating Bit Pattern

The Floating Bit Pattern test selection prints a selected number of pages of characters with a *1* bit floating through the four least significant bits (called a nibble) of each character. The test prints 60 lines per page.

Speed Test

The Speed test calculates the printer speed by measuring the time required to print 120 lines.

Line Feed Test

The Line Feed test sends one carriage return/line feed sequence to the printer, causing it to space the paper one line.

Form Feed Test

The Form Feed test sends a form feed control character to the printer, causing it to eject one page.

Pattern Test

The Pattern test sends one 80-byte print line to the printer.

Chapter 11

Testing Terminals

Selecting Terminal Diagnostics

The following sequence of menu selections provides access to the terminal diagnostics. Terminal diagnostics are available for various devices, such as the build-in communication ports controlled by the **asy** device driver, or the Extended 8/16 port tty devices controlled by the **stty** device driver. Additionally, a selection is available for other tty device drivers that are not fully integrated with ODM. The next figure depicts an example of the display created by the menu selections for devices controlled by the **asy** device driver. The ODM screens for **asy_devices**, **stty_devices**, and **other_tty** devices all operate in the same manner. The only difference between the operation of the screens is concerned with the CHOICES selection for choosing a device. For **asy_devices** and **stty_devices,** the CHOICES key provides a list of devices supported by that device driver. For **other_tty** devices, no choices are available (you must enter the device name).

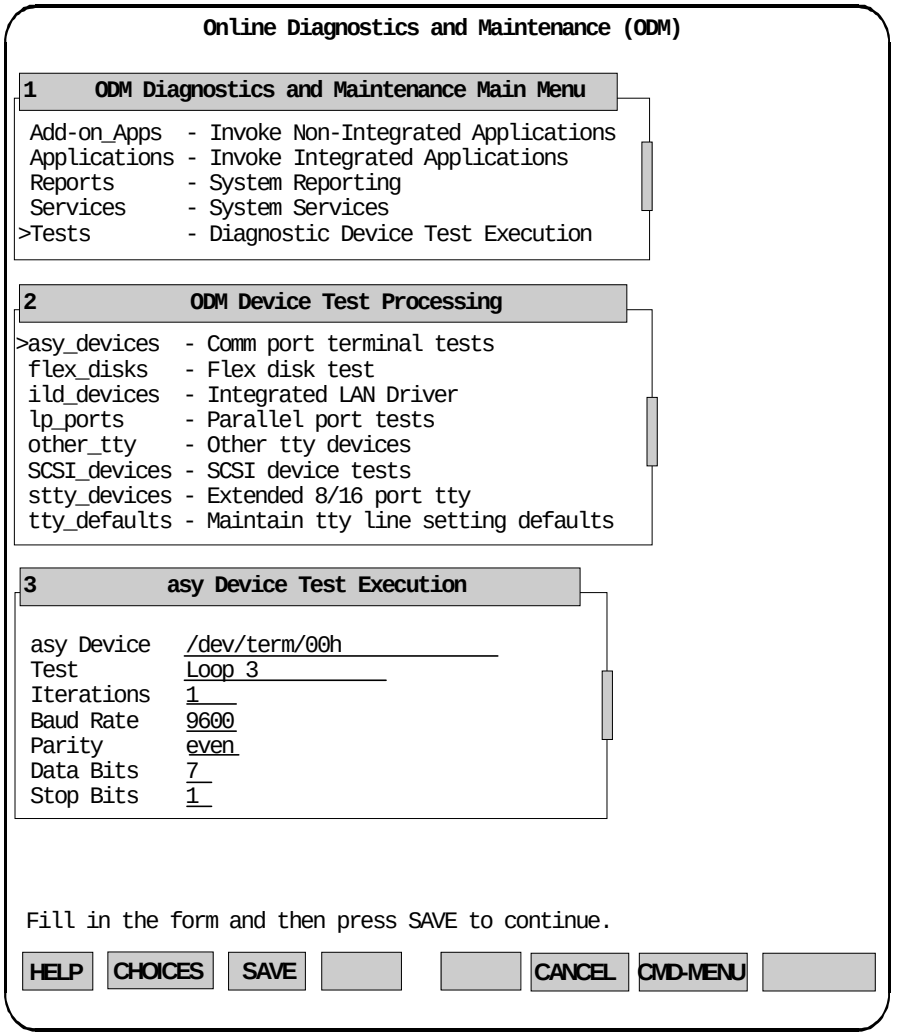
1. Select <Tests> from the initial menu.
2. Select <asy_devices> from the ODM Device Test Processing menu.
3. If you know the name of the terminal, type that name and press ENTER. If you do not know the device name, select CHOICES for a list of available device names, select your choice, and press ENTER to proceed to the Test identification field.
4. If you know the test to perform, enter the desired test name, and press ENTER. If you want to see the available tests for terminal devices (**asy**, **stty**, or **other_tty**), press CHOICES, select the desired test, and press ENTER. Available terminal tests for the device are:

   - Loop 3
   - Pattern
   - Terminal

   Only one test can be selected from the CHOICES menu.
5. If you want to perform the test more than one time, use the arrow keys to move the cursor to the *Iterations* field and enter the number of times you want to perform the test.

6. Use the arrow keys to move the cursor to the *Baud Rate* field, and use the CHOICES key to set the desired baud rate.
7. Use the arrow keys to move the cursor to the *Parity* field, and use the CHOICES key to set the parity to Even, Odd, or None. Repeat this sequence to select the desired number of Data Bits and Stop Bits**.**
8. Press the SAVE function key to activate the selected test characteristics frame:
    - TTY Device Loop 3 Test
    - TTY Device Pattern Test
    - TTY Device Terminal Test

Figure 11-1. Terminal Device Test Menu Selections



When you press SAVE in the asy Device Test Execution screen, the diagnostic program displays the asy Device Test status window which is used to start the requested test. This window is used to report the status of the selected test and any errors that occur while the test is in progress. Once a test is started, it can be interrupted by pressing HALT. The “TTY Device Test” section below provides examples of expected test output.

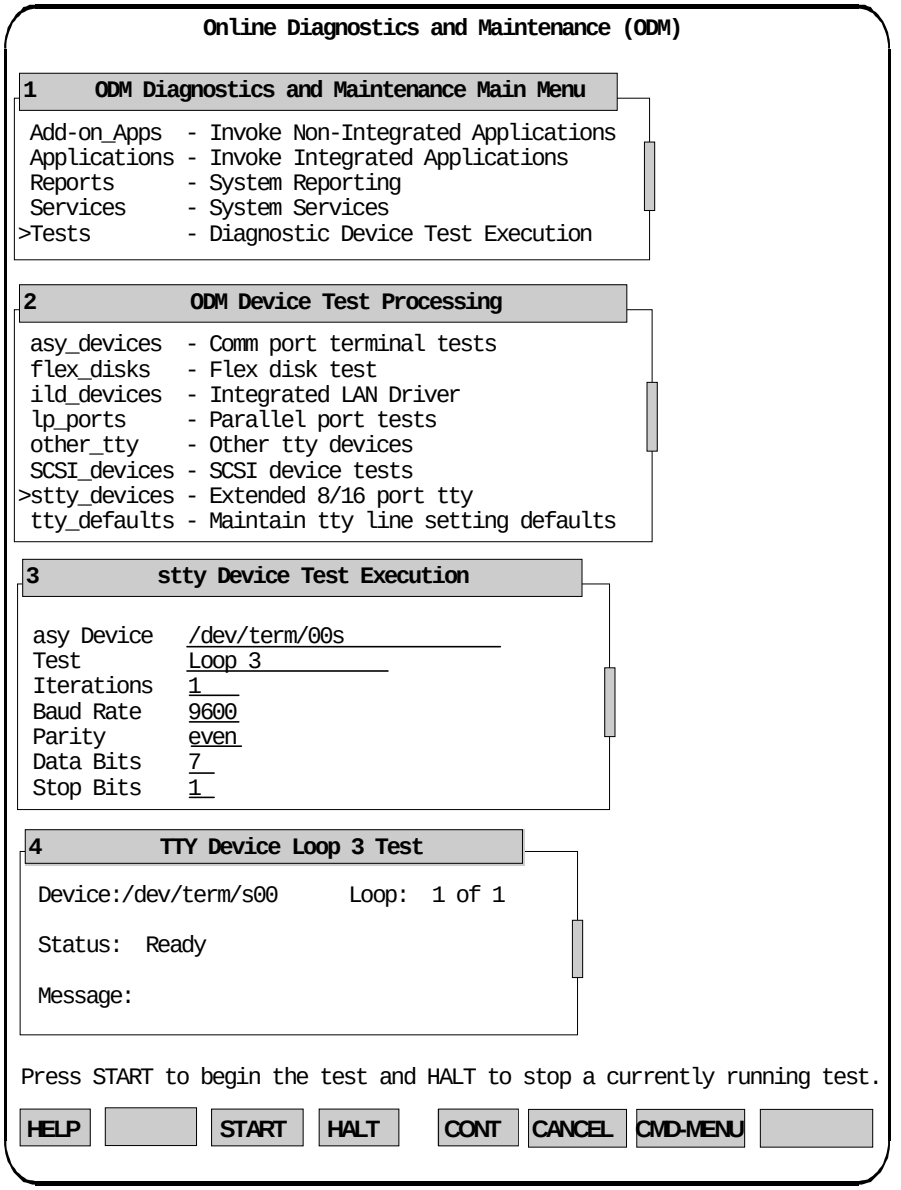
TTY Device Test

The diagnostic program uses one window to report error and status information for the selected test.

The next figure contains an example of the TTY Device test window before starting the **Loop 3** test. To control test execution, perform the following:

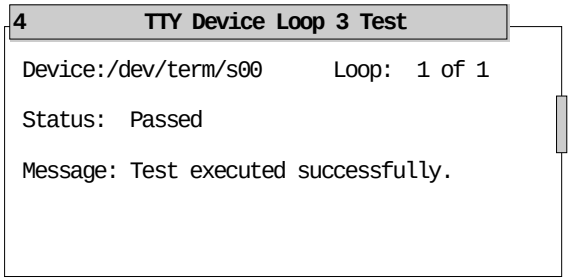
- To **begin** the test. Press START and the test will begin for the ***first*** device selected.
- To **terminate** a test in progress. Press HALT.
- To **proceed** to the next device if multiple devices were selected. The test will pause after it completes execution on the previous device. Press CONT to proceed to the next device.

Figure 11-2. TTY Device Test Status Window



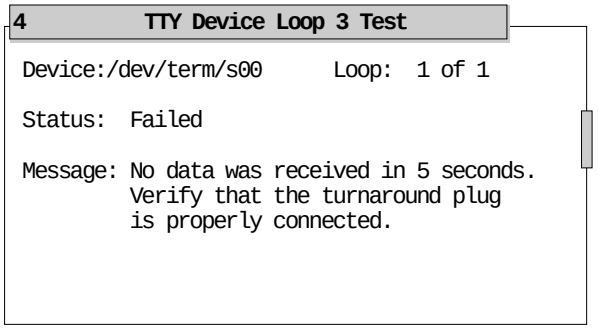
When the test completes, the status window is updated. If the Loop 3 test passed, the status window will look similar to the window depicted in the following figure.

Figure 11-3. TTY Device Loop 3 Test: Success



If the Loop 3 test failed, a window similar to the window depicted in the next figure is displayed, which includes a reason for the failure.

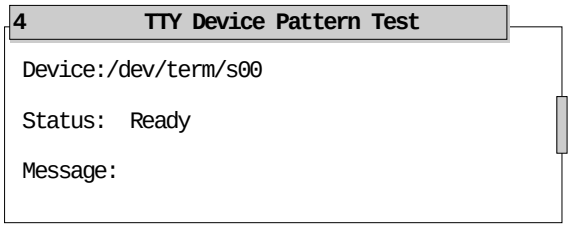
Figure 11-4. TTY Device Loop 3 Test: Failure



TTY Device Pattern Test

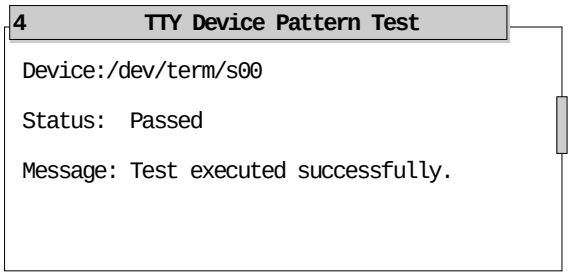
The following figure is an example of the TTY Device Pattern Test status window before starting the test.

Figure 11-5. TTY Device Pattern Test Status Window



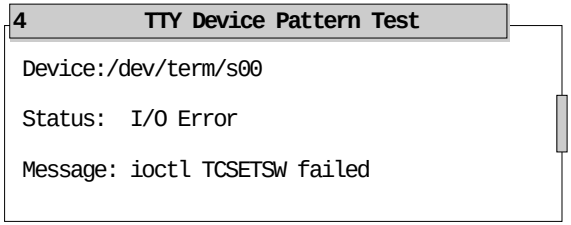
When the test completes, the status window is updated. If the test passed, the status window will look similar to the window depicted in the following figure.

Figure 11-6. TTY Device Pattern Test: Success



If the test failed, a window similar to the window depicted in the next figure is displayed, which includes a reason for the failure.

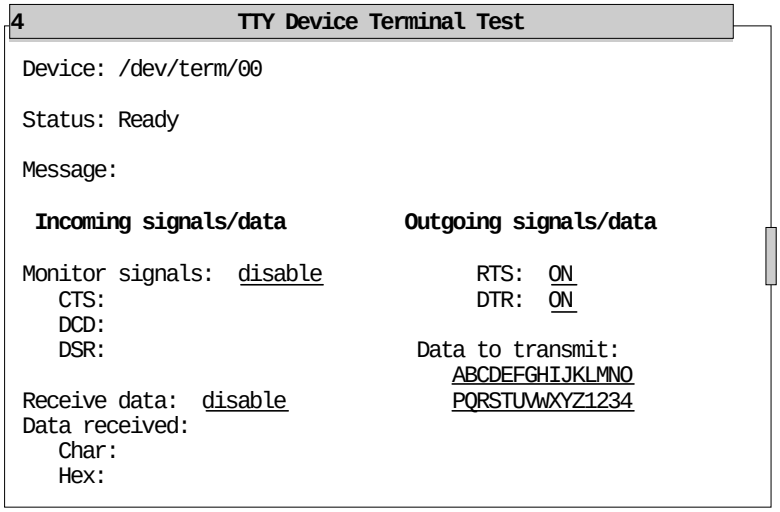
Figure 11-7. TTY Device Pattern Test: Failure



TTY Device Terminal Test

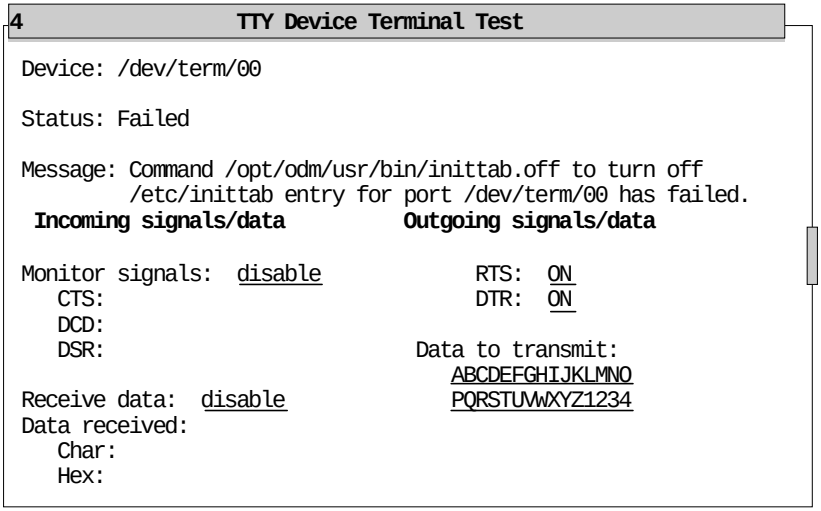
The next figure is an example of the TTY Device Terminal Test status window before starting the test.

Figure 11-8. TTY Device Terminal Test



If the test failed, a window similar to the window depicted in the following figure is displayed, which includes a reason for the failure.

Figure 11-9. TTY Device Terminal Test Failure



Terminal Test Descriptions

After you select the terminal device name as shown in the “Selecting Terminal Diagnostics” section, use the CHOICES function key to display the list of available terminal tests. The following tests may be selected from that list:

- Loop 3 Test
- Pattern Test
- Terminal Test

Loop 3 Test

The Loop 3 test verifies the operation of the local DTE logic and the local DCE logic.

The test requires a turnaround test connector on the selected port or a local modem that has loop 3 turnaround capabilities.

The test transmits and receives a data pattern one time.

Pattern Test

The Pattern test transmits a continuous data pattern over the communication link connected to the selected port.

The user must watch the terminal for the pattern. If the pattern is not printing then the test failed and the user must exit.

Terminal (TTY Devices Only) Test

The Terminal test verifies that a selected TTY terminal can receive and transmit data correctly and permits the checking and/or changing of basic communications control signals associated with TTY terminals.

The terminal from which the diagnostic test is activated is called the command terminal, and the terminal selected for testing is called the test terminal. The test does not permit you to test the same device that you are using to activate the test.

To verify that the test terminal receives data properly, use the command terminal to highlight the TRANSMIT, and enter a message on the command terminal keyboard. Observe the test terminal to see if it displays the same message that you entered at the command terminal.

