Sun Enterprise 10000 DR Error Messages

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto,CA 94303-4900
U.S.A. 650-960-1300

Part No. 806-5229-10
January 2001, Revision A

Send comments about this document to: docfeedback@sun.com

Please
Recycle

Sun Enterprise 10000 SSP Attributions:

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim inany distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

RESTRICTED RIGHTS: Use, duplication or disclosure by the government is subject to the restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software Clause as DFARS 252.227-7013 and FAR 52.227-19.

This is scotty, a simple tcl interpreter with some special commands to get information about TCP / IP networks. Copyright (c) 1993, 1994, 1995, J. Schoenwaelder, TU Braunschweig, Germany, Institute for Operating Systems and Computer Networks. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that this copyright notice appears in all copies. The University of Braunschweig makes no representations about the suitability of this software for any purpose. It is provided “as is" without express or implied warranty.

Contents

Tables

Preface

This book contains the Dynamic Reconfiguration (DR) and InterDomain Network (IDN) error messages that occur on Sun Enterprise™ 10000 domains.

Before You Read This Book

This book is intended for the Sun Enterprise 10000 server system administrator who has a working knowledge of UNIX® systems, particularly those based on the Solaris™ operating environment. If you do not have such knowledge, first read the Solaris user and system administrator books in AnswerBook2™ format provided with this system and consider UNIX system administration training.

How This Book Is Organized

This book contains the following chapters:

Chapter 1 introduces the DRerror messages.

Chapter 2 contains the DR error messages that occur on the domain.

Using UNIX Commands

This document may not contain information on basic UNIX commands and procedures such as shutting down the system, booting the system, and configuring devices.

Refer to one or more of the following for this information:

- AnswerBook2 online documentation for the Solaris software environment
- Other software documentation that you received with your system

Typographic Conventions

Typeface or Symbol	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`% su` `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the *User's Guide.* These are called *class* options. You *must* be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type `rm` *filename.*

Shell Prompts

Shell	Prompt
C shell	*machine_name*%
C shell superuser	*machine_name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#

Related Documentation

Application	Title	Part Number
User	*Sun Enterprise 10000 SSP 3.4 User Guide*	806-4870
	Sun Enterprise 10000 Dynamic Reconfiguration User Guide	806-4122
Reference	*Sun Enterprise 10000 SSP 3.4 Reference Manual*	806-4871
	Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual	806-4123
Release Notes	*Solaris 8 1/01 Release Notes Supplement for Sun Hardware*	Printed in Media Kit.

Accessing Sun Documentation Online

The `docs.sun.com`SM web site enables you to access Sun technical documentation on the Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

`http://docs.sun.com`

Ordering Sun Documentation

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatrain.com at:

`http://www.fatbrain.com/documentation/sun`

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

`docfeedback@sun.com`

Please include the part number (806-5229-10) of your document in the subject line of your email.

CHAPTER 1

DR Errors Introduction

This chapter contains an introduction to the dynamic reconfiguration (DR) error messages that occur on the domain.

Note – If you do not find the error message in this book, refer to the error messages in the *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* in the SSP 3.4 Answerbook collection.

Searching the Tables in This Book

Before you use the tables in this book, take time to read the following list of search tips so that you can find a specific message.

- Search on a specific string of text in the error message.
- Avoid using numeric values. They are treated as replaceable text in this book.
- Avoid using text that is replaceable. In this book, the following names are used to represent replaceable text in the error messages: *descriptive message*, *errno_description*. *device_name*, *target_path*, *mount_point*, *interface_name_instance*, *interface_name*, and *partition_name*.
- If you are reading this text in hard-copy form, the tables are presented in order by the type of error or failure. The contents of the tables is sorted alphabetically in decending order.

Online Searching

You can use the search engine provided in the AnswerBook2™ environment or the search engine in your browser to find a specific string of characters from an error message. Before you construct the search string, keep in mind that this appendix contains special typographical conventions. In addition, you may need to search all of the tables individually. If you know the error type (that is, where the error was encountered), use the hypertext links in “DR Error Messages on the Domain” on page 2 to start your search.

Special Typographical Conventions

The tables in this appendix contain special typographical conventions for the names of words and values that change, depending on the type of error. When you search for an error message, keep in mind that these names appear as generic representations in italic font. The following list contains the commonly used representations used in this appendix.

- *domain_ID* for the value of the domain ID
- *domain_name* for the names of all domains
- *platform_name* for the name of the Sun Enterprise™ 10000 platform
- *process_id* for the value of the process ID (pid number)
- *system_board_number* for the number of a system board (that is, 1 through 15)
- *number* for numeric values

DR Error Messages on the Domain

This book contains a list of some of the error messages that you might see while you are performing DR operations. The list does not include Protocol Independent Module (PIM) layer errors, which are more generic than the error messages in this book.

Use one of the following links to start your search.

CHAPTER 2

Domain DR Error Messages

All DR error messages are sent to the one or both of the following locations:

- SSP applications
- System error logs

DR Daemon Start-Up Errors

The following table contains a list of the DR daemon start-up errors. These messages are sent only to the domain console window.

TABLE 2-1 DR Daemon Start-Up Error Messages

Error Message	Probable Cause	Suggested Action
`Cannot create server handle`	The DR daemon could not start up the RPC server. You will see this message only if you manually execute the DR daemon without properly configuring the network services on the domain. Normally, network services spawn the DR daemon in response to an incoming RPC from the SSP.	On the domain, fix the `inetd.conf` entry for the DR daemon.
`Cannot fork:` *descriptive message*	The DR daemon could not fork a process from which to run its RPC server.	The descriptive error message corresponds to an *errno_value* and offers clues as to why the DR daemon could not fork off the RPC server. Check the resource limits and the load of the system to find a way to fix this error.