To verify that the test terminal transmits data properly, use the command terminal to highlight the RECEIVE, and then enter a message on the test terminal keyboard. Observe the command terminal to see if it displays the same message that you entered at the test terminal.

Terminate the Terminal test by selecting HALT or CANCEL at the command terminal. The test can not be terminated at the test terminal.

The Terminal test may also be performed on serial printers connected to TTY ports.

Chapter 12

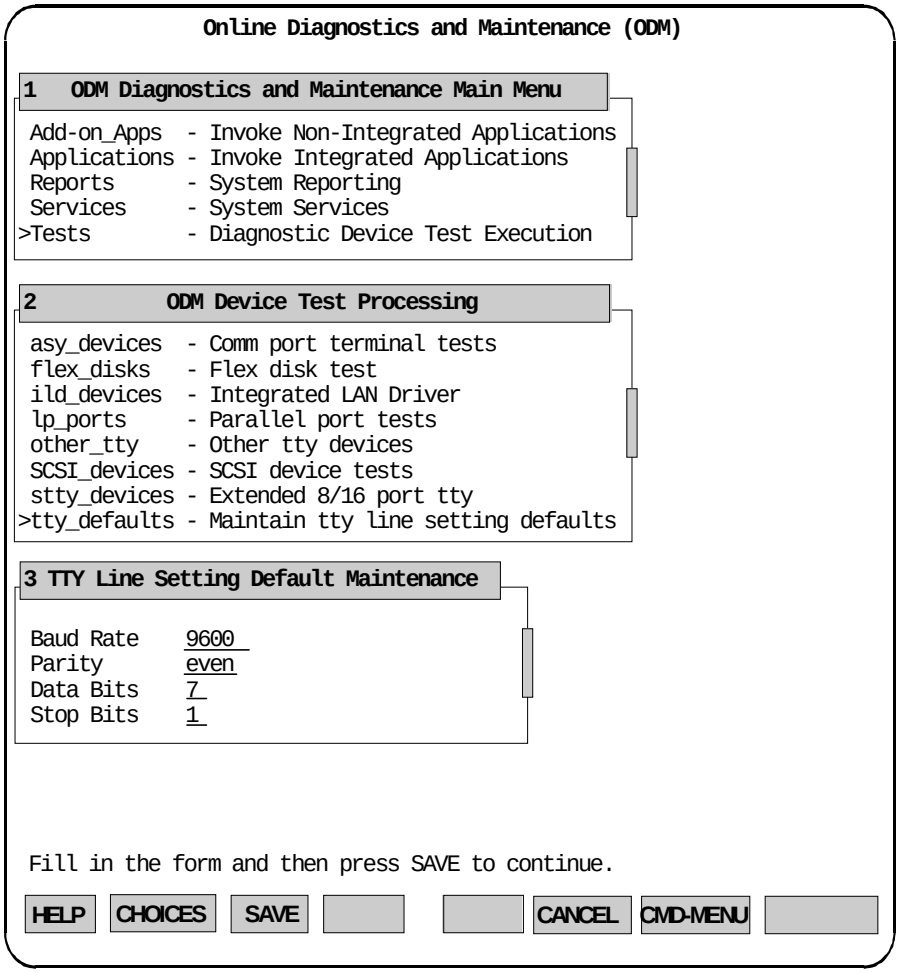
Maintain Default TTY Line Characteristics

Selecting TTY Default Line Characteristics

The following sequence of menu selections provides access to the **tty defaults**, used to maintain defaults for tty line characteristics used for testing tty devices. An example of the display created by these menu selections is depicted in the next figure.

1. Select <Tests> from the initial menu.
2. Select <tty_defaults> from the ODM Device Test Processing menu.
3. If you know the default baud rate to use, enter that rate and press ENTER. If you do not know the default baud rate, select CHOICES for a list of available baud rates, select your choice, and press ENTER to proceed to the Parity field.
4. If you know the parity value for the line, enter the desired value and press ENTER . If you want to see the available parity settings, press CHOICES, select the desired parity value, and press ENTER.
5. If you know the data bits value, enter the desired value and press ENTER. If you want to see the available data bit settings, press CHOICES, select the desired data bit value, and press ENTER.
6. If you know the stop bit value, enter the desired value and press ENTER. If you want to see the available stop bit settings, press CHOICES, select the desired data bit value, and press ENTER.
7. Press SAVE to set the defaults for tty line characteristics.

Figure 12-1. TTY Line Setting Default Maintenance



When SAVE is pressed, the values entered on the screen are saved. These values are then used as defaults in the ODM tty test screens for asy_devices, stty_devices and other_tty devices.

Chapter 13

Testing HDLC WANs

Accessing HDLC WAN Diagnostics

In-Service (Level 2) diagnostics permit you to perform basic HDLC WAN tests and diagnose HDLC WAN activities. These in-service diagnostic test programs interact with the HDLC driver and the MPCA through various **ioctl** commands and driver utilities**.**

You may perform most diagnostic functions though HDLC driver utilities, but the Driver interface trace and Retrieve attached count diagnostics can only be performed through UNIX **ioctl**() calls.

Command Line Tests

You can perform all of the HDLC WAN diagnostic functions described in this chapter by using command line entries.

For additional information about performing HDLC WAN diagnostics from the command line, refer to the *Installation, Configuration and Diagnostics* book(s) for the hardware controllers (or adapters) that provide the HDLC interface with the operating system software.

odm Menu Tests

You can perform the following functions through the **odm** Online Diagnostics and Maintenance menus**:**

- Loop 1 test
- Loop 2 test
- Loop 4 test
- Modem Status
- Adapter Status
- Reset Adapter
- Link Test Statistics
- Clear Test Statistics

You can also use the **odm** menus to display help information about running data capture, viewing tallies, viewing version number, and displaying driver status.

Using the Maintenance Menus

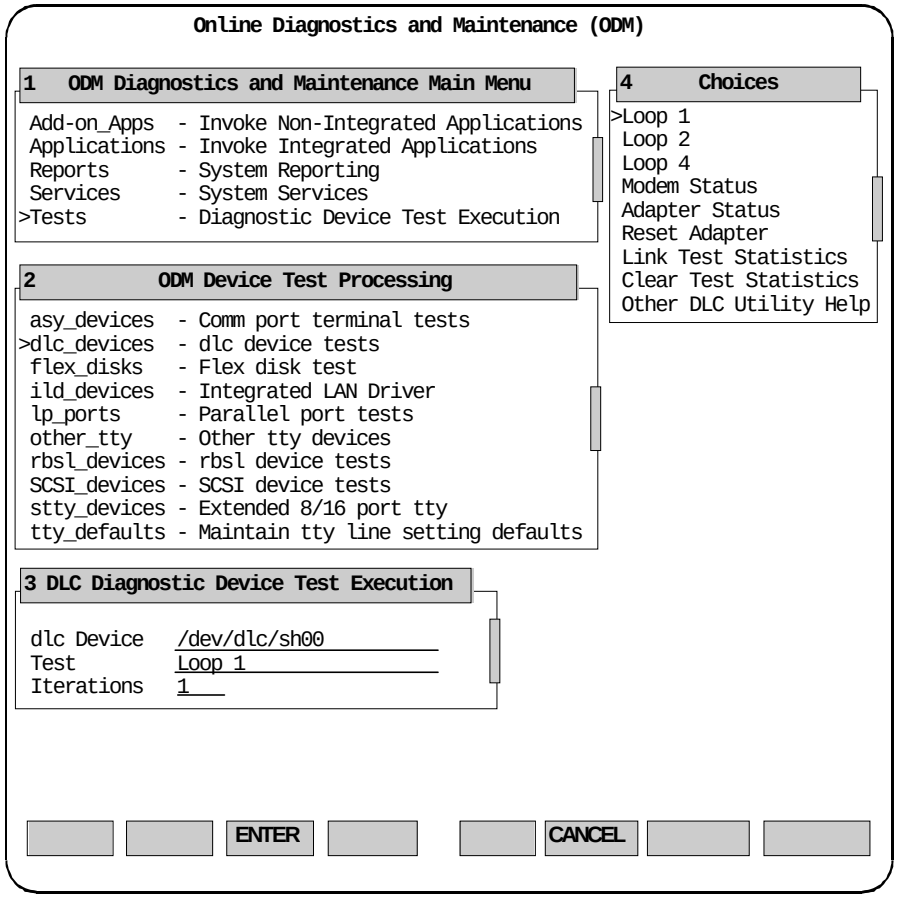
You can perform some of the most commonly used HDLC diagnostics and troubleshooting tasks from the **odm** maintenance menus**.**

The following sequence of menu selections permits you to select HDLC WAN tests through the **odm** menus. An example of the display created by these menu selections is depicted in the next figure. These menu selection steps are the same for all HDLC WAN tests that may be selected through the menus.

1. Select <Tests> from the initial menu.
2. Select <dlc_devices> from the ODM Device Test Processing menu.
3. If you know the name of the HDLC WAN device, type that name and press ENTER. If you do not know the device name, select CHOICES for a list of available device names, select your choice, and press ENTER to proceed to the *Test identification* field. HDLC devices have names in the form *sh0x*, where *x* is 1 through 4.
4. If you know the name of the test to perform, enter the desired test name, and press ENTER. If you want to see the tests available from the **odm** menus, press CHOICES, select the desired test, and press ENTER.

   Only one test can be selected from the CHOICES menu.
   The remaining sections of this chapter provide information about the HDLC WAN tests.
5. If you want to perform the test more than one time, use the arrow keys to move the cursor to the *Iterations* field and enter the number of times you want to perform the test.
6. Press the SAVE function key to activate the selected test for the selected number of iterations.

Figure 13-1. HDLC WAN Test Menu Selections



Performing the Loop1 Test

The **loop1** test is the simplest device test. It verifies the adapter's internal UART. This test generates its own internal clocking and requires no external hardware.

The **loop1** test generates a test pattern which is sent and looped back either under software control or under manual control. It then compares the received pattern to the pattern it sent.

Using the Command Line

Use the following procedure to run the **loop1** test from the command line prompt:

1. Make sure the directory */opt/net/mpd* is in your search path.
2. Create a file containing your test parameters. You must specify values for the parameters shown in the following table. The parameters must be entered in a KEYWORD=*value* format.

Table 13-1. Loop 1 Parameters

KEYWORD	VALUE
BITRATE	Specifies the bitrate to be used for loop tests 1, 2 and 4. The acceptable bitrate values are 1200, 2400, 4800, 9600, and 19200.
ITERATIONS	Specifies the number of loop tests to run before terminating. Values range from 1 to 65000. The default value is 1.
PATTERN	Specifies a string of characters that are assembled into a pattern to loop. If the string begins with a single quote mark, the pattern is assembled as an ASCII string. A terminating quote is not needed. If the string does not begin with a single quote, the characters should consist of pairs of hexadecimal digits. The default string value is 00551199.

KEYWORD	VALUE
NRZI	Specifies whether or not to use NRZI encoding. Acceptable values are Y if NRZI encoding is used and N if it is not.

If you do not create this file, **loop1** will read the options from standard input.

3. Enter the following command to begin testing:

```
# /opt/net/mpd/loop1 -i parameter_file -l device
```

Valid devices are */dev/mpd00* through */dev/mpd03*.

For each iteration, **loop1** displays the message "LOOP TEST SUCCESSFUL" if the test iteration was successful or "LOOP TEST FAILED" if the test iteration failed.

Using the Menus

You may use the **odm** menus to run the **loop1** test by selecting the **loop1** test during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter.

Performing the Loop2 Test

The **loop2** test verifies the adapter UART. In addition, it also tests the modem cable, modem circuitry, and the telephone line. This test requires a synchronous modem or similar device to provide external clocking. Before running this test, you must manually create an electrical loopback by either throwing a modem switch or by placing a loopback connection on the modem's analog interface. After testing, be sure to remove the loopback before resuming normal communications.

The **loop2** test generates a test pattern which is sent and looped back either under software control or under manual control. It then compares the received pattern to the pattern it sent.

Using the Command Line

Use the following procedure to run the **loop2** test from the command line prompt:

1. Make sure the directory */opt/net/mpd* is in your search path.
2. Create a file containing your test parameters. You must specify values for the parameters shown in the following table. The parameters must be entered in a KEYWORD=*value* format.

Table 13-2. Loop 2 Parameters

KEYWORD	VALUE
BITRATE	Specifies the bitrate to be used for loop tests 1, 2 and 4. The acceptable bitrate values are 1200, 2400, 4800, 9600, and 19200.
ITERATIONS	Specifies the number of loop tests to run before terminating. Values range from 1 to 65000. The default value is 1.

KEYWORD	VALUE
PATTERN	Specifies a string of characters that are assembled into a pattern to loop. If the string begins with a single quote mark, the pattern is assembled as an ASCII string. A terminating quote is not needed. If the string does not begin with a single quote, the characters should consist of pairs of hexadecimal digits. The default string value is 00551199.
NRZI	Specifies whether or not to use NRZI encoding. Acceptable values are Y if NRZI encoding is used and N if it is not.

If you do not create this file, **loop2** will read the options from standard input.

3. Enter the following command to begin testing:

```
# /opt/net/mpd/loop2 -i parameter_file -l device
```

Valid devices are */dev/mpd00* through */dev/mpd03*.

For each iteration, **loop2** displays the message "LOOP TEST SUCCESSFUL" if the test iteration was successful or "LOOP TEST FAILED" if the test iteration failed.

Using the Menus

You may use the **odm** menus to run the **loop2** test by selecting the **loop2** test during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter.

Performing the Loop4 Test

The **loop4** test verifies the adapter UART and the RS-232 interface. This test does not require external clocking, but you must place a loopback connector on the RS-232 interface before running the test. This test can also check modem cable continuity when the loopback connector temporarily replaces the modem. Be sure to remove this connector when testing is complete.

The **loop4** test generates a test pattern which is sent and looped back either under software control or under manual control. It then compares the received pattern to the pattern it sent.

Using the Command Line

Use the following procedure to run the **loop4** test from the command line prompt:

1. Make sure the directory */opt/net/mpd* is in your search path.
2. Create a file containing your test parameters. You must specify values for the parameters shown in the following table. The parameters must be entered in a KEYWORD=*value* format.

Table 13-3. Loop 4 Parameters

KEYWORD	VALUE
BITRATE	Specifies the bitrate to be used for loop tests 1, 2 and 4. The acceptable bitrate values are 1200, 2400, 4800, 9600, and 19200.
ITERATIONS	Specifies the number of loop tests to run before terminating. Values range from 1 to 65000. The default value is 1.

KEYWORD	VALUE
PATTERN	Specifies a string of characters that are assembled into a pattern to loop. If the string begins with a single quote mark, the pattern is assembled as an ASCII string. A terminating quote is not needed. If the string does not begin with a single quote, the characters should consist of pairs of hexadecimal digits. The default string value is 00551199.
NRZI	Specifies whether or not to use NRZI encoding. Acceptable values are Y if NRZI encoding is used and N if it is not.

If you do not create this file, **loop4** will read the options from standard input.

3. Enter the following command to begin testing:

```
# /opt/net/mpd/loop4 -i parameter_file -l device
```

Valid devices are */dev/mpd00* through */dev/mpd03*.

For each iteration, **loop4** displays the message "LOOP TEST SUCCESSFUL" if the test iteration was successful or "LOOP TEST FAILED" if the test iteration failed.

Using the Menus

You may use the **odm** menus to run the **loop4** test by selecting the **loop4** test during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter.

Performing the Pingpong Test

The pingpong test transmits a test pattern to the remote station using the SDLC TEST command. Your local station must be in the role of SDLC primary and the remote station must be in the role of SDLC secondary.

Using the Command Line

Enter the following command to run the **pingpong** test**:**

```
# /opt/net/dlc/pingpong -P -D device
```

This test sends the default test pattern on the device you specify. HDLC devices have the form */dev/dlc/sh0n* where *n* identifies the line number and ranges from zero through three.

The default pattern for the pingpong test is the following 16 bytes:

```
0F 64 AE 0A
0F 64 AE 0A
0F 64 AE 0A
0F 64 AE 0A
```

You can use the -i *test_pattern_file* option to specify a different test pattern. You can also use the -o *output_file* option to write output to a file instead of standard output.

The **pingpong** test only displays a message if the test fails.

Using the Menus

You can not run the **pingpong** test from the **odm** maintenance menus.

Displaying Adapter Status

Displaying adapter status produces a display like the following example:

```
----------------------------------------
Adapter State              = [ READY/DEAD ]
Micro Channel Slot         = 1
Adapter Version            = 01.01-A
Adapter IRQ Level          = 10
Adapter Queue Size         = 5032

Adapter Register Addresses:
Base I/O Address           = 0x380:
Message-In Register        = 0x382:
Message-Out Register       = 0x383:
Control Register           = 0x384:
Status Register            = 0x386:
Local Address Pointe       = 0x385:
Remote Address Pointer     = 0x387:
Reset Control Register     = 0x38c:
```

The *Adapter State* indicates the status of the adapter. If the status is READY, the adapter can be used. If the status is DEAD, the adapter failed level 0 diagnostics and cannot be accessed by any driver.

The *Micro Channel Slot* number shows the bus slot occupied by the adapter.

The *Adapter Version* is the version number of the adapter firmware.

The *Adapter IRQ Level* is the adapter's interrupt request level and should match the interrupt level specified in the UNIX kernel configuration file */etc/conf/sdevice.d/mpd*.