TABLE 2-1 DR Daemon Start-Up Error Messages

Error Message	Probable Cause	Suggested Action
`Permission denied`	A user other than root tried to run the DR daemon.	Only the superuser (root) can run the DR daemon because the daemon needs all of the root privileges to fully explore the system and to access the driver to detach and attach boards.
`Unable to register (300326, 4)`	The DR daemon was executed without being properly registered with the network services in the domain. The first number represents the RPC number that is registered for the DR daemon. The second number represents the RPC version used by the DR daemon.	On the domain, fix the `inetd.conf` entry for the DR daemon.
`Unable to create (300326, 4) for netpath`	The DR daemon was executed without being properly registered with the network services in the domain. The first number represents the RPC number that is registered for the DR daemon. The second number represents the RPC version used by the DR daemon.	On the domain, fix the `inetd.conf` entry for the DR daemon.

Memory Allocation Error Messages

The following table contains the memory allocation error messages that are sent to the system logs and to the SSP applications. Although the list contains several error messages, each of them describe one of two possible errors: ENOMEM or EAGAIN. All of the ENOMEM errors have the same suggested action, as do the EAGAIN errors.

TABLE 2-2 Memory Allocation Error Messages

Error Message	Probable Cause	Suggested Action
`DR Error: malloc failed (add notnet ap info)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the ps(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An ENOMEM error means that the DR daemon is in a state from which it cannot recover. An EAGAIN error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (alias_namelen)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the ps(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An ENOMEM error means that the DR daemon is in a state from which it cannot recover. An EAGAIN error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.

TABLE 2-2 Memory Allocation Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: malloc failed (AP ctlr_t array)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (ap_controller)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (board_cpu_config_t)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.

TABLE 2-2 Memory Allocation Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: malloc failed (board_mem_config_t)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (board_mem_cost_t)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
DR Error: malloc failed (board_mem_drain_t) errno_description	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.

TABLE 2-2 Memory Allocation Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: malloc failed (dr_io)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (leaf array)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (leaf)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.

TABLE 2-2 Memory Allocation Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: malloc failed (net_leaf_array)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (sbus_cntrl_t)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
DR Error: malloc failed (sbus_config) errno_description	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.

TABLE 2-2 Memory Allocation Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: malloc failed (sbus_device_t)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (sbus_usage_t)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. You may have to stop and restart the daemon. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (struct devnm)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.

TABLE 2-2 Memory Allocation Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: malloc failed (swap name entries)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (swaptbl)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon is larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.
`DR Error: malloc failed (unsafe_devs)` *errno_description*	While it queried the system information, the DR daemon could not allocate enough memory for a structure in which to return the requested information. The daemon may have encountered a resource limit. If the DR daemon cannot allocate memory, then it cannot continue to work. The *errno_description* usually describes an `ENOMEM` or `EAGAIN` error.	First, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon larger than the above memory sizes, then it may have a memory leak. If it does, you should report this problem. An `ENOMEM` error means that the DR daemon is in a state from which it cannot recover. An `EAGAIN` error means that the problem may have been temporary. You can retry the operation, which may succeed eventually, or you may have to stop and restart the daemon.

DR Driver Failures

The following table contains the DR driver failures that are sent to the system logs and to the SSP applications. In general, refer to the descriptions of the daemon and PSM errors for details about what goes to the system logs and what goes to the SSP.

Note – All of the possible DR driver failure messages are related to the three probable causes given in the table. Likewise, all of the failure messages have one suggested action.

TABLE 2-3 Memory Allocation Error Messages

Error Message	Probable Cause	Suggested Action

`DR: Error: initiate_attach: ioctl failed`
 	
`DR: Error: complete_attach: ioctl failed`
 	
`DR: Error: abort_attach: ioctl failed`
 	
`DR: Error: get_cpu_info: ioctl failed`
 	
`DR: Error: get_mem_config: ioctl failed`
	An `ioctl()` failure (that is, a failure that was encountered by the DR daemon when it tried to use the DR driver) can occur at three separate levels.
	
	At the first level—within the DR daemon, errors occur when the DR daemon and the DR driver are not interacting properly. The driver could be missing; the DR driver files in the `/devices/pseudo` directory could be missing, or the file permissions could be wrong. The DR daemon could also be experiencing memory corruption or resource limitations. The `ioctl()` failure message is followed by a message in the form: `Daemon (errno` *#error_number*`):` *error description*.
		The context of the `ioctl()` failure (that is, which function preceeds the `ioctl()` failed portion of the message), combined with the text of the error message, indicates what failed. Use the error number to identify the probable cause by checking the information on the `ioctl`(2) man page. You can also use the `/usr/include/errno.h` header file if the `ioctl`(2) man page does not have a specific reference for the error number.

TABLE 2-3 Memory Allocation Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action

`DR: Error: get_mem_cost: ioctl failed`
 	
`DR: Error: get_mem_drain: ioctl failed`
 	
`DR: Error: update_attach: ioctl failed`
 	
`DR: Error: ioctl failed, error draining resources`
 	
`DR: Error: detach_board: UNCONFIGURE ioctl failed`
 	
`DR: Error: detach_board: DISCONNECT ioctl failed`
 	
`DR: Error: abort_detach: CANCEL ioctl failed`
 	
`DR: Error: abort_detach: CONFIGURE ioctl failed`
 	
`DR: Error: get_dr_state: ioctl failed`
 	
`DR: Error: get_dr_status: ioctl failed`
	At the second level—within the platform independent module (PIM) layer of the DR driver, an ioctl failure could indicate busy resources, failing I/O devices on the system board, or improper interaction between the PIM and the platform specific module (PSM) layers. The `ioctl()` failure message is followed by a PIM message in the form: `PIM (error #errornumber):` *errno_description*.
	
	At the third level—the PSM layer, an `ioctl()` failure could indicate busy resources, failing I/O devices on the system board, memory detach failures, CPU detach failures, or internal failures encountered by the PSM driver. The error description usually cites specific physical devices that are failing or includes detailed explanations for a memory or CPU detachment failure. The `ioctl()` failure message is followed by a PSM message that appears in the following form: `PSM (error #errornumber):` *errno_description*.
	
	Note that failures in the PSM layer do not have corresponding `errno` values. PSM failure messages use an error number. You can find explanations of the error numbers in the `/usr/include/sys/sfdr.h` header file.
		See above.

PSM Error Messages

The following table contains a list of PSM error messages that are sent to the system logs and to the SSP applications.

TABLE 2-4 PSM Error Messages

Error Message	Probable Cause	Suggested Action
1 SFDR_ERR_INTERNAL	An internal driver failed.	None
2 SFDR_ERR_SUSPEND	Failed to suspend devices.	None
3 SFDR_ERR_RESUME	Failed to resume suspended devices.	None
4 SFDR_ERR_UNSAFE	Failed to quiesce the operating system due to referenced suspend-unsafe devices.	Determine the I/O usage of unsafe devices in the domain, and manually suspend the unsafe devices.
5 SFDR_ERR_UTHREAD	User thread could not be stopped.	Retry the operation. If this error persists, try stopping the process with the `kill`(1) command.
6 SFDR_ERR_RTTHREAD	Realtime thread could not be stopped.	Retry the operation. If this error persists, try stopping the process with the `kill`(1) command.
7 SFDR_ERR_KTHREAD	Kernel thread could not be stopped.	Retry the operation. If this error persists, try stopping the process with the `kill`(1) command.
8 SFDR_ERR_OSFAILURE	The kernel is not processing DR operations properly for the DR driver.	None
9 SFDR_ERR_OUTSTANDING	The `ioctl()` failed because an error from a previous DR drain operation still has not been reported through the DR status command.	Retry the operation.
11 SFDR_ERR_CONFIG	The current system configuration will not allow the DR operation to execute.	Check the `/etc/system` file to ensure that memory detach is enabled.
12 SFDR_ERR_NOMEM	Not enough memory	None
13 SFDR_ERR_PROTO	Protocol failure	None

TABLE 2-4 PSM Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
14 SFDR_ERR_BUSY	The device is busy.	Check the I/O usage of the device to determine the cause of this error (for example, a mounted file system or the last path to an AP device). If possible, manually adjust the system to correct this error (for instance, unmount the file system). If the cause of the error is not apparent, contact your Sun service provider.
15 SFDR_ERR_NODEV	No devices are present.	None
16 SFDR_ERR_INVAL	Invalid argument and/or operation	None
17 SFDR_ERR_STATE	Invalid board state (transition)	None
18 SFDR_ERR_PROBE	Failed to probe OBP nodes for a board.	None
19 SFDR_ERR_DEPROBE	Failed to deprobe OBP nodes for a board.	None
20 SFDR_ERR_HW_INTERCONNECT	Interconnect hardware failed.	None
21 SFDR_ERR_OFFLINE	Failed to place a CPU offline.	None
22 SFDR_ERR_ONLINE	Failed to bring a CPU online.	None
23 SFDR_ERR_CPUSTART	Failed to start a CPU.	None
24 SFDR_ERR_CPUSTOP	Failed to stop a CPU.	None
25 SFDR_ERR_JUGGLE_ BOOTPROC	Failed to move the clock-signal CPU.	None
26 SFDR_ERR_CANCEL	Could not cancel a RELEASE operation.	Retry the Abort Detach operation after the Drain operation is complete.

DR General Domain Failures

The following table contains a list of the general failure error messages that are sent to the system logs and/or to the SSP applications.

TABLE 2-5 DR General Domain Failure Error Messages

Error Message	Probable Cause	Suggested Action
`DR Error: Cannot fork() process . . .` *errno_description*	The DR daemon could not fork off a process for the command to run in. A message in the form "running command" appears in the system logs prior to this error message, or any other error message about failed commands.	The *errno_description* offers hints on how to fix the command that you want to run. Also check the man page for the command. It may have an explanation of the error.
`DR Error:` *command* `has continued`	While the DR daemon was running external commands, one of the commands failed or exited abnormally. The DR feature executes external commands (for example, `drvconf`) to configure the software subsystems.	Run the program manually on the domain. If the command fails again, refer to the man page for the command. It may have an explanation of the error.
`DR Error:` *command* `stopped by signal` *signal_number*	While the DR daemon was running external commands, one of the commands failed or exited abnormally. The DR feature executes external commands (for example, `drvconf`) to configure the software subsystems.	Run the program manually on the domain. If the command fails again, refer to the man page for the command. It may have an explanation of the error.
`DR Error:` *command* `terminated due to signal` *signal_number*	While the DR daemon was running external commands, one of the commands failed or exited abnormally. The DR feature executes external commands (for example, drvconf) to configure the software subsystems.	Run the program manually on the domain. If the command fails again, refer to the man page for the command. It may have an explanation of the error.
`DR Error:` *command* `terminated due to signal` *signal_number*`. Core dumped.`	While the DR daemon was running external commands, one of the commands failed or exited abnormally. The DR feature executes external commands (for example, `drvconf`) to configure the software subsystems.	Run the program manually on the domain. If the command fails again, refer to the man page for the command. It may have an explanation of the error.