The *Adapter Queue Size* shows the memory capacity of the adapter in bytes.

The *Adapter Register Addresses* shown should agree with the addresses configured for the MPCA with the Micro Channel Reference Disk tool. The address labeled *Base I/O Address* is configured with the reference disk and the remaining addresses are based on it.

Using the Command Line

Enter the following command to display the adapter status:

```
# /opt/net/mpd/query -l device
```

The *device* argument specifies the MPCA device to query. Valid devices are */dev/mpd00* through */dev/mpd03*.

Using the Menus

You may use the **odm** menus to display the current adapter status by selecting **Adapter Status** during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter.

Displaying HDLC Line Status

Displaying the HDLC line status (modem status) produces a display like the following example:

```
/dev/dlc/sh00:
Line status is disconnected
Station status is disconnected
00000 pending input I-field’s
00000 pending output I-field’s
00000 outstanding I-field’s
00000 pending output buffers
00000 receive buffers
```

The *line status* describes the electrical interfaces as one of the following:

- Connected
- Connecting
- Resetting
- Disconnecting
- Disconnected

The *station status* marks this station’s protocol level contact with the remote station. *Pending input* values indicate the number of input buffers and BTUs to be read by the applications. *Pending output* values indicate the number of output buffers and BTUs to be sent to the other station. The *receive buffer* count indicates the actual number of buffers currently reserved.

Using the Command Line

Enter the following command to display the HDLC electrical, protocol, and buffering statuses:

```
# /opt/net/dlc/hdstat -D device
```

HDLC devices have the form */dev/dlc/sh0n* where *n* identifies the line number and ranges from zero through three. You may also specify the keyword **all** as the device name to display HDLC line status for all existing adapters. If you do not specify a device name, the status is displayed for */dev/dlc/sh00* only.

You may use the -o *output_file* argument to write the output to a file instead of displaying the information on your screen.

Using the Menus

You may use the **odm** menus to display the current HDLC line modem status by selecting **Modem Status** during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter.

Displaying X.25 Statistics

Displaying the X.25 link level statistics produces a display like the following example:

```
/dev/dlc/sh00:
00000 I frames transmitted
00000 I frames received
00000 RR frames transmitted
00000 RR frames received
00000 RNR frames transmitted
00000 RNR frames received
00000 REJ frames transmitted
00000 REJ frames received
00000 retrans'd frames (all types)
00000 FCS errors
00000 receive side errors
00000 receiver overruns
00000 transmitter underruns
```

These statistics are broken down by HDLC instruction type. They are defined even when you are using SNA/SDLC instead of X.25.

Using the Command Line

Enter the following command to display the X.25 link level statistics on the number of frames sent and received:

```
# /opt/net/dlc/hdx25 -D device
```

HDLC devices have the form */dev/dlc/sh0n* where *n* identifies the line number and ranges from zero through three. You may also specify the keyword **all** as the device name to display X.25 statistics for all existing adapters. If you do not specify a device name, the status is displayed for */dev/dlc/sh00* only.

You may use the -o *output_file* argument to write the output to a file instead of displaying the information on your screen.

Using the Menus

You can not use the **odm** menus to display X.25 statistics.

You may use the **odm** menus to display help information about X.25 statistics by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other DLC Utility Help> from the Test type CHOICES menu
<X.25 Tallies>
```

Displaying Adapter Statistics

Displaying adapter statistics produces a display like the following example:

```
/dev/dlc/sh00:
000 nonproductive receives
000 idle timeouts
000 write retries
000 receiver overruns
000 transmitter underruns
000 connect problems
000 FCS errors
000 aborts received
000 command rejects sent
000 DCE errors
000 write timeouts
000 bad adapter statuses
000 adapter checks
```

Using the Command Line

Enter the following command to display the adapter statistics about framing errors, overruns, retries, and timeouts:

```
# /opt/net/dlc/hdca -D device
```

HDLC devices have the form */dev/dlc/sh0n* where *n* identifies the line number and ranges from zero through three. You may also specify the keyword **all** as the device name to display adapter statistics for all existing adapters. If you do not specify a device name, the status is displayed for */dev/dlc/sh00* only.

You may use the -o *output_file* argument to write the output to a file instead of displaying the information on your screen.

Using the Menus

You can not use the **odm** menus to display adapter statistics.

You may use the **odm** menus to display help information about adapter statistics by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other DLC Utility Help> from the Test type CHOICES menu
```

```
<Adapter Tallies>
```

Displaying Link Test Statistics

Displaying link test statistics produces a display like the following example:

```
/dev/dlc/sh00:
00000 TEST frames received
00000 TEST frames sent
```

Using the Command Line

Enter the following command to display the number of SDLC TEST frames sent and received:

```
# /opt/net/dlc/hdlt -D device
```

HDLC devices have the form */dev/dlc/sh0n* where *n* identifies the line number and ranges from zero through three. You may also specify the keyword **all** as the device name to display link test statistics for all existing adapters. If you do not specify a device name, the status is displayed for */dev/dlc/sh00* only.

You may use the -o *output_file* argument to write the output to a file instead of displaying the information on your screen.

Using the Menus

You may use the **odm** menus to display the link test statistics by selecting **Link Test Statistics** during step 4 of the procedure described in the “Using the Maintenance Menus” section of this chapter.

To clear link test statistics through **odm,** select the **Clear Test Statistics** during step 4 of the procedure described in the “Using the Maintenance Menus” section of this chapter.

Displaying STREAMS Statistics

Displaying STREAMS statistics produces a display like the following example:

```
/dev/dlc/sh00:
00000 No STREAMS-buffers conditions
00000 STREAMS flow control conditions
```

When the driver cannot allocate a STREAMS buffer for its use, it tallies a STREAMS “no-buffer” condition. The driver also tallies STREAMS flow-control conditions when they are encountered upstream.

Using the Command Line

Enter the following command to display the STREAMS conditions (statistics) encountered by the HDLC driver:

```
# /opt/net/dlc/hdstr -D device
```

HDLC devices have the form */dev/dlc/sh0n* where *n* identifies the line number and ranges from zero through three. You may also specify the keyword **all** as the device name to display STREAMS statistics for all existing adapters. If you do not specify a device name, the status is displayed for */dev/dlc/sh00* only.

You may use the -o *output_file* argument to write the output to a file instead of displaying the information on your screen.

Using the Menus

You can not use the **odm** menus to display STREAMS statistics.

You may use the **odm** menus to display help information about STREAMS statistics by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other DLC Utility Help> from the Test type CHOICES menu
<STREAMS Tallies>
```

Resetting the Adapter

Resetting the adapter produces a display like the following example:

```
----------------------------------------
Reset of adapter [ 0/1 ] [ SUCCESSFUL/
FAILED ]

Level 0 diagnostic byte 1 = 0xff
   Scan List/Direction Test [
Passed/Failed ]
   Line Turnaround Test
[ Passed/Failed ]
   EPROM CCITT FCS Test
Passed/Failed ]
   Memory Access Test
[ Passed/Failed ]
   Memory Retention Test
[ Passed/Failed ]
   Register Access Test
[ Passed/Failed ]
   Register Retention Test
[ Passed/Failed ]
Condition Code Test
[ Passed/Failed ]

level 0 diagnostic byte 2 = 0x40
          Real Time Clock Test
[ Passed/Failed ]

Adapter State    =   READY/DEAD ]
Micro Channel Slot      = 1
Adapter Version     = 01.01-A
Adapter IRQ Level       = 10
Adapter Queue Size      = 5032
----------------------------------------
```

The *Adapter State* indicates the status of the adapter. If the status is READY, the adapter can be used. If the status is DEAD, the adapter failed level 0 diagnostics and cannot be accessed by any driver.

The *Micro Channel Slot* number shows the bus slot occupied by the adapter.

The *Adapter Version* is the version number of the adapter firmware.

The *Adapter IRQ Level* is the adapter's interrupt request level and should match the interrupt level specified in the UNIX kernel configuration file */etc/conf/sdevice.d/mpd*.

The *Adapter Queue Size* shows the memory capacity of the adapter in bytes.

Using the Command Line

Enter the following command to reset an MPCA device (adapter), execute level 0 diagnostics, and display the results of the diagnostic tests:

```
# /opt/net/mpd/reset -l device
```

The *device* argument specifies the MPCA device to reset. Valid devices are */dev/mpd00* through */dev/mpd03*.

Using the Menus

You may use the **odm** menus to reset the adapter by selecting the **Reset Adapter** option during step 4 of the procedure described in the “Using the Maintenance Menus” section of this chapter.

Reading Tally Values

The following is an example of how tallies are displayed when you read tally values for an HDLC device.

```
Tallies for device /dev/mpd00
=============================

ID of last driver to access tallies        : -x11
 1.I frames received                       : 16
 2.I frames sent                           : 338
 3.I frames sent and receive               : 354
 4.Frames received with FCS error          : 0
 5.I frames discarded with wrong
   sequence number                         : 0
 6.Frames discarded as too short           : 0
 7.Frames discarded due to buffer
   overrun                                 : 0
 8.Frames discarded due to critical
   buffering                               : 76
 9.Frames received containing framing
   errors                                  : 0
10.Frames received with invalid
   station address                         : 0
11.Times the local station indicated
   busy                                    : 0
12.Aborted frames received                 : 0
13.I frames retransmitted                  : 0
14.Transmitter underruns                   : 0
15.Times the remote station indicated
   busy                                    : 0
16.Remote station busy timeout
   occurrences                             : 0
17.Timeouts following successful
   connection                              : 0
18.Locally-initiated station
   connections                             : 0
19.Locally-initiated station resets        : 0
20.Remote-initiated station resets         : 0
21.Remote-initiated station
   disconnections                          : 0
22.Locally-initiated station
   disconnections                          : 0
23.Remote occurrences of FRMR
   condition                               : 0
```

```
24.Local occurrences of FRMR
   condition                              : 0
25.Frames discarded with bad P/F bit
   state                                  : 0
26.DSR losses after successful
   connection                             : 0
27.CTS losses                             : 0
28.Remote carrier drops                   : 0
29.Line connection failures               : 0
30.MPCA failures                          : 0
```

Using the Command Line

Enter the following command to display the standard set of network tallies that are maintained for all WAN drivers:

```
# /opt/net/mpd/tally -p1 -l device
```

The **-p1** option specifies that you want to read HDLC protocol tally values. The *device* argument specifies the name of the device for which you want to see the tallies. Possible values for the device name are of the form **sh0n**, where the value of *n* is 0-3 and represents the line used.

You may also use the **-c** argument to reset tally values to zero after displaying them.

The following table describes the tally ID numbers displayed on the screen.

Table 13-4. Tally ID Descriptions

ID	Tally	Description
1	Error-free I-frames received and accepted	This tally is normally non-zero and indicates that frames are being received from the other station.
2	I-frames sent	This tally indicates the number of frames accepted and acknowledged by the other station. Non-zero values are expected and indicate the amount of output traffic.
3	I-frames sent and received	This field represents the total of tallies 1 and 2 above.

ID	Tally	Description
4	Frames received with FCS errors	This tally is normally zero or very low. It counts frames received with transmission errors, such as noisy lines.
5	I-frames discarded with wrong sequence number	Non-zero values of this tally indicate reception difficulties. Frames which are discarded by this station must be retransmitted by the other.
6	Frames discarded as too short	Non-zero values indicate reception problems. They indicate that the remote station is sending frames which are too short, or that line noise is causing the false appearance of a frame.
7	I-frames discarded due to buffer overrun	Non-zero values indicate that aggregate line speeds are too great for the multiprotocol adapter, or too few buffers have been configured.
8	I-frames discarded due to critical buffering	Non-zero values indicate that either too few receive buffers have been specified, or the remote station is not stopping I-frame transmission when this station indicates busy.
9	Frames received with invalid residue count	Non-zero values indicate the reception of frames whose information length is not a multiple of eight bits. This is typically due to corruption during transmission.
10	Frames received with invalid station address	In an X.25 LAPB station, non-zero values indicate that the remote station is sending an HDLC address field value other than the one configured. This should not occur in a secondary station since frames with improper address field values are discarded by the adapter.

ID	Tally	Description
11	Number of times local station went busy	This tally counts the times this station entered a busy condition and began sending RNR within the protocol. Non-zero values are normal. High values indicate that this station often cannot consume data as fast as the remote station generates it.
12	Aborted frames received	This tally counts the number of incomplete frames received due to aborted framing. This is a symptom of line errors or of adapter clocking disruptions.
13	I-frames retransmitted	This tally counts I-frames which the remote rejected, causing them to be retransmitted. Non-zero values indicate problems in output to the remote.
14	Number of transmitter underrun conditions	This tally counts the times the adapter had to abort a transmission because it could not provide data to its USART rapidly enough. This is symptomatic of an overloaded adapter or, occasionally, of modem cable disconnections.
15	Number of times remote station went busy	This tally counts the times the other station entered a busy condition and began sending RNR within the protocol. Non-zero values are normal. High values indicate that the remote often cannot consume data as fast as the local station generates it.
16	Times remote exceeded remote busy timeout	This tally counts the times the other station remained in a busy condition longer than the remote busy timer interval. It indicates that the remote station often slows down throughput since it cannot consume data as fast as this station generates it.
17	Timeouts following successful connection	This tally counts the times the remote secondary or LAPB station failed to answer a poll after it was successfully contacted. A non-zero value indicates line problems, such as noise.

ID	Tally	Description
20	Remote initiated station reset	This tally is relevant in secondary and LAPB stations. Non-zero values are abnormal, indicating how many times the other station had to reset frame reject condition in this station. This implies problems in the HDLC software logic.
21	Remote initiated disconnections	This tally indicates the number of times the station unexpectedly went from the station connected state to the station disconnected state because of an HDLC DISC command from the other station. Non-zero values are normal and should equal the number of times the line was taken down during the day.
22	Locally initiated disconnections	This tally indicates the number of times the station unexpectedly went from the station connected state to the station disconnected state because of a command from a network application. Non-zero values are normal and should equal the number of times the line was taken down during the day.
23	Remote occurrences of FRMR condition	This tally designates the number of times the remote station went into frame reject condition due to a command or response which it received from this station. It indicates an incompatibility between HDLC stations. This tally usually equals the number of locally initiated station resets above.
24	Local occurrences of FRMR condition	This tally designates the number of times this station went into frame reject condition due to a command or response which it received from the other station. It indicates an incompatibility between HDLC stations. It usually equals the number of remotely initiated station resets above.

ID	Tally	Description
25	Frames discarded with bad P/F bit state	This tally indicates the number of frames this station discarded which violated the HDLC send/receive protocol: 1) as a TWA secondary, it received a frame from the primary while it was this station's turn to send, 2) as an X.25 LAPB combined, it received a command with the P bit set before it had answered a previous one, or 3) as an X.25 LAPB combined, it received a response with the F bit set before issuing one with the P bit set. This occurs if transmission is not being clocked or modem CTS is inhibited, but reception is still functional.
26	Modem DSR losses after good connection	This tally counts the times communications were interrupted due to loss of either modem DSR, modem CTS, or the remote dropped carrier if carrier loss indication was configured.
27	CTS losses	This tally counts the times communications were interrupted due to loss of either modem DSR, modem CTS, or the remote dropped carrier if carrier loss indication was configured.
28	Remote carrier drops	This tally indicates the number of times the remote carrier was lost on duplex line. Non-zero values indicate line drops. This tally is enabled only if the station is configured to honor carrier loss.
29	Line connection failures	This tally indicates the number of times the adapter initially asserted modem signal DTR, but modem signal DSR did not go active after waiting the interval of the connect line timer.
30	MPCA failures	This tally counts the number of unexpected adapter resets that have been encountered.

Using the Menus

You can not use the **odm** menus to display tallies (network statistics).

You may use the **odm** menus to display help information about tallies by selecting the following functions during the procedure described in the "Using the Maintenance Menus" section of this chapter:

```
<Other DLC Utility Help> from the Test type CHOICES menu
<Link Tallies>
```

Starting Data Capture

The **start data capture** operation captures and stores data that is transmitted and/or received on a selected device. This operation stores the raw captured data in a file that can be used later to help determine the accuracy level of the communication path to and/or from the device.

Using the Command Line

Enter the following command to start data capture:

```
# /opt/net/mpd/rundcap device > /tmp/datacap.raw &
```

This command causes the driver to begin writing data capture information to a raw data capture file. The raw data capture file specified above is */tmp/datacap.raw*, but you may specify any name you like.

The *device* argument specifies the MPCA device to query. Valid devices are */dev/mpd00* through */dev/mpd03*.

You may capture only data received by the driver, only data transmitted by the driver, or both. If you want to capture only received data, use the **-r** option. If you want to capture only transmitted data, use the **-t** option. If you do not specify either of these options, **rundcap** captures both received and transmitted data.

You may also specify the maximum message size, in bytes, to capture. To do this, use the **-m** *max_message_size* option, which causes only the first *max_message_size* bytes to be captured**.**

Note: The output from this command is NOT in readable form. See the "Viewing Data Capture Files" section of this chapter for more information.

If you are running data capture on a line with a great deal of communications traffic, do not let your raw data capture file grow too large. If the file gets too large, this may cause performance or storage problems when you view the file with **hdumper**(1M).

Using the Menus

You can not use the **odm** menus to start data capture operations.

You may use the **odm** menus to display help information about data capture operations by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other DLC Utility Help> from the Test type CHOICES menu
<Data Capture>
```

Stopping Data Capture

The stop data capture operation terminates active data capture processes.