TABLE 2-5 DR General Domain Failure Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: dr_issue_ioctl: failed closing driver . . .` *errno_description*	The DR daemon encountered a failure while it tried to close a DR driver entry point. A more detailed explanation of this failure accompanies the error message.	Use the `close`(2) man page and the *errno_description* to determine what caused this error and how to solve it.
`Cannot exec command (errno =` *errno_value*`).`	The DR daemon could not execute the external command. A more detailed explanation of this failure accompanies the error message.	Check the system logs to determine which command failed. See the `exec`(2) man page for more details about the specified *errno_value.* Use this information to solve the error.
`dr_get_sysbrd_info: NULL parameter`	An invalid pointer was given to the DR daemon during a query of the slot-to-memory address mapping. Either an RPC gave an incorrect value, or the DR daemon called itself with an invalid parameter.	You should gather as much information about this problem as possible from the system logs so that you can determine the cause of the failure. Try stopping and starting the DR daemon and the SSP application. If this error persists, report it to your Sun service representative.
`update_cpu_info: bad board number`	A problem within the DR daemon occured, causing it to call its internal routines with incorrect values.	You should gather as much information about this problem as possible from the system logs so that you can determine the cause of the failure. You should also report this problem, and if it persists, you may have to stop and restart the daemon.
`WARNING: Failed to update board` *board_number*`'s modification time [non-fatal].`	Updating the board modification time has failed. After a board has been modified (for example, memory or CPUs added), it is probed or deprobed by OBP so that OBP can inform other programs of the change. Then, the modification time is updated.	This error is non-fatal.

DR Domain Exploration Error Messages

The following table contains the system exploration error messages that are sent to the system logs and/or to the SSP applications.