Using the Command Line

Use the following procedure to stop the **rundcap** process from writing data capture information to a file:

1. To determine the process ID (PID) for the **rundcap** process, enter the following command**:**

   ```
   # ps -eaf | grep rundcap
   ```

2. To stop the **rundcap** process from running, enter the following command:

   ```
   # kill -9 PID
   ```

   The *PID* argument is the process ID retrieved in step number 1.

Using the Menus

You can not use the **odm** menus to stop data capture operations.

You may use the **odm** menus to display help information about data capture operations by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other DLC Utility Help> from the Test type CHOICES menu
<Data Capture>
```

Viewing Data Capture Files

The view data capture operation displays the contents of a raw data capture file.

Using the Command Line

You may use commands to generate the following types of data capture file displays:

- Display Entire File
- Display Errors Only
- Display Summary Information Only
- Display Specific Sequence Numbers Only
- Display Minimum Message Size
- Display Real-Time Data Capture

Display Entire File

Enter the following command to display the entire contents of a raw data capture file on your screen:

```
# /opt/net/mpd/hdumper -p1 raw_data_capture_filename
```

The **-p1** argument specifies that you want to view HDLC protocol data.

The *raw_data_capture_filename* argument specifies the name of the raw data capture file designated in the **rundcap** command**.**

Display Errors Only

To display only blocks or frames which are in error, enter the following command:

```
# /opt/net/mpd/hdumper -p1 -e raw_data_capture_filename
```

The **-p1** argument specifies that you want to view HDLC protocol data.

The *raw_data_capture_filename* argument specifies the name of the raw data capture file designated in the **rundcap** command**.**

Display Summary Information Only

To display only a summary of the number of messages captured and errors detected, enter the following command:

```
# /opt/net/mpd/hdumper -p1 -s raw_data_capture_filename
```

The **-p1** argument specifies that you want to view HDLC protocol data.

The *raw_data_capture_filename* argument specifies the name of the raw data capture file designated in the **rundcap** command**.**

Display Specific Sequence Numbers Only

To display only data capture frames within a certain range of sequence numbers, enter the following command:

```
# /opt/net/mpd/hdumper -p1 -f first -l last \
raw_data_capture_filename
```

The **-p1** argument specifies that you want to view HDLC protocol data.

The *first* option specifies the sequence number of the first frame you wish to display; the *last* option specifies the sequence number of the last frame you want to display. Only the frames with sequence numbers in this range are displayed.

The *raw_data_capture_filename* argument specifies the name of the raw data capture file designated in the **rundcap** command**.**

Display Minimum Message Size

You may also specify the minimum message size, in bytes, to be displayed. Only messages larger than the value in this field appear on your screen. This feature allows you to suppress lengthy exchanges of protocol polling sequences, since these messages are usually very short.

To specify a minimum message size, include the *min_message_size* option with any of the preceding commands.

Display Real-Time Data Capture

If you wish to start data capture and view the output in real time, enter the following command:

```
# rundcap device | hdumper -p1 &
```

The *device* argument specifies the MPCA device to query. Valid devices are */dev/mpd00* through */dev/mpd03*.

See the “Interpreting Data Capture” section of this chapter for information on how to interpret the data capture information displayed.

Using the Menus

You can not use the **odm** menus to view data capture files.

You may use the **odm** menus to display help information about data capture operations by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter.

```
<Other DLC Utility Help> from the Test type CHOICES menu
<Data Capture>
```

Interpreting Data Capture

The output displayed below is an example of a converted data capture file. In this example, the station is configured for LAPB operation.

The first line of the data capture file includes the name of the device the data was captured from and the date and time the data capture started. The file then contains the description and contents of each frame sent or received from the time data capture was started until it was stopped. The data capture utility displays the contents of each frame in hexadecimal characters.

```
DEVICE = mpd00    DATA CAPTURE STARTED: Thu
Jan 31 15:30:01 1991

Seq=0001  Type=GOOD ASCII OUTPUT  Drvr
Stat=0000  Tue Feb 20 15:31:23 1990

Address   01
Command   SABM with P/F

0000:  01   3F

Seq=0002  Type=GOOD ASCII INPUT  Drvr
Stat=0000  Tue Feb 20 15:31:23 1990

Address   01
Command: UA with F

0000:  01   73

Seq=0003  Type=GOOD ASCII OUTPUT  Drvr
Stat=0000  Tue Feb 20 15:31:39 1990

Address   01
Command   IFRAME
Nr        0
Ns        0

0000:  01   00  54  68  69  73  20  69  73
20  61  20  74  65  73  74
0010:  2E

Seq=0004  Type=GOOD ASCII INPUT  Drvr
Stat=0000  Tue Feb 20 15:31:39 1990

Address   01
Command   RR
Nr        1
```

```
0000:  01   21

Seq=0005  Type=GOOD ASCII OUTPUT  Drvr
Stat=0000  Tue Feb 20 15:31:46 1990

Address   01
Command   DISC with P/F

0000:  01   53

Seq=0006  Type=GOOD ASCII INPUT  Drvr
Stat=8000  Tue Feb 20 15:31:50 1990
Driver Status means:  link level timeout

Seq=0007  Type=GOOD ASCII OUTPUT  Drvr
Stat=0000  Tue Feb 20 15:31:50 1990

Address   01
Command   DISC with P/F
0000:  01   53
```

Frame Description

For each frame received or transmitted, the data capture facility displays a description of the frame. This description includes the following information:

- Sequence Number
- Type
- Driver Status
- Date and Time

Sequence Number

The sequence number indicates the sequence of frames in the data capture file. Sequence numbers begin with 0000 and continue sequentially until you stop data capture.

Type

The type field describes the type of data in the frame. Error-free frames have a type of “good.” Frames containing errors have types of “bad” or “error.” Received frames are designated as “input,” while sent frames have a type of “output.”

Driver Status

When the HDLC driver encounters an error or exception condition, you can interpret the driver status code to identify the error. A driver status code of 0000 indicates that the driver has not encountered any errors or exception conditions. The “Driver Status” section, later in this chapter, explains the values for this field.

Date and Time

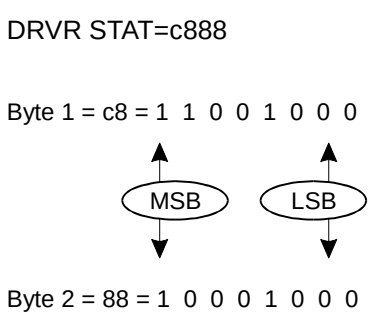
This field contains the date and time that the frame was either transmitted or received.

Driver Status Codes

If the HDLC driver encounters an error or exception condition, the driver status code field will contain a non-zero value. The next line of the data capture file contains a brief description of the status code’s meaning. Sometimes the description will not give you enough information to diagnose a problem and you will need to decipher the status code.

To interpret the driver status code, you must first convert the code from the hexadecimal value shown in the data capture file to binary format. For example, if the driver status code is c888, it is converted to binary format as shown in the following figure.

Figure 13-2. Driver Status Code Converted to Binary Format

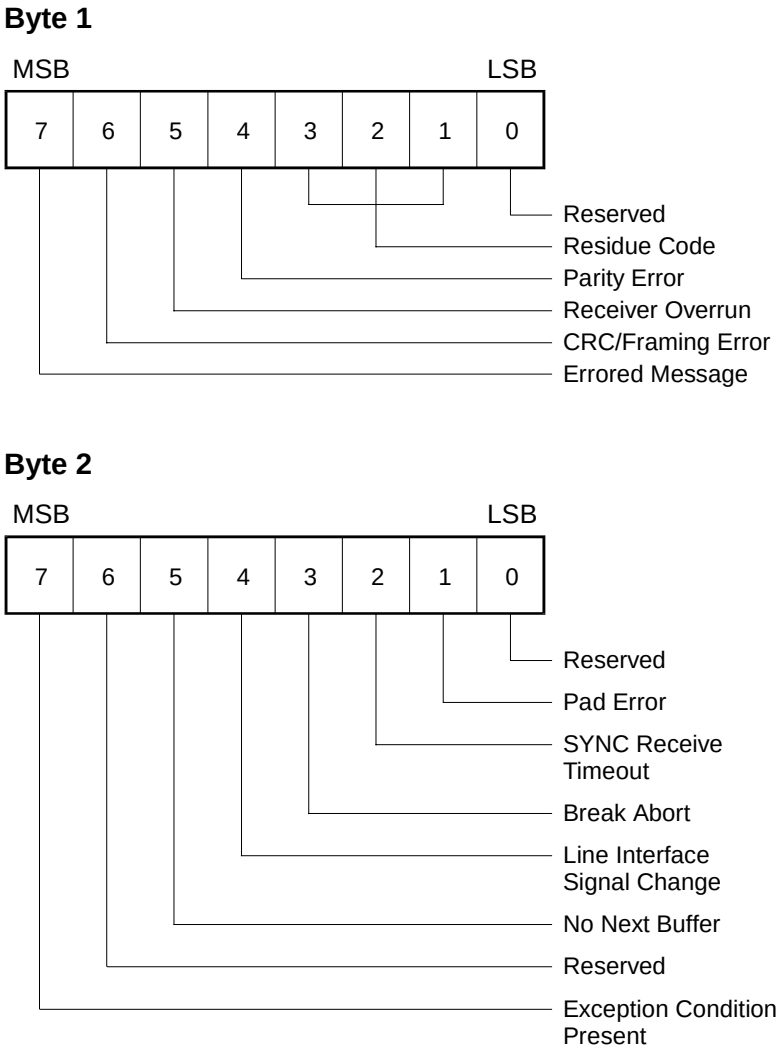


The interpretation of the two-byte status code depends on whether the HDLC driver received the frame or transmitted it.

Received Frames

For frames received by the HDLC driver, interpret the driver status code as shown in the following figure.

Figure 13-3. Received Frame Driver Status Format



The next table describes the bit assignments within Byte 1 of the status field.

Table 13-5. Received Frame Driver Status Field Byte 1 Bits

Bits	Function
0	Reserved for future use.
1-3	Residue code. Defined only for SDLC communication lines, this code defines the boundary between the last data character and the first CRC character in the receive buffer. In octet oriented SDLC communications, the residue code appears in the status field as **011.** Any other code appearing in this field is considered an error. The residue code does not affect the state of the receiver.
4	Parity error. Indicates that the received parity bit does not match the parity bit calculated from the received data bits. This bit is defined only in cases where parity checking is enabled.
5	Receiver overrun. Means that incoming data has overwritten data previously received.
6	CRC/Framing error. Indicates the received CRC check character calculated from the received data bits. This code does not affect the state of the receiver.
7	Error message. Indicates that an error occurred during reception and is being reported in this status byte; this bit is usually set in conjunction with one or more of bits 4, 5, and 6.

The following table describes the bit assignments within Byte 2 of the status field.

Table 13-6. Received Frame Driver Status Field Byte 2 Bits

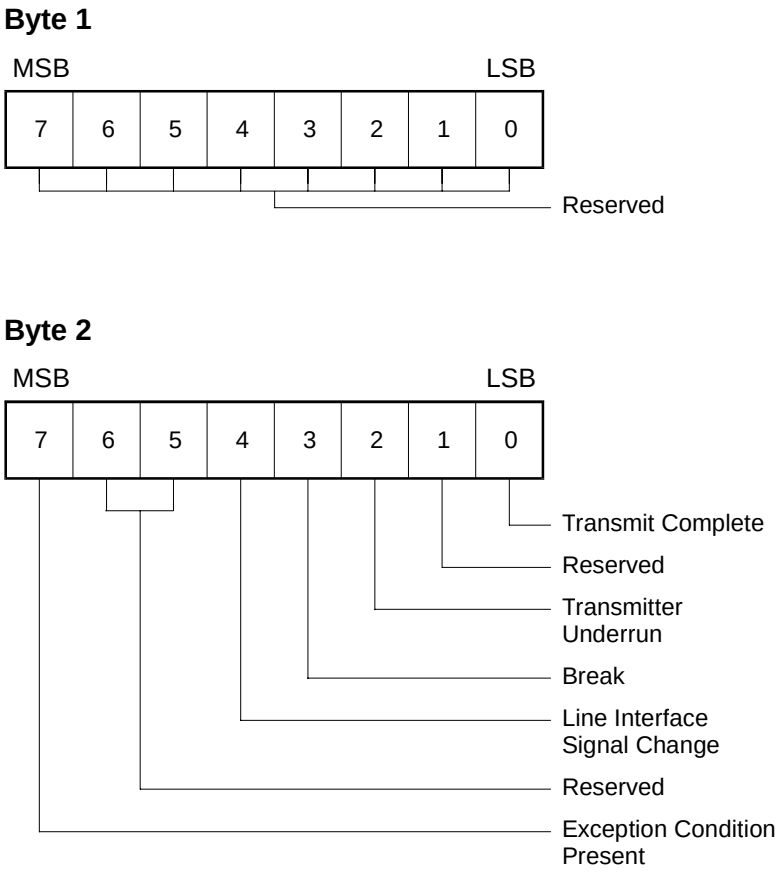
Bits	Function
0	Reserved for future use.
1	Pad error. Indicates that the requested pad test failed. This error does not affect the state of the receiver.
2	SYNC receive timeout. Not used for HDLC communications protocols.
3	Break/Abort. The meaning of this bit depends on the transmission mode of the communication line.

Bits	Function
4	Line interface signal change. Indicates that one or more of the line interface signals changed unexpectedly while data was being received into this buffer; therefore, data present in the receive buffer may not be valid. The following line interface signal changes are reported with this bit. Note that changes in more than one line signal may be reported simultaneously. • Loss of Data Set Ready (DSR) • Loss of Carrier On (CO) • Change in Test Indicator (TI) The receiver is disabled if this error occurs.
5	No next buffer. Indicates that the pool of receive buffers allocated to the communication line specified by this receiver interrupt has been exhausted and reception on this line has halted. The receive buffer descriptor in which this bit is set points to the last receive buffer that was available to and used by the adapter firmware. The receiver is disabled if this error occurs.
6	Reserved for future use.
7	Exception condition present. Indicates that an exception condition occurred during reception and is being reported in this status byte; this bit is usually set in conjunction with one or more of bits 3, 4, and 5.

Transmitted Frames

For frames transmitted by the HDLC driver, interpret the driver status code as shown in the following figure.

Figure 13-4. Transmitted Frame Driver Status Format



Byte 1 of the driver status field is entirely reserved for future use. The next table describes the bit assignments within Byte 2 of the status field.

Table 13-7. Transmitted Frame Driver Status Field Byte 2 Bits

Bits	Function
0	Transmit complete. Indicates that a transmit operation has completed. The state of this bit does not indicate the success or failure of the transmit operation; bits 2, 3, 4, and 7 indicate failure.
1	Reserved for future use.
2	Transmitter underrun. Indicates that the adapter was unable to obtain data characters quickly enough to maintain the synchronous transmission. This bit is defined only for SDLC communications.
3	Break. Not used for HDLC communications.
4	• Line interface signal change. Indicates that one or more of the line interface signals changed unexpectedly while data was being transmitted from this buffer; therefore, data present in the receive buffer may not be valid. The following line interface signal changes are reported with this bit. Note that changes in more than one line signal may be reported simultaneously. Loss of Data Set Ready (DSR) • Loss of Carrier On (CO) in a configuration where the signal is not allowed to drop • Loss of Clear to Send (CTS) • Change in Test Indicator (TI)
5-6	Reserved for future use.
7	Exception condition present. Indicates that an exception condition occurred during transmission and is being reported in this status byte; this bit is usually set in conjunction with one or more of bits 2, 3, and 4. If any of these errors occur, the transmitter is disabled. The system software is responsible for placing the line interface signal (RTS) in the state (active or inactive) where it is to be left after all transmissions are complete.

Frame Contents

After the frame description, the next lines in the data capture file contain the address of the sending machine, the command sent, and, if applicable, the send and receive sequence numbers. The **Ns** value is the sequence number for send frames; the **Nr** value is the sequence number for received frames. The information in these lines comes from the address and control fields in the first line of the data capture frame.

The following is an example of data capture frame contents. The contents consist of:

- A line number
- An address field
- A control field
- An optional data field

Only the first line of the frame contains the address and control fields; all subsequent lines contain only the line number and data fields.

```
0000:  01 00 54 68 69 73 20 69 73 20 61 20 74 65 73 74
0010:  2E
```

The address field identifies the intended receiver of a command frame and the transmitter of a response frame.

The control field identifies the command sent or received by the driver. Its value depends on whether the driver is configured for SDLC or LAPB operation. The following sections explain the possible values for each type of operation. To interpret the control field, you must first convert its value from hexadecimal to binary format.

LAPB Control Field Values

The next table shows the commands and responses used in LAPB operation, and their binary values. The value **Ns** is the sequence number for sent frames; the value **Nr** is the sequence number for received frames.

Note: The values in fields **7** through **0** in the following table are **binary values**.

Table 13-8. LAPB Control Field Values

Format	Command	Response	7	6	6	5	4	3	2	1	0
Information Transfer	I	I	Nr				P	Ns			0
Supervisory	RR	RR	Nr				P/F	0	0	0	1
	RNR	RNR	Nr				P/F	0	1	0	1
	REJ	REJ	Nr				P/F	1	0	0	1
Unnumbered	SABM		0	0	1		P	1	1	1	1
	DISC		0	1	0		P	0	0	1	1
		DM	0	0	0		F	1	1	1	1
		UA	0	1	1		F	0	0	1	1
		FRMR	1	0	0		F	0	1	1	1

LAPB Commands and Responses

The following table describes the commands and responses.

Table 13-9. LAPB Command/Response Descriptions

Command	Description
I	Information. Transfers a frame containing an information field across a data link.
RR	Receive ready. Indicates sender is ready to receive an I-frame or acknowledges previously received I-frames through Nr-1. May also be used to indicate that a previously reported busy condition is clear.
RNR	Receive not ready. Temporarily unable to accept additional incoming I-frames; acknowledges I-frames through Nr-1.
REJ	Reject. Requests retransmission of I-frames starting with Nr. I-frames Nr-1 and below are acknowledged; additional I-frames pending initial transmission may be transmitted following the retransmitted I-frames. REJ is cleared when I-frame with Ns equal to Nr is received.
SABM	Set asynchronous balanced mode. Places the addressed receiver in an asynchronous balanced mode information transfer phase where all

	command/response control fields are one octet in length. No information field is permitted with this command.
DISC	Disconnect. Informs receiver that sender is suspending operation. No information field is permitted with this command. Sender waits for UA from receiver before actually disconnecting. Previously unacknowledged I-frames remain unacknowledged.
UA	Unnumbered acknowledgment. Acknowledges receipt and acceptance of SABM and DISC commands. No information field is permitted with this response.
DM	Disconnected mode. Reports that receiver is logically disconnected from the link without having received a DISC. No information field is permitted with this response.
FRMR	Frame reject. Reports an error condition not recoverable by retransmission of the identical frame. One of the following conditions occurred: Undefined or unimplemented command or response Information field exceeds maximum length Invalid Nr Supervisory or unnumbered frame with information field or incorrect length

SDLC Control Field Values

The next table shows the commands and responses used in SDLC operation and their binary values. The value **Ns** is the sequence number for sent frames; the value **Nr** is the sequence number for received frames**.**

Note: The values in fields **7** through **0** in the “SDLC Control Field Values” table are **binary values**.

Table 13-10. SDLC Control Field Values

Format	Command	Response	7	6	5	4	3	2	1	0
Information Transfer	I	I	Nr			P	Ns			0
Supervisory	RR	RR	Nr			P/F	0	0	0	1
	RNR	RNR	Nr			P/F	0	1	0	1
	REJ	REJ	Nr			P/F	1	0	0	1
	NSI	NSI	0	0	0	P/F	0	0	1	1
		ROI	0	0	0	F	0	1	1	1
	SIM		0	0	0	P	0	1	1	1
Nonsequenced	SNRM		1	0	0	P	0	0	1	1
		ROL	0	0	0	F	1	1	1	1
	DISC		0	1	0	P	0	0	1	1
		NSA	0	1	1	F	0	0	1	1
		CMDR	1	0	0	F	0	1	1	1
	XID	XID	1	0	1	P/F	1	1	1	1
	NSP		0	0	1	P/F	0	0	1	1
	TEST	TEST	1	1	1	P/F	0	0	1	1

SDLC Commands and Responses

The following table describes the SDLC commands and responses.

Table 13-11. SDLC Command/Response Descriptions

Command	Description
I	Information. Transfers a frame containing an information field across a data link.
RR	Receive ready. Indicates sender is ready to receive an I-frame or acknowledges previously received I-frames through Nr-1. May also be used to indicate that a previously reported busy condition is clear.
RNR	Receive not ready. Temporarily unable to accept additional incoming I-frames; acknowledges I-frames through Nr-1.
REJ	Reject. Requests retransmission of I-frames starting with Nr. I-

	frames Nr-1 and below are acknowledged; additional I-frames pending initial transmission may be transmitted following the retransmitted I-frames. REJ is cleared when I-frame with Ns equal to Nr is received.
NSI	Command or response that requires nonsequenced information. The HDLC driver does not support this protocol element.
ROI	Initialization needed; not permitted in response to SIM. No information field is permitted with this command. The HDLC driver does not support this protocol element.
SIM	Set initialization mode; the using system prescribes the procedures. No information field is permitted with this command. The HDLC driver does not support this protocol element.
SNRM	Set normal response mode. Places the addressed receiver in an normal response mode information transfer phase. No information field is permitted with this command.
ROL	The addressed receiver station is offline. No information field is permitted with this command. The HDLC driver does not support this protocol element.
DISC	Disconnect. Informs receiver that sender is suspending operation. No information field is permitted with this command. Sender waits for NSA from receiver before actually disconnecting. Previously unacknowledged I-frames remain unacknowledged.
NSA	Nonsequence acknowledge. Acknowledges receipt and acceptance of nonsequenced commands. No information field is permitted with this command. The HDLC driver does not support this protocol element.
CMDR	Invalid command received by receiver; must receive SNRM, DISC, or SIM.
XID	Information field contains system exchange identification.
NSP	If the Pbit is set to zero, response is optional for the receiver. No information field is permitted with this command. The HDLC driver does not support this protocol element.
TEST	Check pattern in information field.

Driver Interface Trace

The driver maintains a history of the DLPI commands received and of the DLPI states that it has encountered. This information is kept for each adapter link.

The DLPI history is accessible by UNIX **ioctl**() commands. The HDRDHISTORY command retrieves the history information for the current link. The HDRDCHISTORY command retrieves the history information and then resets the structure.

The data is returned in a data structure of the form *struct DLPI_HISTORY* and has the form shown in the following table.

Table 13-12. Driver Interface Trace History

Field	Contents
ulong primitives[MAXPRIMHISTORY]	Last MAXPRIMHISTORY primitives received by the driver.
unsigned char states[MAXSTATEHISTORY]	Last MAXSTATEHISTORY states encountered by the driver.
int nprim	Index of the oldest primitive in the history. The entries wrap around.
int nstate	Index of the oldest state recorded in the history. The entries wrap around.

Retrieve Attached Count

The driver maintains a count of the number of stations currently attached to each link. This information is available through the HDRDATTACHED **ioctl** command**.**

The number of stations attached to the current link is returned in an integer (type **int**).

Chapter 14

Testing LANs

Accessing LAN Diagnostics

In-Service (Level 2) diagnostics permit you to perform basic Local Area Network (LAN) tests and diagnose LAN activities. These in-service diagnostic test programs interact with the Integrated LAN Driver (ILD) software through various **ioctl** commands and driver utilities.

The ILD includes diagnostic device tests, data capture operations and STREAMS error logging capabilities that you may use to gather network or driver statistics or to exercise the system's communications hardware.

Command Line Tests

You can perform all of the ILD LAN diagnostic functions described in this chapter by using command line entries.

For additional information about performing ILD LAN diagnostics from the command line, refer to the *Installation and Diagnostics* book(s) for the hardware controllers (such as the Ethernet and/or Token Ring adapters) that provide the ILD interface with the operating system software.

odm Menu Tests

You can perform the following tests through the **odm** Online Diagnostics and Maintenance menus**:**

- Loop 1 test
- Loop 2 test
- Loop 3 test

You can also use the odm menus to display driver configuration, hardware configuration and data capture operation help information.

Using the Maintenance Menus

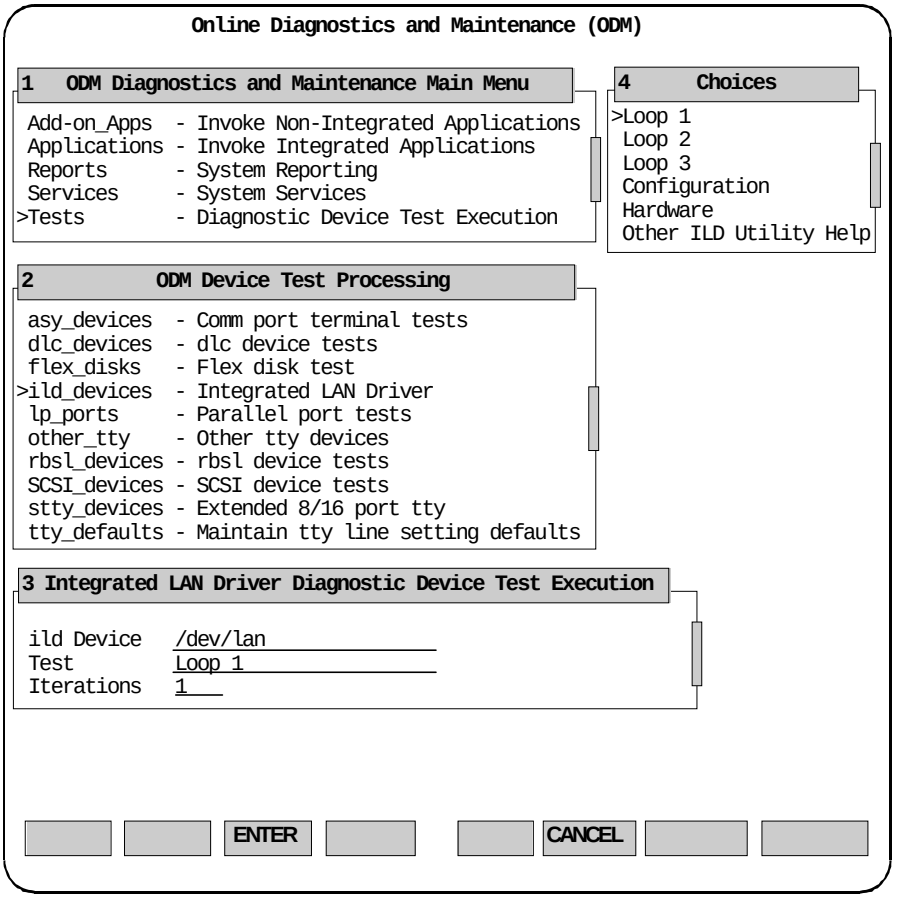
You can perform some of the most commonly used ILD diagnostics and troubleshooting tasks from the **odm** Online Diagnostics and Maintenance menus**.**

The following sequence of menu selections permits you to select ILD LAN tests through the **odm** menus. An example of the display created by these menu selections is depicted in the next figure. These menu selection steps are the same for all ILD LAN tests that may be selected through the menus.

1. Select **<Tests>** from the initial menu**.**
2. Select <ild_devices> from the ODM Device Test Processing menu.
3. If you know the name of the LAN, type that name and press ENTER. If you do not know the device name, select CHOICES for a list of available device names, select your choice, and press ENTER to proceed to the *Test identification* field. LANs (also known as ILD devices) have names in the form *lanx*, where *x* is 0 through 15.
4. If you know the name of the test to perform, enter the desired test name, and press ENTER. If you want to see the tests available for **ild** devices from the **odm** menus, press CHOICES, select the desired test, and press ENTER.

   Only one test can be selected from the CHOICES menu. The remaining sections of this chapter provide information about the ILD LAN tests.
5. If you want to perform the test more than one time, use the arrow keys to move the cursor to the *Iterations* field and enter the number of times you want to perform the test.
6. Press the SAVE function key to activate the selected test for the selected number of iterations.

Figure 14-1. ILD Test Menu Selections



Performing the Loop 1 Test

The **loop1** test executes level 0 diagnostics. It includes memory and register tests as well as hardware tests on the adapter. The network must be down to perform this test.

Using the Command Line

Use the following procedure to run the **loop1** test from the command line prompt:

1. Shut down the network on the PPA you are testing by entering the following command:

   ```
   # /etc/ildinit -u ppa
   ```

   Note: Before shutting down the network, make sure that you shut down any other packages (such as WIN—TCP or NetBIOS) that use the network.

   The *ppa* argument specifies the channel on which to perform the test. The *ppa* is the relative physical position of the adapter channel, beginning with zero, from left to right and top to bottom.

2. Run the **loop1** test by entering the following command:

   ```
   # /etc/ild/loop1 ppa
   ```

   This test only displays a message if it fails. Possible error messages for Ethernet adapters are:

   ```
   WDENET: network address ROM checksum error
   WDENET: shared memory test error
   ```

   Possible error messages for Token-Ring adapters are:

   ```
   NCRTR: adapter memory test error
   ```

Using the Menus

You may use the **odm** menus to run the **loop1** test by selecting the **loop1** test during step 4 of the procedure described in the “Using the Maintenance Menus” section of this chapter.

Performing the Loop 2 Test

The **loop2** test sends a broadcast message out on the network and collects responses. This test finds out who is listening and can receive frames sent out from your node.

Using the Command Line

Use the following procedure to run the **loop2** test from the command line prompt:

1. To run the **loop2** test in verbose mode, enter the following command:

   ```
   # /etc/ild/loop2 -v ppa
   ```

   The *ppa* argument specifies the channel on which to perform the test. The *ppa* is the relative physical position of the adapter channel, beginning with zero, from left to right and top to bottom. The **-v** option specifies that output should be in verbose mode.

2. The **loop2** test displays information as depicted below, according to the result of the test.

   If other nodes respond (in verbose mode), the display looks like the following example.

   ```
   -Attaching
      -Binding
          -Sending XID
          -Responders:
             1 0080035b24c1
             2 008003c52ac1
             3 ffffffffffff
             4 ffffffffffff
             5 0000435b441a
             6 0003430a2a90
          -Unbinding
          -Detaching
          6 nodes responding
   ```

   If no other nodes respond, the **loop2** test displays information similar to the following example (verbose mode).

   ```
   -Attaching
   ```

```
-Binding
   -Sending XID
   -Responders:
   -Unbinding
   -Detaching
   No nodes responding
   0 nodes responding
```

If any errors occur, the **loop2** test displays an appropriate error message. Possible error messages are:

```
Unexpected DLPI primitive x,expected y
Invalid primitive acknowledged x,expected y
Error_Ack, primitive x, errno y, unitx_errno z
```

Using the Menus

You may use the **odm** menus to run the **loop2** test by selecting the **loop2** test during step 4 of the procedure described in the “Using the Maintenance Menus” section of this chapter.

Performing the Loop 3 Test

The **loop3** test sends UI frames across the network. You should use this test along with data capture by starting data capture, running this test, and then stopping data capture.

Using the Command Line

Use the following procedure to run the **loop2** test from the command line prompt:

1. To run the **loop3** test, enter the following command:

   ```
   # /etc/ild/loop3 ppa sap|type #frames
   ```

   The *ppa* argument specifies the channel on which to perform the test. The *ppa* is the relative physical position of the adapter channel, beginning with zero, from left to right and top to bottom.

   For token-ring, specify a *sap* value from 1 through 255; for ethernet, specify a *type* value from 1519 through 65535. You should specify a number that you can easily recognize when you view the data capture output.

2. This test displays a message if it fails. Possible error messages are:

   ```
   Unexpected DLPI primitive x, expected y
   Invalid primitive acknowledged x, expected y
   Error_Ack, primitive x, errno y, unitx_errno z
   ```

Using the Menus

You may use the **odm** menus to run the **loop3** test by selecting the **loop3** test during step 4 of the procedure described in the “Using the Maintenance Menus” section of this chapter.

Starting Data Capture

The start data capture operation captures and stores data that is transmitted and/or received on a selected device. This operation stores the raw captured data in a file that can be used later to help determine the accuracy level of the communication path to and/or from the device.

Using the Command Line

Use the following command to start data capture on an adapter:

```
# /etc/ild/dcap -p -s size -P ppa -o outfile &
```

Note: You must run data capture in background mode.

The **dcap** process runs continually, reading frames into memory, until you stop it. When you stop **dcap**, it writes the frames into the specified *outfile*. The following table describes the arguments of the **dcap** command.

Table 14-1. dcap Command Arguments

ARGUMENT	DESCRIPTION
-p	Promiscuous mode. Causes **dcap** to capture all frames being transmitted on the network. If promiscuous mode is not used, **dcap** only captures those frames whose destination address is the specified adapter or broadcast frames. Generally, you should use promiscuous mode when diagnosing network problems.
-s *size*	Maximum number of frames to be written to the data capture output file. The output file is a circular file; once it reaches the maximum number of frames, it deletes the oldest frame each time it adds a new one, so that it always contains the most recent frames. *Size* may be any integer from 1 through 10000. The default is 1000.

ARGUMENT	DESCRIPTION
-P *ppa*	The ppa on which to perform data capture. The *ppa* is the relative physical position of the adapter channel, beginning with zero, from left to right and top to bottom.
-o *outfile*	The name of the file in which to write data capture frames.

Using the Menus

You can not use the **odm** menus to start data capture operations.

You may use the **odm** menus to display help information about data capture operations by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other ILD Utility Help> from the Test type CHOICES menu
<Data Capture>
```

Stopping Data Capture

The **stop data capture** operation terminates active data capture processes.

Using the Command Line

Use the following procedure to stop the data capture process and write the frames to a file:

1. Enter the following command to find the process ID of the **dcap** process you wish to stop**:**

   ```
   # ps -ef
   ```

2. Enter the following command to stop the process:

   ```
   # kill PID
   ```

   where *PID* is the process *ID* from step 1. When you kill the **dcap** process, it writes the data capture information to the *outfile* you specified when you initiated data capture.

Using the Menus

You can not use the **odm** menus to view data capture files.