TABLE 2-6 DR Domain Exploration Error Messages

Error Message	Probable Cause	Suggested Action
`Cannot open /etc/driver_aliases; dr_daemon may not operate correctly without driver alias mappings . . .` *errno_description*	The DR daemon made an incorrect decision about the detachability and usage of devices in the domain. It is a non-fatal error.	Analyze what caused this error by using the *errno_description*, and try to correct the error. Look for incorrect file permissions or some kind of resource limit that has been encountered. After you correct the error, you must stop the DR daemon, then restart it so that it attempts to read the driver alias mappings again.
`Cannot open mnttab (errno=`*errno_value*`)`	The DR daemon does not allow a detachability test to pass if the `mnttab` file cannot be opened and examined to determine which file systems are mounted. If the test is not stopped, a mounted file system could be detached from the domain.	Analyze the cause of this error by using the *errno_value*, and try to correct the error. The DR deamon may have encountered a resource limit. If so, stop the daemon then restart it. Also, check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon then restart it.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`Cannot open socket (errno=`*errno_value*`)` This error message is sent only to the system logs.	The DR daemon could not open a network device. All network devices are opened to test their usage.	Determine what caused this error by using the *errno_value.* The DR daemon may have encountered a resource limit. If so, stop the daemon then restart it. Also, check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon then restart it. If you cannot recover the domain from this error or if symptoms of a memory leak exist, report this error to your Sun service representative, providing as much information from the system logs as possible.
`get_cpu_bindings: can't access /proc filesystem [non-fatal].`	The `/proc` filesystem cannot be opened. When the DR daemon explores the domain to determine the CPU information for a board, the `/proc` filesystem is examined to determine which PIDs, if any, are bound to the CPUs on the board. Bound processes negatively affect the detachability of a board. A complete detach operation will fail if processes are bound to a CPU.	Check to see why the `/proc` filesystem cannot be accessed. In the domain, process binding and processor set management programs, or processor management programs, can be used to manually determine the CPU information for a board.
`get_mem_config: couldn't determine total system memory size; only 1 board counted [non-fatal].`	When the DR daemon tried to count the amount of total memory, it could report only the amount of memory on the selected board, meaning that the system memory field reported by the `drshow` *board_number* `mem` command is inaccurate. The inaccuracy also negatively affects the eligibility of a board for a Detach operation because if the total memory cannot be calculated, then the effects of removing a board from the domain cannot be calculated as well.	Stop and restart the DR daemon and driver. Report this error, providing as much information from the system logs as possible. A memory leak could also have occured over time. Check the size of the DR daemon by using the `ps`(1) command. The size should be between 300- and 400-Kbytes. If the size is not within this range, stop and start the DR daemon and driver.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`get_net_config_info: interface_name no address (errno=`*errno_value*`)`	The DR daemon encountered a failure while it tried to obtain information about a network interface that was configured by using the `ifconfig`(1M) command.	Determine what caused this error by using the *errno_value*, then correct the error.
`getmntent returned error`	The `getmntent`(3c) system call failed because the mount-point entries could not be properly examined. If the mount-point entries cannot be properly examined, a mounted file system could be detached from the domain.	Analyze the `mnttab` file for possible corruption. If any exists, correct it. Also, the DR daemon may have encountered a resource limit. If so, stop the daemon then restart it. Finally, check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon then restart it. If you cannot recover the domain from this error or symptoms of a memory leak exist, report this error to your Sun service representative, providing as much information from the system logs as possible.
`Host addr for` *interface_name* `not found (h_errno=`*errno_value*`)`	The file that is needed to test each active network device may not exist, or it may be corrupted. While the network devices are examined, each active network device is tested to determine if it is the primary network interface for the domain. The DR daemon will not allow the detachability test to pass if it cannot determine which active network device is the primary network interface for the domain.	Use the *errno_value* to determine if the file exists or if it is corrupted, and correct the error as necessary. The file is named `/etc/hostname.`*interface_name*, where *interface_name* is the interface named in the error message.
`Host address field for` *interface_name* `is null!!`	The IP address for the primary interface (*interface_name*) is not set properly. While the network devices are examined, each active network device is tested to determine if it is the primary network interface for the domain. The DR daemon will not allow the detachability test to pass if it cannot determine which active network device is the primary network interface for the domain.	Reconfigure the network setup for the domain. You may need to reboot the domain to configure network devices.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`Host address for` *interface_name* `must be internet address.`	The file that is needed to test each active network device may have a corrupted value or an incorrect network address. While the network devices are examined, each active network device is tested to determine if it is the primary network interface for the domain. The DR daemon will not allow the detachability test to pass if it cannot determine which active network device is the primary network interface for the domain.	Make sure that the hostname file for the primary network interface contains an IP address in the proper form (that is, *xxx.xxx.xxx.xxx*). The file is named `/etc/hostname.`*interface_name*, where *interface_name* is the interface named in the error message.
`I/O bus device tree not built.`	This error message continues added information about the `DR Error: device tree not built` error message, in which the `libdevinfo` API failed to build the device tree for the system board.	See the `DR Error: device tree not built` error message.
`minor_walk: failed to build net leaf.`	This error message continues added information about the `DR Error: device tree not built` error message, in which the `libdevinfo` API failed to build the device tree for the system board. This message indicates that the `libdevinfo` API at least started to look at the minor devices for a network leaf node.	See the `DR Error: device tree not built` error message.
`minor_walk: failed to build non-net leaf.`	This error message continues added information about the `device tree not built` error message, indicating that the `libdevinfo` API at least started to look at the minor devices for a non-network leaf node.	See the `DR Error: I/O bus device tree not built` error message.
`Partition` *partition_name* `does not have parent.`	The device tree is in error because it includes a disk partition that does not have a parent device, such as the disk to which the partition belongs.	A device could be bad, or a reboot may be necessary. If this error continues to appear, report the error to your Sun service representative, providing as much information from the system logs as possible.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`Recursive symlink found '`*symbolic_link_name*`'. Please remove it.`	The DR daemon found a symbolic link as it walked the `/dev` and `/devices` directories. Some symbolic links create a recursive loop. The DR daemon will not allow the detachability test to pass if it finds a symbolic link in one of these directories.	Remove the symbolic link so that the test can be retried.
`swapctl SC_GETNSWP failed (errno=`*errno_value*`)`	The `swapctl`(2) system call failed. This system call is used to determine which disk partitions are in use as swap space. The DR daemon will not allow the detachability test to pass if the use of swap partitions cannot be determined.	Analyze what caused this error by using the *errno_value*, and try to correct it. Use the `swapctl`(2) man page and the *errno_value* to determine why the command failed. The DR daemon may have encountered a resource limit. If so, stop the daemon then restart it. Also, check the size of the DR daemon. it should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon then restart it. If you cannot recover the domain from this error or if symptoms of a memory leak exist, report this error to your Sun service representative, providing as much information from the system logs as possible.
`Unable to find cwd` *errno_value*	The DR daemon could not save the current working directory. The daemon switches into the `/dev` and `/devices` directories to produce the *real* pathnames that correspond to device drivers.	Determine what caused this error by using the `getcmd`(3c) man page and the *errno_value*, then correct the error.
`Unable to find the cwd` *errno_value*	The DR daemon could not determine the name of the driver directory. The daemon switches into the `/dev` and `/devices` directories to produce the *real* pathnames that correspond to device drivers.	Determine what caused this error by using the `getcmd`(3c) man page and the *errno_value*, then correct the error.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`Unable to get swap entries (errno=`*errno_value*`)`	The `swapctl`(2) system call failed. This system called is used to determine which disk partitions are in use as swap space. The DR daemon will not allow the detachability test to pass if swap partitions cannot be determined.	Analyze what caused this error by using the `swapctl`(2) man page and the *errno_value*, and try to correct it. The DR daemon may have encountered a resource limit. If so, stop the daemon then restart it. Also, check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon then restart it. If you cannot recover from this error or if symptoms of a memory leak exist, report this error to your Sun service representative, providing as much information from the system logs as possible.
`Unable to lstat` *devlink_file errno_value*	The `lstat`(2) system call failed when it encountered the *devlink_file*, where *devlink* is the name of the symbolic link in the `/dev` directory.	Determine what caused this error by using the `lstat`(2) man page and the *errno_value*. The DR daemon may have encountered a resource limit. If so, stop the daemon then restart it. Also, check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon then restart it. If you cannot recover the domain from this error or if symptoms of a memory leak exist, report this error to your Sun service representative, providing as much information from the system logs as possible.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`Unable to open` *hostname_file* `(errno=`*errno_value*`)`	The information that is needed to test each active network device could not be acquired. While the network devices are examined, each active network device is tested to determine if it is the primary network interface for the domain. The DR daemon will not allow the detachability test to pass if it cannot determine which active network device is the primary network interface for the domain.	Analyze what caused this error by using the `open`(2) man page and the *errno_value*, and try to correct it. Look for incorrect file permissions or non-existent files. The *hostname_file* value consists of a file named `/etc/hostname.`*ifname*, where *ifname* is a device name, such as `hme0` or `le0`.
`Unable to read host name from` *hostname_file*	The file that is needed to test each active network device could not be read. While the network devices are examined, each active network device is tested to determine if it is the primary network interface for the domain. The DR daemon will not allow the detachability test to pass if it cannot determine which active network device is the primary network interface for the domain.	Ensure that the file has the correct permissions and that it has not been corrupted.
`Unable to readlink` *devlink_file errno_value*	The `readlink`(2) system call failed when it encountered the *devlink_file*, where *devlink* is the name of the symbolic link in the `/dev` directory.	Determine what caused this error by using the `readlink`(2) man page and the *errno_value*. The DR daemon may have encountered a resource limit. If so, stop the daemon, then restart it. Also, check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon, then restart it. If you cannot recover the domain from this error or if symptoms of a memory leak exist, report this error to your Sun service representative, providing as much information from the system logs as possible.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`Unable to restore cwd` *errno_value*	The DR daemon was unable to change back to the original directory after it changed into `/dev` or `/devices` directory. The DR daemon changes into the `/dev` and `/devices` directories to explore the relationships of the device driver with other drivers.	This error should not pose a problem for the domain, but you should determine what caused the error by using the *errno_value*.
`Unable to set cwd` *errno_value*	The DR daemon could not change into the `/dev` and `/devices` directories. The daemon switches into these directories to produce the *real* pathnames that correspond to device drivers.	Determine what caused this error by using the `chdir`(2) man page and the *errno_value*, then correct the error.
`unknown node type`	The device tree was built incorrectly. Several functions create the device tree for a system board by using the `libdevinfo` API, and searches the `/dev` and `/devices` directories. After the tree is constructed, it is passed on to the `rpc_info()` function, which builds the tree, performs some verifications, then translates the tree into a structure that can be returned from an RPC.	Check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon, then restart it. If you cannot recover the domain from this error, report this error to your Sun service representative, providing as much information form the system logs as possible.
`utssys failed (`*errno_value*`) for` *mount_point*	The `utssys`() system call failed. This system call is used to determine the usage count for a mounted partition. The DR daemon will not allow the detachability test to pass if the usage count cannot be determined.	Analyze what caused this error by using the *errno_value*, and try to correct it. The DR daemon may have encountered a resource limit. If so, stop the daemon then restart it. Also, check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon then restart it. If you cannot recover the domain from this error or if symptoms of a memory leak exist, report this error to your Sun service representative, providing as much information from the system logs as possible.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`walk_dir: dirlist buffer overflow.`	As it walked the `/dev` and `/devices` directories, the DR daemon encountered too many directories, causing a buffer overflow, . If this message occurs, detection of or protection against recursive symbolic links is disabled.	Check the `/dev` and `/devices` directories for recursive symbolic links. Remove any recursive symbolic links that you find.
`walk_dir: tpath buffer overflow.` *target_path*, *device_name*	The DR daemon cannot add another directory to the *target_path*. The daemon walks the `/dev` and `/devices` directories to discover device name links so that it can add them to the target path. If the daemon encounters this limit, it cannot explore any more directories because the buffer is full. If the daemon stops it search, some of the devices will not appear in the views (DR daemon and SSP) of the domain device tree. You may also see improper autoswitching of AP devices if this error occurs.	Devices that are not added to the target path must be manually unconfigured and switched to other boards in the domain. You may also need to stop any daemon that is keeping a device open.
`WARNING: cannot check for cvc/ssp interface.`	The information that is needed to test each active network device could not be acquired. While the network devices are examined, each active network device is tested to determine if it corresponds to the SSP network interface for the domain. The DR daemon will not allow the detachability test to pass if it cannot determine the SSP network interface. If the network loses the SSP network interface during a detach operation, DR operations are disabled in the domain, and `netcon`(1M) sessions are disabled.	Switch the suspected interface to a redundant network connection on another board. You may have to reboot the domain to recover from this error.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`WARNING: Cannot check for primary interface`	The information that is needed to test each active network device could not be acquired. While the network devices are examined, each active network device is tested to determine if it is the primary network interface for the domain. The DR daemon will not allow the detachability test to pass if it cannot determine which active network device is the primary network interface for the domain.	Determine which board hosts the primary network interface and re-attach the board to the domain. Or, switch the interface to a redundant network connection on another board in the domain. You may have to reboot the domain to recover from this error.
`WARNING: Cannot determine if` *interface_name_instance* `is cvc/ssp interface. SIOCGIFNETMASK errno=`*errno_value*	The DR daemon failed to obtain the necessary information to test an active network interface to determine if it is the SSP connection. While the network devices are examined, each active network device is tested to determine if it is the SSP connection for the domain. The DR daemon will not allow the detachability test to pass if it cannot determine which active network device is the SSP connection for the domain. If the network loses the SSP connection during a DR Detach operation, DR operations and `netcon`(1M) sessions are disabled.	Switch the network interface (*interface_name*) to another board. If you cannot correct this error, you may have to reboot the domain.
`WARNING: cannot stat` *device_name* `errno=`*errno_value*	The `stat`(2) system call cannot access the `/dev` entry point for a device in the system device tree.	Use the `stat`(2) man page and the *errno_value* why the file device_name could not be accessed.
`DR Error: Bad page size from sysconf . . .` *errno_description*	The `sysconf`(3c) system call returned an incorrect value for the system page size, meaning that the system call is broken or that it is not providing a required feature. This error may also explain why queries for memory information or detachability tests are failing due to incorrect reporting of memory sizes.	Use the `sysconf`(3c) man page and the *errno_value* to determine the cause of the error.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: device tree not built.`	The `libdevinfo` API failed to build the device tree for the system board. More detailed information about this error accompanies the error message.	Make sure that the correct version of the `libdevinfo` is included on the domain and that a version mismatch does not exist between the DR daemon's libraries, the operating environment on the domain, or the DR daemon itself. If no cause can be found, report this error to your Sun service representative.
`DR Error: dr_get_partn_cpus: cannot get cpu's partition . . .` *errno_description*	The DR daemon tried to use the `pset_assign`(2) function, but the function failed. The DR daemon uses this function to obtain the processor set and partitioning information, which it sends to the CPU Configuration window.	Use the `pset_assign`(2) man page and the *errno_description* to determine and correct the cause of this error.
`DR Error: dr_get_partn_cpus: failed to get cpu partition info . . .` *errno_description*	The DR daemon tried to use the `pset_info`(2) function, but the function failed. The DR daemon uses this function to obtain the processor set and partitioning information, which it sends to the CPU Configuration window.	Use the `pset_info`(2) man page and the *errno_description* to determine and correct the causse of this error.
`DR Error: dr_page_to_kb: page size smaller than a KB`	A math error occured, or an incorrect memory value was used in a memory calculation.	Report this error to your Sun service representative.
`DR Error: get_board_config: invalid board state`	A communication protocol has been breached over the eligibility of a board. To the SSP, the board is part of the domain. However, to the DR daemon and driver, the board is not part of the domain.	Stop and start the DR application, then retry the operation. If the error persists, use the `kill`(1M) command to stop the DR daemon, then start the DR daemon and retry the DR operation.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: get_board_config: invalid flag`	The SSP passed an invalid or unsupported flag to the DR daemon when the daemon tried to ascertain the configuration of a board.	Make sure that the version numbers match for the SSP and the DR daemon. Also, check the size of the daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon has grown far beyond the above memory sizes, then an internal error may have occur within it. You may have to stop and restart the DR daemon to recover from this error.
`DR Error: libdevinfo failed.`	The initial routine used to open the `libdevinfo` API failed, so the DR daemon could not explore the device tree for that board. The `libdevinfo` API builds a tree of dev-info nodes for a board as part of the DR daemon's exploration of the domain devices and their usage. The tree is required by AP and DR operations to test the detachability of a board I/O devices. It is also used to inform the user of what devices are on what system boards.	Make sure that the correct version of the `libdevinfo` is included on the domain and that a version mismatch does not exist between the DR daemon's libraries, the operating environment on the domain, or the DR daemon itself. If no cause can be found, report this error to your Sun service provider.
`get_cpu_info: cpu state info is incomplete [non-fatal].`	The DR daemon could not gather the states of the CPUs (either online or offline). Therefore, the information about each CPU in the CPU Configuration window will not be accurate.	None

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: build_rpc_info: bad slot number`	The device tree was built incorrectly. Several functions create the device tree for a system board by searching through the `/dev` and `/devices` directories and by using the `libdevinfo` API. After the tree is built, it is passed to the `build_rpc_info()` function that performs some verification of the tree as it translates the DR daemon device tree into a structure that can be returned from an RPC.	Check the size of the DR daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon has grown far beyond the above memory sizes, then an internal error may have occur within it. You may have to stop and restart the DR daemon to resolve this error. Report this error to your Sun service representative, providing as much information from the system logs as possible.
`DR Error: build_rpc_info: device address format error`	The device tree was built incorrectly. Several functions create the device tree for a system board by searching through the `/dev` and `/devices` directories and by using the `libdevinfo` API. After the tree is built, it is passed to the `build_rpc_info()` function that performs some verification of the tree as it translates the DR daemon device tree into a structure that can be returned from an RPC.	Check the size of the DR daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon has grown far beyond the above memory sizes, then an internal error may have occur within it. You may have to stop and restart the DR daemon to resolve this error. Report this error to your Sun service representative, providing as much information from the system logs as possible.
`DR Error: build_rpc_info: I/O bus node address format error`	The device tree was built incorrectly. Several functions create the device tree for a system board by searching through the `/dev` and `/devices` directories and by using the `libdevinfo` API. After the tree is built, it is passed to the `build_rpc_info()` function that performs some verification of the tree as it translates the DR daemon device tree into a structure that can be returned from an RPC.	Check the size of the DR daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon has grown far beyond the above memory sizes, then an internal error may have occur within it. You may have to stop and restart the DR daemon to resolve this error. Report this error to your Sun service representative, providing as much information from the system logs as possible.