You may use the **odm** menus to display help information about data capture operations by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other ILD Utility Help> from the Test type CHOICES menu
<Data Capture>
```

Viewing Data Capture Files

Because the data capture output file is a circular file, you cannot simply view it with **vi** or **pg.** You must first parse the file to make the data capture information meaningful.

Using the Command Line

Enter the following command to parse the data capture output file:

```
#/etc/ild/parser -hs -iinfile -ooutfile
```

The following table describes the **parser** command arguments.

Table 14-2. parser Command Arguments

ARGUMENT	DESCRIPTION
-h	Prints a hex dump of each frame.
-s	Prints a summary line for each frame.
-i *infile*	The name of the file to parse (the *outfile* specified in the **dcap** command).
-o *outfile*	The name of the file in which to write parsed data capture frames. By default, the parsed file is written to your screen.

You may also specify any of the options shown in the following table.

Table 14-3. parser Command Options

OPTION	DESCRIPTION
-k	Prints output in **keyword=value** format. Currently not supported**.**
-t *tracefile*	Specifies the name of the file to write error messages to. By default, error messages are written to your screen.
-v *verbose_level*	Specifies the level of verbose messages to print. Not currently supported.

The following sections provide examples of the different types of display(s) that are created by viewing data capture file operations.

Default Output Example

If you do not specify the **-h** or **-s** arguments, the output looks like the following example. The example shows the ASCII equivalent beside the hex data.

```
Filename: outfile MAC Type:   CSMA/CD Start Time: 15:48:42
Records: 1000

     === Frame: 0 === Length:60 === Sequence:1
    === Time: 15:48:42 =======
    DLC: --- DLC Header ---
    DLC:
    DLC: Destination: 080014110450
    DLC: Source:      08000e02c24b
    DLC:
    ETH: --- Ethernet Header ---
    ETH:
    ETH: Type:  0800 IP
    ETH:
    IP:  --- IP Header ---
    IP:
    IP:  Hleng:   20 bytes  Version:  4 TOS: 0
    IP:  Tleng:   41 bytes  Proto: TCP Id:   29104
    IP:  Offset:         0  TTLive:   30  Chksum:   6b81
    IP:  Src: 90.1.5.178
    IP:  Dst: 90.1.5.234
    IP:
    TCP: --- TCP Header ---
    TCP:
    TCP: Src Port:      13330  Dst Port:     23
    TCP: Seq No.:   120262838  Ack No.:      113512463
    TCP: Data Offset:  20 bytes   Window:       2048
    TCP: Flags: ACK
    TCP: Checksum:       7b58  Urp:      0
    ETH:
    ETH: --- Data ---
    0000: 0a 25 00 00 03 00
    |.%....   |

    === Frame:     1 === Length:   60 === Sequence:     2
    === Time: 15:48:42 =======
    DLC: --- DLC Header ---
    DLC:
    DLC: Destination: 08000e02c24b
    DLC: Source:      080014110450
    DLC:
```

```
ETH: --- Ethernet Header ---
ETH:
ETH: Type:  0800 IP
ETH:
IP:  --- IP Header ---
IP:

IP:  Hleng:   20 bytes  Version:  4  TOS:   0
IP:  Tleng:   40 bytes  Proto: TCP Id: 58825
IP:  Offset:         0  TTLive:   15  Chksum:  669
IP:  Src: 90.1.5.234
IP:  Dst: 90.1.5.178
IP:
TCP: --- TCP Header ---
TCP:
TCP: Src Port:          23  Dst Port:     13330
TCP: Seq No.:   113512464  Ack No.:      120262839
TCP: Data Offset:  20 bytes   Window:       512
TCP: Flags: ACK
TCP: Checksum:       8b57  Urp:      0
ETH:
ETH: --- Data ---
0000: 5a 01 06 31 5a 01
|Z..1Z.          |
=== Frame:  2 === Length:  60 === Sequence:  3 === Time:
15:48:42 =======
DLC: --- DLC Header ---
DLC:
DLC: Destination: 08000e02c1fc
DLC: Source:      08000e020013
DLC:
ETH: --- Ethernet Header ---
ETH:
ETH: Type:  0800 IP
ETH:
IP:  --- IP Header ---
IP:
IP:  Hleng:   20 bytes  Version:  4  TOS:   0
IP:  Tleng:   40 bytes  Proto: TCP Id: 27177
IP:  Offset:         0  TTLive:   30  Chksum:   748c
IP:  Src: 90.1.4.110
IP:  Dst: 90.1.5.171
IP:
TCP: --- TCP Header ---
TCP:
TCP: Src Port:       2074  Dst Port:     6000
TCP: Seq No.:   142166402  Ack No.:      71731218
TCP: Data Offset:  20 bytes   Window:       8192
TCP: Flags: ACK
TCP: Checksum:       d3db  Urp:      0
ETH:
ETH: --- Data ---
```

```
0000: 2f 39 30 20 1b 5b
|/90 .[          |
=== Frame:  3 === Length:  60 === Sequence:  4 === Time:
15:48:42 =======
DLC: --- DLC Header ---
DLC:
DLC: Destination: 08000e02cb50
DLC: Source:      080014109773
DLC:
ETH: --- Ethernet Header ---
ETH:
ETH: Type:  0600 XNS
ETH:
ETH: --- Data ---
0000: ff ff 00 22 00 00 00 00 00 05 08 00 0e
02 cb 50 |...".....P|
0010: 02 2e 00 00 00 05 08 00 14 10 97 73 0b
c4 64 60 |.....s..d`|
0020: 15 1f 00 69 00 00 08 3b 00 f2 e7 4c 00
12       |...i...;...L..  |
```

Hex Dump Output Example

If you use the **-h** (hex dump) command option, the output looks like the default output, with the addition of a hex dump of each frame, like the following example.

```
Filename: outfile MAC Type: CSMA/CD Start Time: 15:48:42
Records: 1000
=== Frame:  0 === Length:  60 === Sequence:  1 === Time:
15:48:42 =======
DLC: --- DLC Header ---
DLC:
DLC: Destination: 080014110450
DLC: Source:      08000e02c24b
DLC:
ETH: --- Ethernet Header ---
ETH:
ETH: Type:  0800 IP
ETH:
IP:  --- IP Header ---
IP:
IP:  Hleng:   20 bytes     Version: 4       TOS: 0
IP:  Tleng:   41 bytes     Proto: TCP       Id: 29104
IP:  Offset:          0    TTLive: 30       Chksum:6b81
IP:  Src: 90.1.5.178
IP:  Dst: 90.1.5.234
IP:
TCP: --- TCP Header ---
TCP:
TCP: Src Port:      13330     Dst Port: 23
```

```
TCP: Seq No.:   120262838     Ack No.: 113512463
TCP: Data Offset:  20 bytes       Window: 2048
TCP: Flags: ACK
TCP: Checksum:       7b58     Urp: 0
ETH:
ETH: --- Data ---
0000: 0a 25 00 00 03 00
|.%....          |
--- Frame Hex Dump ---
0000: 08 00 14 11 04 50 08 00 0e 02 c2 4b 08
00 45 00 |.....P.....K..E.|
0010: 00 29 71 b0 00 00 1e 06 6b 81 5a 01 05
b2 5a 01 |.)q.....k.Z...Z.|
0020: 05 ea 34 12 00 17 07 2b 10 b6 06 c4 10
0f 50 10 |..4....+......P.|
0030: 08 00 7b 58 00 00 0a 25 00 00 03 00
|..{X...%....    |

=== Frame:  1 === Length:  60 === Sequence:  2 === Time:
15:48:42 =======
DLC: --- DLC Header ---
DLC:
DLC: Destination: 08000e02c24b
DLC: Source:      080014110450
DLC:
ETH: --- Ethernet Header ---
ETH:
ETH: Type:  0800 IP
ETH:

IP:  --- IP Header ---
IP:
IP:  Hleng:   20 bytes  Version:  4  TOS: 0
IP:  Tleng:   40 bytes  Proto: TCP Id: 58825
IP:  Offset:         0  TTLive:   15  Chksum: 669
IP:  Src: 90.1.5.234
IP:  Dst: 90.1.5.178
IP:
TCP: --- TCP Header ---
TCP:
TCP: Src Port:         23  Dst Port: 13330
TCP: Seq No.:   113512464  Ack No.: 120262839
TCP: Data Offset:  20 bytes   Window: 512
TCP: Flags: ACK
TCP: Checksum:       8b57  Urp: 0
ETH: --- Data ---
0000: 5a 01 06 31 5a 01
|Z..1Z.                |
--- Frame Hex Dump ---
0000: 08 00 0e 02 c2 4b 08 00 14 11 04 50 08
00 45 00 |.....K.....P..E.|
0010: 00 28 e5 c9 00 00 0f 06 06 69 5a 01 05
```

```
ea 5a 01 |.(.......iZ...Z.|
0020: 05 b2 00 17 34 12 06 c4 10 10 07 2b 10
b7 50 10 |....4......+..P.|
0030: 02 00 8b 57 00 00 5a 01 06 31 5a 01
|...W..Z..1Z.    |

=== Frame:  2 === Length:  60 === Sequence:  3
=== Time: 15:48:42 =======
DLC: --- DLC Header ---
DLC:
DLC: Destination: 08000e02c1fc
DLC: Source:      08000e020013
DLC:
ETH: --- Ethernet Header ---
ETH:
ETH: Type:  0800 IP
ETH:

IP:  --- IP Header ---
IP:
IP:  Hleng:   20 bytes  Version:  4  TOS: 0
IP:  Tleng:   40 bytes  Proto: TCP Id: 27177
IP:  Offset:         0  TTLive:   30  Chksum: 748c
IP:  Src: 90.1.4.110
IP:  Dst: 90.1.5.171
IP:
TCP: --- TCP Header ---
TCP:
TCP: Src Port:       2074  Dst Port: 6000
TCP: Seq No.:   142166402  Ack No.: 71731218
TCP: Data Offset:  20 bytes   Window: 8192
TCP: Flags: ACK
ETH:
ETH: --- Data ---
0000: 2f 39 30 20 1b 5b
|/90 .[          |

--- Frame Hex Dump ---
0000: 08 00 0e 02 c1 fc 08 00 0e 02 00 13 08
00 45 00 |..............E.|
0010: 00 28 6a 29 00 00 1e 06 74 8c 5a 01 04
6e 5a 01 |.(j).....t.Z..nZ.|
0020: 05 ab 08 1a 17 70 08 79 49 82 04 46 88
12 50 10 |.....p.yI..F..P.|
0030: 20 00 d3 db 00 00 2f 39 30 20 1b 5b
| ...../90 .[    |

==== Frame:3 === Length:60 === Sequence:4 === Time:15:48:42
=====
DLC: --- DLC Header ---
DLC:
DLC: Destination: 08000e02cb50
```

```
DLC: Source:      080014109773
DLC:
ETH: --- Ethernet Header ---
ETH:
ETH: Type:  0600 XNS
ETH:
ETH: --- Data ---
0000: ff ff 00 22 00 00 00 00 00 05 08 00 0e
02 cb 50 |...".....P|
0010: 02 2e 00 00 00 05 08 00 14 10 97 73 0b
c4 64 60 |.....s..d`|
0020: 15 1f 00 69 00 00 08 3b 00 f2 e7 4c 00 12
|...i...;...L..  |
--- Frame Hex Dump ---
0000: 08 00 0e 02 cb 50 08 00 14 10 97 73 06
00 ff ff |.....P.....s....|
0010: 00 22 00 00 00 00 00 05 08 00 0e 02 cb
50 02 2e |."..........P..|
0020: 00 00 00 05 08 00 14 10 97 73 0b c4 64
60 15 1f |.........s..d`..|
0030: 00 69 00 00 08 3b 00 f2 e7 4c 00 12
|.i...;...L..    |
```

Summary Output Example

If you use the **-s** (summary) command option, the output contains only a summary line for each frame, as shown in the following example.

```
Filename: outfile MAC Type: CSMA/CD Start Time: 15:48:42
Records: 1000
      0     1 15:48:42  080014110450 <- 08000e02c24b ETH  IP
      1     2 15:48:42  08000e02c24b <- 080014110450 ETH  IP
      2     3 15:48:42  08000e02c1fc <- 08000e020013 ETH  IP
      3     4 15:48:42  08000e02cb50 <- 080014109773 ETH  XNS
```

For each frame, the summary line contains the frame number, the sequence number, the time the frame was captured, the destination address, the source address, the type of transmission, and the protocol used.

Summary and Hex Dump Output Example

If you use the **-s** (summary) and -h (hex dump) command options, the output includes a summary line and a hex dump for each frame, as shown in the following example.

```
Filename: outfile MAC Type: CSMA/CD Start Time: 15:48:42
Records: 1000
      0     1 15:48:42  080014110450 <- 08000e02c24b ETH  IP
   --- Frame Hex Dump ---
```

```
0000: 08 00 14 11 04 50 08 00 0e 02 c2 4b
08 00 45 00 |.....P.....K..E.|
0010: 00 29 71 b0 00 00 1e 06 6b 81 5a 01
05 b2 5a 01 |.)q.....k.Z...Z.|
0020: 05 ea 34 12 00 17 07 2b 10 b6 06 c4
10 0f 50 10 |..4....+......P.|
0030: 08 00 7b 58 00 00 0a 25 00 00 03 00
|..{X...%....    |

   1     2 15:48:42  08000e02c24b <- 080014110450 ETH  IP
--- Frame Hex Dump ---
0000: 08 00 0e 02 c2 4b 08 00 14 11 04 50
08 00 45 00 |.....K.....P..E.|
0010: 00 28 e5 c9 00 00 0f 06 06 69 5a 01
05 ea 5a 01 |.(.......iZ...Z.|
0020: 05 b2 00 17 34 12 06 c4 10 10 07 2b
10 b7 50 10 |....4......+..P.|
0030: 02 00 8b 57 00 00 5a 01 06 31 5a 01
|...W..Z..1Z.    |

   2     3 15:48:42  08000e02c1fc <- 08000e020013 ETH  IP
--- Frame Hex Dump ---
0000: 08 00 0e 02 c1 fc 08 00 0e 02 00 13
08 00 45 00 |..............E.|
0010: 00 28 6a 29 00 00 1e 06 74 8c 5a 01
04 6e 5a 01 |.(j)....t.Z..nZ.|
0020: 05 ab 08 1a 17 70 08 79 49 82 04 46
88 12 50 10 |.....p.yI..F..P.|
0030: 20 00 d3 db 00 00 2f 39 30 20 1b 5b
| ...../90 .[    |

   3     4 15:48:42  08000e02cb50 <- 080014109773 ETH  XNS
--- Frame Hex Dump ---
0000: 08 00 0e 02 cb 50 08 00 14 10 97 73
06 00 ff ff |.....P.....s....|
0010: 00 22 00 00 00 00 00 05 08 00 0e 02
cb 50 02 2e |.”...........P..|
0020: 00 00 00 05 08 00 14 10 97 73 0b c4
64 60 15 1f |.........s..d‘..|
0030: 00 69 00 00 08 3b 00 f2 e7 4c 00 12
|.i...;...L..    |
```

Using the Menus

You cannot use the **odm** menus to view data capture files.

You may use the **odm** menus to display help information about data capture operations by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other ILD Utility Help> from the Test type CHOICES menu
<Data Capture>
```

STREAMS Error Logging

The Integrated LAN Driver (ILD) uses the STREAMS error logging facility. All error messages, including those for the ILD, are logged in the file */usr/adm/streams/error.date* where *date* is the day the errors were logged.

All error messages in the STREAMS log have the format shown in the following example.

```
000001 08:37:57 0000332d ... -32762 3010
60002242|comm|1|H|W|O|
1|1|M|1|6|ncrtr_cmd_status()|32768|32768|0|1
#ILD: ncrtr.c-1549: PPA 1: Lobe Media Test Failed

000002 09:46:32 000053ee ... -32762 2100
60002100|comm|1|H|W|O|
1|1|M|1|5|wdent_tx_timeout()|32768|32768|0|1
#ILD: ild_wd.c-915: PPA 0: Transmit Timeout
```

The information up to the pound sign (#) is the common header. See the *NCR UNIX SVR4 Message Manual* for an explanation of this part of the message. The information following the pound sign is the Integrated LAN driver portion of the message. It has the following format:

```
# ILD:file-line_number:PPAppa:error_message
```

The first field after the pound sign identifies this error as being specific to the Integrated LAN drivers. Its value is always ILD for Integrated LAN driver errors.

The second field specifies the file in which the error occurred; the *line_number* field identifies the line within that file where the error occurred.

The *ppa* specifies the relative physical position of the adapter channel, beginning with zero, from left to right and top to bottom.

The *error_message* is the actual error message text. Refer to the “Error Messages” appendix in the *Integrated LAN Driver Installation and Diagnostics* book for a list of possible error messages and their meanings.

Chapter 15

Testing RBS WANs

Accessing RBS WAN Diagnostics

In-Service (Level 2) diagnostics permit you to perform basic RBS WAN activities. These in-service diagnostic test programs interact with the 2780/3780 driver and the MPCA through various **ioctl** commands and driver utilities.

The RBS WAN software includes diagnostic device tests, data capture operations and error logging capabilities that you may use to gather network or driver statistics or to exercise the system's communications hardware. You can perform RBS WAN diagnostic tasks from the command line or through the **odm** Online Diagnostics and Maintenance menus**.**

Command Line Tests

You can perform all the tasks described in this chapter by using command line entries.

For additional information about performing RBS LAN diagnostics from the command line, refer to the *Installation and Diagnostics* book(s) for the hardware controllers (such as the MPCA and/or MCIA adapters) that provide the WAN interface with the operating system software.

odm Menu Tests

You can perform the following tests through the **odm** Online Diagnostics and Maintenance menus**:**

- Loop 1
- Loop 2
- Loop 4
- Display adapter status
- Reset the adapter
- Read tally values

You can also use the **odm** menus to display help information about running data capture, viewing tallies, and displaying driver status.