TABLE 2-6 DR Domain Exploration Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: build_rpc_info: psycho number out of range`	The device tree was built incorrectly. Several functions create the device tree for a system board by searching through the `/dev` and `/devices` directories and by using the `libdevinfo` API. After the tree is built, it is passed to the `build_rpc_info()` function that performs some verification of the tree as it translates the DR daemon device tree into a structure that can be returned from an RPC.	Check the size of the DR daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon has grown far beyond the above memory sizes, then an internal error may have occur within it. You may have to stop and restart the DR daemon to resolve this error. Report this error to your Sun service representative, providing as much information from the system logs as possible.
`DR Error: build_rpc_info: sysio number out of range`	The device tree was built incorrectly. Several functions create the device tree for a system board by searching through the `/dev` and `/devices` directories and by using the `libdevinfo` API. After the tree is built, it is passed to the `build_rpc_info()` function that performs some verification of the tree as it translates the DR daemon device tree into a structure that can be returned from an RPC.	Check the size of the DR daemon by using the `ps`(1) command. Normally, the daemon uses about 300- to 400-Kbytes of memory. If the daemon has grown far beyond the above memory sizes, then an internal error may have occur within it. You may have to stop and restart the DR daemon to resolve this error. Report this error to your Sun service representative, providing as much information from the system logs as possible.

OpenBoot PROM Error Messages

The following table contains the list of OpenBoot™ PROM (OBP) error messages that are sent to the system logs and/or to the SSP applications.