Using the Maintenance Menus

You can perform some of the most commonly used 2780/3780 diagnostics and troubleshooting tasks from the **odm** Online Diagnostics and Maintenance menus.

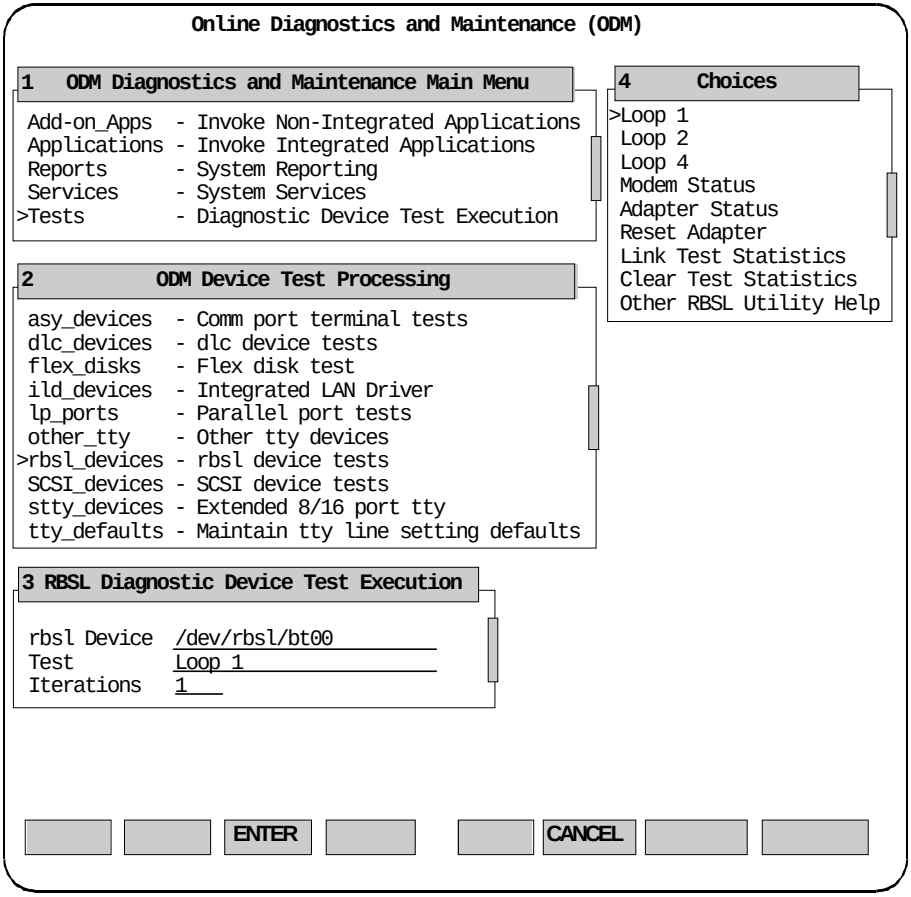
The following sequence of menu selections permits you to select RBS WAN tests through the **odm** menus. An example of the display created by these menu selections is depicted in the next figure. These menu selection steps are the same for all RBS WAN tests that may be selected through the menus.

1. Select <Tests> from the initial menu**.**
2. Select <rbsl_devices> from the ODM Device Test Processing menu.
3. If you know the name of the WAN device, type that name and press ENTER. If you do not know the device name, select CHOICES for a list of available device names, select your choice, and press ENTER to proceed to the *Test identification* field.

   2780/3780 devices have names in the form */dev/rbsl/bt00x*, where *x* is 1 through 4.

   4. If you know the test to perform, enter the desired test name, and press ENTER. If you want to see the tests available from the **odm** menus, press CHOICES, select the desired test, and press ENTER. Only one test can be selected from the CHOICES menu. The remaining sections of this chapter provide information about the RBS WAN tests.
5. If you want to perform the test more than one time, use the arrow keys to move the cursor to the *Iterations* field and enter the number of times you want to perform the test.
6. Press the SAVE function key to activate the selected test for the selected number of iterations.

Figure 15-1. RBS WAN Test Menu Selections



Performing the Loop1 Test

The **loop1** is the simplest device test. It verifies the adapter's internal UART. This test generates its own internal clocking and requires no external hardware.

The **loop1** test generates a test pattern which is sent and looped back either under software control or under manual control. It then compares the received pattern to the pattern it sent.

Using the Command Line

Use the following procedure to run the **loop1** test from the command line prompt:

1. Make sure the directory */opt/net/mpd* is in your search path.
2. Create a file containing your test parameters. You must specify values for the parameters shown in the following table. You must enter the parameters in a KEYWORD=*value* format.

Table 15-1. Loop 1 Parameters

KEYWORD	VALUE
BITRATE	Specifies the bitrate to be used for loop tests 1, 2 and 4. The acceptable bitrate values are 1200, 2400, 4800, 9600, and 19200.
ITERATIONS	Specifies the number of loop tests to run before terminating. Values range from 1 to 65000. The default value is 1.
PATTERN	Specifies a string of characters that are assembled into a pattern to loop. If the string begins with a single quote mark, the pattern is assembled as an ASCII string. A terminating quote is not needed. If the string does not begin with a single quote, the characters should consist of pairs of hexadecimal digits. The default string value is 00551199.
NRZI	Specifies whether or not to use NRZI encoding. Acceptable values are Y if NRZI encoding is used and N if it is not.

If you do not create this file, **loop1** will read the options from standard input.

3. Enter the following command to begin testing:

   ```
   # /opt/net/mpd/loop1 -i parameter_file -l device
   ```

   Valid devices are */dev/mpd00* through */dev/mpd03*.

   For each iteration, **loop1** displays the message "LOOP TEST SUCCESSFUL" if the test iteration was successful or "LOOP TEST FAILED" if the test iteration failed.

Using the Menus

You may use the **odm** menus to run the **loop1** test by selecting the **loop1** test during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter.

Performing the Loop2 Test

The **loop2** test verifies the adapter UART. In addition, it also tests the modem cable, modem circuitry, and the telephone line. This test requires a synchronous modem or similar device to provide external clocking. Before running this test, you must manually create an electrical loopback by either throwing a modem switch or by placing a loopback connection on the modem's analog interface. After testing, be sure to remove the loopback connector before resuming normal communications**.**

The **loop2** test generates a test pattern which is sent and looped back either under software control or under manual control. It then compares the received pattern to the pattern it sent.

Using the Command Line

Use the following procedure to run the **loop2** test from the command line prompt:

1. Make sure the directory */opt/net/mpd* is in your search path.
2. Create a file containing your test parameters. You must specify values for the parameters shown in the following table. You must enter the parameters in a KEYWORD=*value* format.

Table 15-2. Loop 2 Parameters

KEYWORD	VALUE
BITRATE	Specifies the bitrate to be used for loop tests 1, 2 and 4. The acceptable bitrate values are 1200, 2400, 4800, 9600, and 19200.
ITERATIONS	Specifies the number of loop tests to run before terminating. Values range from 1 to 65000. The default value is 1.
PATTERN	Specifies a string of characters that are assembled into a pattern to loop. If the string begins with a single quote mark, the pattern is assembled as an ASCII string. A terminating quote is not needed. If the string does not begin with a single quote, the characters should consist of pairs of hexadecimal digits. The default string value is 00551199.

KEYWORD	VALUE
NRZI	Specifies whether or not to use NRZI encoding. Acceptable values are Y if NRZI encoding is used and N if it is not.

If you do not create this file, **loop2** will read the options from standard input.

3. Enter the following command to begin testing:

```
# /opt/net/mpd/loop2 -i parameter_file -l device
```

Valid devices are */dev/mpd00* through */dev/mpd03*.

For each iteration, **loop2** displays the message "LOOP TEST SUCCESSFUL" if the test iteration was successful or "LOOP TEST FAILED" if the test iteration failed.

Using the Menus

You may use the **odm** menus to run the **loop2** test by selecting the **loop2** test during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter.

Performing the Loop4 Test

The **loop4** test verifies the adapter UART and the RS-232 interface. This test does not require external clocking, but you must place a loopback connector on the RS-232 interface before running the test. This test can also check modem cable continuity when the loopback connector temporarily replaces the modem. Be sure to remove this connector when testing is complete.

The **loop4** test generates a test pattern which is sent and looped back either under software control or under manual control. It then compares the received pattern to the pattern it sent.

Using the Command Line

Use the following procedure to run the **loop4** test from the command line prompt:

1. Make sure the directory */opt/net/mpd* is in your search path.
2. Create a file containing your test parameters. You must specify values for the parameters shown in the following table. You must enter the parameters in a KEYWORD=*value* format.

Table 15-3. Loop 4 Parameters

KEYWORD	VALUE
BITRATE	Specifies the bitrate to be used for loop tests 1, 2 and 4. The acceptable bitrate values are 1200, 2400, 4800, 9600, and 19200.
ITERATIONS	Specifies the number of loop tests to run before terminating. Values range from 1 to 65000. The default value is 1.
PATTERN	Specifies a string of characters that are assembled into a pattern to loop. If the string begins with a single quote mark, the pattern is assembled as an ASCII string. A terminating quote is not needed. If the string does not begin with a single quote, the characters should consist of pairs of hexadecimal digits. The default string value is 00551199.

KEYWORD	VALUE
NRZI	Specifies whether or not to use NRZI encoding. Acceptable values are Y if NRZI encoding is used and N if it is not.

If you do not create this file, **loop4** will read the options from standard input.

3. Enter the following command to begin testing:

```
# /opt/net/mpd/loop4 -i parameter_file -l device
```

Valid devices are */dev/mpd00* through */dev/mpd03*.

For each iteration, **loop4** displays the message "LOOP TEST SUCCESSFUL" if the test iteration was successful or "LOOP TEST FAILED" if the test iteration failed.

Using the Menus

You may use the **odm** menus to run the **loop4** test by selecting the **loop4** test during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter.

Displaying Adapter Status

Displaying adapter status produces a display like the following example.

```
-----------------------------------------------
Adapter State              = [ READY/DEAD ]
Micro Channel Slot         = 1
Adapter Version            = 01.01-A
Adapter IRQ Level          = 10
Adapter Queue Size         = 5032

Adapter Register Addresses:
Base I/O Address           = 0x380:
Message-In Register        = 0x382:
Message-Out Register       = 0x383:
Control Register           = 0x384:
Status Register            = 0x386:
Local Address Pointer      = 0x385:
Remote Address Pointer     = 0x387:
Reset Control Register     = 0x38c:
-----------------------------------------------
```

The *Adapter State* indicates the status of the adapter. If the status is READY, the adapter can be used. If the status is DEAD, the adapter failed level 0 diagnostics and cannot be accessed by any driver.

The *Micro Channel Slot* number shows the bus slot occupied by the adapter.

The *Adapter Version* is the version number of the adapter firmware.

The *Adapter IRQ Level* is the adapter's interrupt request level and should match the interrupt level specified in the UNIX kernel configuration file */etc/conf/sdevice.d/mpd.*

The *Adapter Queue Size* shows the memory capacity of the adapter in bytes.

The *Adapter Register Addresses* shown should agree with the address configured for the MPCA with the Micro Channel Reference Disk tool. The address labeled *Base I/O Address* is configured with the reference disk and the remaining addresses are based on it.

Using the Command Line

Enter the following command to display the adapter status:

```
# /opt/net/mpd/query -l device
```

The *device* argument specifies the MPCA device to query. Valid devices are */dev/mpd00* through */dev/mpd03*.

Using the Menus

You may use the **odm** menus to display the current adapter status by selecting **Adapter Status** during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter.

Resetting the Adapter

Resetting the adapter produces a display like the following example.

```
-----------------------------------------------
Reset of adapter [ 0/1 ] [ SUCCESSFUL/FAILED ]

Level 0 diagnostic byte 1 = 0xff
Scan List/Direction Test   [ Passed/Failed ]
Line Turnaround Test       [ Passed/Failed ]
EPROM CCITT FCS Test       [ Passed/Failed ]
Memory Access Test         [ Passed/Failed ]
Memory Retention Test      [ Passed/Failed ]
Register Access Test       [ Passed/Failed ]
Register Retention Test    [ Passed/Failed ]
Condition Code Test        [ Passed/Failed ]

level 0 diagnostic byte 2 = 0x40
Real Time Clock Test       [ Passed/Failed ]

Adapter State              = [ READY/DEAD ]
Micro Channel Slot         = 1
Adapter Version            = 01.01-A
Adapter IRQ Level          = 10
Adapter Queue Size         = 5032
-----------------------------------------------
```

The *Adapter State* indicates the status of the adapter. If the status is READY, the adapter can be used. If the status is DEAD, the adapter failed level 0 diagnostics and cannot be accessed by any driver.

The *Micro Channel Slot* number shows the bus slot occupied by the adapter.

The *Adapter Version* is the version number of the adapter firmware.

The *Adapter IRQ Level* is the adapter's interrupt request level and should match the interrupt level specified in the UNIX kernel configuration file */etc/conf/sdevice.d/mpd*.

The *Adapter Queue Size* shows the memory capacity of the adapter in bytes.

Using the Command Line

Enter the following command to reset an MPCA device (adapter), execute level 0 diagnostics, and display the results of the diagnostic tests:

```
# /opt/net/mpd/reset -l device
```

The *device* argument specifies the MPCA device to reset. Valid devices are */dev/mpd00* through */dev/mpd03*.

Using the Menus

You may use the **odm** menus to reset the adapter by selecting the **Reset Adapter** option during step 4 of the procedure described in the “Using the Maintenance Menus” section of this chapter.

Reading Tally Values

In the following example shows how tallies are displayed when you read tally values for an RBS device.

```
Tallies for device /dev/mpd00
=============================

ID of last driver to access tallies     : 0x11
 1.  Blocks received                    : 1111
 2.  Blocks sent                        : 1111
 3.  Total blocks sent and received     : 2222
 4.  Blocks received with CRC error     : 0
 5.  WACK's sent                        : 0
 6.  Inputs aborted due to WACK's sent  : 0
 7.  Receiver overruns                  : 0
 8.  Timeouts in establishment phase    : 9
 9.  Blocks re-transmitted              : 0
10.  WACK's sreceived                   : 0
11.  Timeouts waiting for ACK           : 0
12.  TTD's sent                         : 0
13.  Ouputs aborted due to TTD limit    : 0
14.  Transmitter underruns              : 0
15.  Circuit assurance failure          : 0
16.  DSR losses                         : 0
17.  CTS losses                         : 0
18.  Carrier losses                     : 0
19.  Line connection failures           : 0
20.  MPCC failures                      : 0
```

Using the Command Line

Enter the following command to display the standard set of network tallies that are maintained for all WAN drivers:

```
# /opt/net/mpd/tally -p2 -l device
```

The **-p2** argument specifies that you want to view 2780/3780 protocol data.

The *device* argument specifies the name of the device you want to see the tallies for. Valid devices are */dev/mpd00* through */dev/mpd03*.

You may also use the **-c** argument to reset tally values to zero after displaying them.

The following table describes the tally ID numbers displayed on the screen.

Table 15-4. RBS Device Tally IDs

ID	Tally	Description
1	Blocks received	This tally indicates the number of error free blocks received and accepted from the other station.
2	Blocks sent	This tally indicates the number of blocks accepted and acknowledged by the other station. Non-zero values are expected and indicate the amount of output traffic.
3	Total blocks sent and received	This field represents the total of tallies 1 and 2 above.
4	Frames received with CRC errors	This tally is normally zero or very low. It counts frames received with transmission errors, such as noisy lines.
5	WACK's sent	Non-zero values of this tally indicate pacing problems; the application cannot accept data as fast as it is received from the remote station. Blocks which are discarded by this station must be retransmitted by the other.
6	Inputs aborted due to WACK's sent	Non-zero values indicate pacing problems; the maximum number of WACKs has been reached. Indicates that the application consistently cannot accept data as fast as it is received from the remote station.
7	Receiver overruns	Non-zero values indicate that aggregate line speeds are too great for the multiprotocol adapter, or too few buffers have been configured.
8	Timeouts in establishment phase	This tally counts the number of times the remote station failed to answer a poll during the establishment phase. A non-zero value indicates possible line problems, such as noise.

ID	Tally	Description
9	Blocks retransmitted	This tally counts the number of blocks which the remote rejected, causing them to be retransmitted. Non-zero values indicate problems in output to the remote.
10	WACK's received	Non-zero values indicate pacing problems; the application is sending data faster than the remote station can accept it..
11	Timeouts waiting for ACK	This tally counts the number of times the remote station failed to answer a poll after it was successfully contacted. A non-zero value indicates possible line problems, such as noise.
12	TTD's sent	Non-zero values indicate pacing problems; the application cannot generate blocks as quickly as the transmission process delivers them.
13	Outputs aborted due to TTD limit	Non-zero values indicate pacing problems; the maximum number of TTDs has been reached. Indicates that the application consistently cannot generate blocks as quickly as the transmission process delivers them.
14	Transmitter underruns	This tally counts the number of times the adapter had to abort a transmission because it could not provide data to its USART rapidly enough. This is symptomatic of an overloaded adapter or, occasionally, of modem cable disconnections.
15	Circuit assurance failure	The number of times the received circuit assurance ID does not agree with the one expected.
16	DSR losses	This tally counts the number of times communications were interrupted due to loss of modem DSR.
17	CTS losses	This tally counts the number of times communications were interrupted due to loss of modem CTS.

ID	Tally	Description
18	Carrier losses	This tally counts the number of times the remote carrier was lost on duplex line. Non-zero values indicate line drops. This tally is enabled only if the station is configured to honor carrier loss.
19	Line connection failures	This tally indicates the number of times the adapter initially asserted modem signal DTR, but modem signal DSR did not go active after waiting the interval of the connect line timer.
20	MPCC failures	This tally counts the number of unexpected adapter resets that have been encountered.

Using the Menus

You may use the **odm** menus to display tally values by selecting the **Link Test Statistics** during step 4 of the procedure described in the "Using the Maintenance Menus" section of this chapter. To reset tally values to zero after displaying them, use the **Clear Test Statistics** selection item.

Starting Data Capture

The start data capture operation captures and stores data that is transmitted and/or received on a selected device. This operation stores the raw captured data in a file that can be used later to help determine the accuracy level of the communication path to and/or from the device.

Using the Command Line

Enter the following command to start data capture:

```
# /opt/net/mpd/rundcap device > /tmp/datacap.raw &
```

This command causes the driver to begin writing data capture information to a raw data capture file. The raw data capture file specified above is */tmp/datacap.raw*, but you may specify any name you like.

The *device* argument specifies the MPCA device to query. Valid devices are */dev/mpd00* through */dev/mpd03*.

You may capture only data received by the driver, only data transmitted by the driver, or both. If you want to capture only received data, use the **-r** option. If you want to capture only transmitted data, use the **-t** option. If you do not specify either of these options, **rundcap** captures both received and transmitted data.

You may also specify the maximum message size, in bytes, to capture. To do this, use the **-m** *max_message_size* option, which causes only the first *max_message_size* bytes to be captured.

Note: The output from this command is NOT in readable form. To read the data capture information, use the **hdumper**(1M) command. See the "Viewing Data Capture Files" section of this chapter for more information.

If you are running data capture on a line with a great deal of communications traffic, do not let your raw data capture file grow too large. If the file gets too large, this may cause performance or storage problems when you view the file with **hdumper**(1M).

Using the Menus

You can not use the **odm** menus to start data capture operations.

You may use the **odm** menus to display help information about data capture operations by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other DLC Utility Help> from the Test type CHOICES menu
<Data Capture>
```

Stopping Data Capture

The **stop data capture** operation terminates active data capture processes.

Using the Command Line

Use the following procedure to stop the **rundcap** process from writing data capture information to a file:

1. To determine the process ID (PID) for the **rundcap** process, enter the following command:

   ```
   # ps -eaf | grep rundcap
   ```

2. To stop the **rundcap** process from running, enter the following command:

   ```
   # kill -9 PID
   ```

   The *PID* argument is the process ID retrieved in step number 1.

Using the Menus

You can not use the **odm** menus to stop data capture operations.

You may use the **odm** menus to display help information about data capture operations by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other DLC Utility Help> from the Test type CHOICES menu
<Data Capture>
```

Viewing Data Capture Files

The **view data capture** operation displays the contents of a raw data capture file.

Using the Command Line

You may use commands to generate the following types of data capture file displays:

- Display Entire File
- Display Errors Only
- Display Summary Information Only
- Display Specific Sequence Numbers Only
- Display Minimum Message Size
- Display Real-Time Data Capture

Display Entire File

To display the entire contents of a raw data capture file on your screen, enter the following command:

```
# /opt/net/mpd/hdumper -p2 raw_data_capture_filename
```

The **-p2** argument specifies that you want to view 2780/3780 protocol data. The *raw_data_capture_filename* argument specifies the name of the raw data capture file used in the **rundcap** command.

Display Errors Only

To display only blocks or frames which are in error, enter the following command:

```
# /opt/net/mpd/hdumper -p2 -e  raw_data_capture_filename
```

The **-p2** argument specifies that you want to view 2780/3780 protocol data. The *raw_data_capture_filename* argument specifies the name of the raw data capture file used in the **rundcap** command.

Display Summary Information Only

To display only a summary of the number messages captured and errors detected, enter the following command:

```
# /opt/net/mpd/hdumper -p2 -s raw_data_capture_filename
```

The **-p2** argument specifies that you want to view 2780/3780 protocol data. The *raw_data_capture_filename* argument specifies the name of the raw data capture file used in the **rundcap** command.

Display Specific Sequence Numbers Only

To display only data capture frames within a certain range of sequence numbers, enter the following command:

```
# /opt/net/mpd/hdumper -p2 -f first -l last    \
raw_data_capture_filename
```

The **-p2** argument specifies that you want to view 2780/3780 protocol data. The **-f** *first* option specifies the sequence number of the first frame you wish to display; the **-l** *last* option specifies the sequence number of the last frame you want to display. Only the frames with sequence numbers in this range are displayed.

The *raw_data_capture_filename* argument specifies the name of the raw data capture file used in the **rundcap** command**.**

Display Minimum Message Size

You may also specify the minimum message size, in bytes, to be displayed. Only messages larger than the value in this field appear on your screen. This feature allows you to suppress lengthy exchanges of protocol polling sequences, since these messages are usually very short.

To specify a minimum message size, include the **-m** *min_message_size* option with any of the preceding commands.

Display Real-Time Data Capture

If you wish to start data capture and view the output in real time, enter the following command:

```
# rundcap device | hdumper -p2 &
```

The **-p2** argument specifies that you want to view 2780/3780 protocol data. The *device* argument specifies the MPCA device to query. Allowable devices are */dev/mpd00* through */dev/mpd03*.

See the “Interpreting Data Capture” section of this chapter for information on how to interpret the data capture information displayed.

Using the Menus

You can not use the **odm** menus to view data capture files.

You may use the **odm** menus to display help information about data capture operations by selecting the following functions during the procedure described in the “Using the Maintenance Menus” section of this chapter:

```
<Other DLC Utility Help> from the Test type CHOICES menu
<Data Capture>
```

Interpreting Data Capture

The following example shows a converted data capture file.

The first line of the data capture file lists the name of the device the data was captured from and the date and time the data capture started. The file then lists the description and contents of each message sent or received from the time data capture was started until it was stopped. The data capture utility displays the contents of each message in hexadecimal characters and in the corresponding EBCDIC values.

```
DEVICE = mpd00    DATA CAPTURE STARTED: Thu Jan 31 14:26:09
1991

Seq=60670 Type=GOOD EBCDIC OUTPUT Drvr Stat=0000  Thu Jan 31
16:54:46 1991

0000:  10   70
      DLE   70

Seq=60671 Type=GOOD EBCDIC INPUT Drvr Stat=0000  Thu Jan 31
16:54:47 1991

0000:     02  F0  F0  F0  F0  F0  F1  C1  C1  C1  C1  C1  C1
C1  C1  C1
         STX   0   0   0   0   0   1   A   A   A   A   A   A
A   A   A
0010:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  C1
           A   A   A   A   A   A   A   A   A   A   A   A   A
A   A   A
0020:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  C1
           A   A   A   A   A   A   A   A   A   A   A   A   A
A   A   A
0030:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  C1
           A   A   A   A   A   A   A   A   A   A   A   A   A
A   A   A
0040:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  C1
           A   A   A   A   A   A   A   A   A   A   A   A   A
A   A   A
0050:     C1  1E  F0  F0  F0  F0  F0  F2  C1  C1  C1  C1  C1
C1  C1  C1
           A IRS   0   0   0   0   0   2   A   A   A   A   A
A   A   A
0060:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  C1
```

```
            A   A   A   A   A   A   A   A   A   A   A   A   A   A
A   A   A
0070:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1
            A   A   A   A   A   A   A   A   A   A   A   A   A   A
A

Seq=60671 Type=GOOD EBCDIC INPUT Drvr Stat=0000  Thu Jan 31
16:54:47 1991

0000:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  C1
            A   A   A   A   A   A   A   A   A   A   A   A   A   A
A   A   A
0010:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  1E  F0
F0  F0  F0
            A   A   A   A   A   A   A   A   A   A   A IRS   0
0   0   0
0020:     F0  F6  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  1D
            0   6   A   A   A   A   A   A   A   A   A   A   A   A
A   A IGS
0030:     4A  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  C1
            4A   A   A   A   A   A   A   A   A   A   A   A   A   A
A   A   A
0040:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  C1
            A   A   A   A   A   A   A   A   A   A   A   A   A   A   A
A   A
0050:     C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1  C1
C1  C1  C1
            A   A   A   A   A   A   A   A   A   A   A   A   A   A
A   A   A
0060:     C1  C1  C1  C1  1E  26
            A   A   A   A IRS ETB

Seq=60672 Type=GOOD EBCDIC OUTPUT Drvr Stat=0000 Thu Jan 31
16:54:47 1991

0000:  10  61
      DLE   /
```

Message Description

For each message received or transmitted, the data capture facility displays a description of the message. The description includes the following information:

- Sequence number
- Type
- Driver status
- Date and time

The following sections describe the information in these message fields.

Sequence Number

The sequence number indicates the sequence of messages in the data capture file. Sequence numbers begin with 0000 and continue sequentially until you stop data capture.

Type

The type field describes the type of data in the message. Error-free messages have a type of “good.” Messages containing errors have types of “bad” or “error.” Received messages are designated as “input,” while sent messages have a type of “output.”

Driver Status

When the 2780/3780 driver encounters an error or exception condition, you can interpret the driver status code to identify the error. A driver status code of 0000 indicates that the driver has not encountered any errors or exception conditions.

If the driver status code field contains a non-zero value, the next line of the data capture file contains a brief description of the status code’s meaning. Sometimes the description will not give you enough information to diagnose a problem and you will need to decipher the status code. The following table shows the 2780/3780 driver status codes.

Table 15-5. 2780/3780 Driver Status Codes

Hex Value	Status	Description
10	DSRDN	Unexpected DSR drop
03	DLEEOT	DLE end of transmission
14	NOBUF	No receive buffers are available
15	FRMLONG	Incoming frame is longer than the receive buffer
0A	DCDDROP	Unexpected DCD drop
0B	CTSDROP	Unexpected CTS drop

Date and Time

This field lists the date and time that the message was either transmitted or received.

Message Contents

After the message description, the next lines display the contents of each message in hexadecimal characters and in the corresponding EBCDIC characters. The following shows an example of data capture message contents.

```
0000:     02  F0  F0  F0  F0  F0  F1  C1  C1  C1  C1  C1  C1
C1  C1  C1
         STX   0   0   0   0   0   1   A   A   A   A   A   A
A   A   A
     .
     .
     .
0060:     C1  C1  C1  C1  1E  26
           A   A   A   A IRS ETB
```

Some bytes contain control characters instead of alphabetic characters, such as STX, IRS, and ETB in the example. The following table lists all possible control characters.

Table 15-6. Data Capture Control Characters for RBS WANs

Control Characters			
ACK	Acknowledge	IL	Idle
BEL	Bell	IRS	Information Record Separator
BS	Backspace	IUS	Information Unit Separator
BYP	Bypass	LC	Lower Case
CAN	Cancel	LF	Line Feed
CC	Cursor Control	NAK	Negative Acknowledge
CR	Cursor Return	NL	New Line
DC1	Device Control 1	NUL	Null
DC2	Device Control 2	PF	Punch Off
DC4	Device Control 4	PN	Punch On
DEL	Delete	RES	Restore
DLE	Data Link Escape	RS	Reader Stop
DS	Digit Select	SI	Shift In
EM	End of Media	SM	Set Mode
ENQ	Enquiry	SMM	Start of Manual Message
EOT	End of Transmission	SO	Shift Out
ETB	End of Transmission Block	SOH	Start of Heading
ETX	End of Text	SOS	Start of Significance
FF	Form Feed	STX	Start of Text
FS	Field Separator	SUB	Substitute
HT	Horizontal Tabulation	SYN	Synchronous
IFS	Information Field Separator	UC	Uppercase
IGS	Information Group Separator	VT	Vertical Tabulation

Appendix A

Application Integration

Overview

Appendix A describes the steps to follow to incorporate other diagnostic and maintenance functions into the **odm** menu interface. These diagnostic functions are accessed from the following selections in the **odm** initial menu:

- <Add-on_Apps>
- <Applications>

The integration tasks would normally be performed as part of the installation stream of a particular product. The system administrator also has the capability to incorporate access to diagnostic or maintenance scripts into the <Add-on_Apps> menu selections.

Non-Integrated Applications

Non-integrated applications are third-party packages that provide diagnostic and maintenance functions that *do not interface* to the Operations, Administration, and Maintenance menu system (**sysadm**), or the Operations Advantage Base menu system (**osa**). All diagnostic and maintenance functions for the package must be accessed from **odm** through an executable program installed by the package.

Non-integrated applications may also be custom diagnostic or maintenance scripts or executables which have been created for the system, and which the system administrator would like to incorporate into the **odm** interface.

To include access to a third-party or custom diagnostic executable program in the menu built by the **odm** <Add-on_Apps> initial menu selection, perform the following steps:

1. The executable file that controls the diagnostic/maintenance functions for the package must be named:

   <pkginst>.odm

where <*pkginst*> is the name of the third party package. If this is a custom diagnostic program, which is not associated with a particular third party package, <*pkginst*> should be a meaningful name which does not coincide with the name of any package. This name will appear in the "ODM Non-Integrated Applications" menu.

2. The executable file <*pkginst*>.*odm* must be placed in the directory:

   /opt/odm/add-on

   The file should have at least **read** and **execute** permission for **owner**.

When <Add-on_Apps> is selected from the **odm** initial menu, an "ODM Non-Integrated Applications" menu is dynamically built based upon the **.odm* files found in the directory */opt/odm/add-on*. For example, if the file */opt/odm/add-on/XYZ.odm* exists, the menu item "XYZ - XYZ Diagnostics" will be added to the "ODM Non-Integrated Applications" menu.

Integrated Applications

Integrated applications provide links to **fmli**-based (Form and Menu Language Interpreter) diagnostic and maintenance functions available through either the **sysadm** or **osa** menu system. **odm** provides the means to access these **fmli**-based diagnostics through the <Applications> selection in the **odm** initial menu. When <Applications> is selected, a list of integrated applications is built and presented in an "ODM Applications" menu.

To include an integrated application in the "ODM Applications" menu, perform the following:

1. Create a configuration file for the application, named <*pkginst*>.*conf*, where <*pkginst*> is the package name of the application. This file has the following contents:

```
name="aaa"
description="bbb"
action=open menu \
${ODM_HOME}/menu/applications/<pkginst>/Menu.<pkginst> \
/usr/sadm/sysadm/add-ons/xxx Menu.<pkginst>
lininfo="${OAMBASE}/add-ons/xxx \"bbb\"" <pkginst>
```

   where:

- ***aaa*** is the name that will appear in the “ODM Applications” menu
- ***bbb*** is the description that appears next to the name *aaa* in the “ODM Applications” menu.
- ***<pkginst>*** is the name of the package
- ***xxx*** is the relative path to the **fmli** frames
- the “\” character at the end of a line indicates continuation of the current line on the next line

2. The configuration file <*pkginst*>.*conf* must be placed in the directory:

   /opt/odm/usr/intf_install/applications

3. Create a menu file for the application, named *Menu.*<*pkginst*>, where <*pkginst*> is the package name of the application. The file has the following contents:

```
`/usr/sadm/sysadm/bin/object_gen $ARG1 $ARG2;`
`if [ “${INTERFACE}” != “motif” ];
then
     set -e SYSDIR=${OAMBASE}/menu;
     set -e SYSSTART=main.menu;
     /usr/sadm/sysadm/bin/object_gen ${ARG1} ${ARG2};
else
     set -e SYSDIR=${OSABASE}/add-ons/xxx;
     set -e SYSSTART=Menu.<pkginst>;
     /usr/sadm/osa/bin/osa_gen ${ARG1} ${ARG2};
fi`
```

   where:

   - ***<pkginst>*** is the name of the package
   - ***xxx*** is the path to the **fmli** frames

4. The menu file *Menu.*<*pkginst*> must be placed in the directory:

   /opt/odm/menu/applications/<*pkginst*>

When <Applications> is selected from the **odm** initial menu, an “ODM Applications” menu is dynamically built based upon the information in the **.conf* files found in the directory */opt/odm/usr/intf_install/applications*.

For the **madf** package, the contents of the configuration file, named *madf.conf*, would look like that depicted in the figure below.

Figure A-1. madf.conf Configuration File

```
name="MCA_board"
description="Micro Channel Board Configuration"
action=open menu \
${ODM_HOME}/menu/applications/madf/Menu.madf \
/usr/sadm/sysadm/add-ons/madf/machinemgnt/configmgnt/MCA_board\
 Menu.madf
lininfo="${OAMBASE}/\
add-ons/madf/machinemgnt/configmgnt/MCA_board" \
\"Micro Channel Board Configuration\"" madf
```

The contents of the menu file for the **madf** package, named *Menu.madf*, would look like that depicted in the figure below.

Figure A-2. Menu.madf Menu File

```
`/usr/sadm/sysadm/bin/object_gen $ARG1 $ARG2;`
`if [ "${INTERFACE}" != "motif" ];
then
     set -e SYSDIR=${OAMBASE}/menu;
     set -e SYSSTART=main.menu;
     /usr/sadm/sysadm/bin/object_gen ${ARG1} ${ARG2};
else
     set -e SYSDIR=${OSABASE}/\
         add-ons/madf/machinemgnt/configmgnt/MCA_board;
     set -e SYSSTART=Menu.madf;
     /usr/sadm/osa/bin/osa_gen ${ARG1} ${ARG2};
fi`
```

Index

D

E

I

K

L

M

N

O

P

R

S

T

U

V

W