TABLE 2-7 OBP Error Messages

Error Message	Probable Cause	Suggested Action
`cpu unit without upa-portid [non-fatal]`	This message indicates that corrupted or incorrect values were found in the OBP structures, meaning that the information in the OBP Configuration window will not be correct.	This is a non-fatal error. If this error persists, reboot the domain. If the error persists after the reboot, report it to your Sun service representative, providing as much information about the error as possible.
`OBP_info: bad child units [non-fatal]`	This message indicates that corrupted or incorrect values were found in the OBP structures, meaning that the information in the OBP Configuration window will not be correct.	This is a non-fatal error. If this error persists, reboot the domain. If the error persists after the reboot, report it to your Sun service representative, providing as much information about the error as possible.
`obp_info: bad slot number [non-fatal]`	This message indicates that corrupted or incorrect values were found in the OBP structures, meaning that the information in the OBP Configuration window will not be correct.	This is a non-fatal error. If this error persists, reboot the domain. If the error persists after the reboot, report it to your Sun service representative, providing as much information about the error as possible.
`obp_info: missing sbus name [non-fatal]`	This message indicates that corrupted or incorrect values were found in the OBP structures, meaning that the information in the OBP Configuration window will not be correct.	This is a non-fatal error. If this error persists, reboot the domain. If the error persists after the reboot, report it to your Sun service representative, providing as much information about the error as possible.

TABLE 2-7 OBP Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`obp_info: missing slot number [non-fatal]`	This message indicates that corrupted or incorrect values were found in the OBP structures, meaning that the information in the OBP Configuration window will not be correct.	This is a non-fatal error. If this error persists, reboot the domain. If the error persists after the reboot, report it to your Sun service representative, providing as much information about the error as possible.
`sbus node without upa-portid [non-fatal]`	This message indicates that corrupted or incorrect values were found in the OBP structures, meaning that the information in the OBP Configuration window will not be correct.	This is a non-fatal error. If this error persists, reboot the domain. If the error persists after the reboot, report it to your Sun service representative, providing as much information about the error as possible.
`sysio_num out of range [non-fatal]`	This message indicates that corrupted or incorrect values were found in the OBP structures, meaning that the information in the OBP Configuration window will not be correct.	This is a non-fatal error. If this error persists, reboot the domain. If the error persists after the reboot, report it to your Sun service representative, providing as much information about the error as possible.
`DR Error: cannot open /dev/openprom. . .` *errno_description*	The DR daemon could not open the entry point for the domain OBP information, meaning that no information will appear in the OBP Configuration window. This error is not fatal.	Determine what caused this error by using the open(2) man page and the *errno_description*. The DR daemon may have encountered a resource limit. If so, stop the daemon then restart it. Also, check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon then restart it. If you cannot recover the domain from this error or symptoms of a memory leak exist, report this error to your Sun service representative, providing as much information from the system logs as possible.

TABLE 2-7 OBP Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: close error on /dev/openprom`	The DR daemon failed to close the entry point for the OBP driver.	Determine what caused this error by using the error messages that preceeded this error message. Fix the error if possible.
`DR Error: dev/openprom busy. Cannot open.`	The entry point for the domain OBP information was busy, meaning that no information will appear in the OBP Configuration window. This error is non-fatal.	Retry the operation. Check for process that may be keeping the entry point open by using the `ps`(1M) command. Stop any processes that are keeping the entry point open.
`DR Error: get_obp_board_config: invalid board state`	Communication protocol was breached over the eligibility of a board when the SSP application tried to query the OBP information for a board. To the SSP, the board is part of the domain, so the SSP attempts to drain the board resources. However, to the DR driver and daemon, the board is not part of the domain.	None
`DR Error: OBP config: too many CPUs`	The DR daemon found too many CPUs attributed to a system board in the OBP structures. To OBP, the board has more CPUs than it could possibly have (for instance, five or more).	Ensure that OBP is operating properly. If it is not, reboot the domain.
`DR Error: OPROMCHILD. . .` *errno_description*	An `ioctr()` performed on the OBP driver entry point failed, specifically the `ioctr()` used to walk the child OBP node in the device tree, meaning that the information in the OBP Configuration window will not be complete.	Determine what caused this error by using the *errno_value* or the *errno_description* that accompanies this error message. Fix the error is possible.
`DR Error: OPROMGETPROP. . .` *errno_description*	An `ioctl()` performed on the OBP driver entry point failed, specifically the `ioctl()` used to acquire the OBP properties, meaning that the information in the OBP Configuration window will be incomplete.	Determine what caused this error by using the `ioctl`(2) man page and the *errno_description* that accompanies this error message. Fix the error if possible.

TABLE 2-7 OBP Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`DR Error: OPROMNEXT. . .` *errno_description*	An `ioctl()` performed on the OBP driver entry point failed, specifically the `ioctr()` used to walk to the next OBP node in the device tree, meaning that the information in the OBP Configuration window will not be complete.	Determine what caused this error by using the `ioctl`(2) man page and the *errno_description* that accompanies this error message. Fix the error if possible.
`DR Error: System architecture does not support this option of this command.`	An unsupported option was given to the DR daemon as it walked the OBP tree for the domain, meaning that part of the information in the OBP Configuration window will be incorrect. This error is non-fatal.	None

Unsafe-Device Query Failures

The following table contains the list of unsafe-device query failure error messages that are sent to the system logs and/or to the SSP applications.

TABLE 2-8 Unsafe-Device Query Error Messages

Error Message	Probable Cause	Suggested Action
`unsafe_devices: couldn't determine name of unsafe device` *major_number*	The mechanism that the DR daemon uses to combine a driver name with a major number failed so that no name could be discovered. If this failure occurs, the DR daemon constructs a string for the device, marking it as "(unknown, *major_number*)".	This message notifies the user that the DR daemon was unable to find the name of one of the devices, but it does not constitute a correctable error. The daemon can use the major number to identify the drive.

TABLE 2-8 Unsafe-Device Query Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`WARNING: board` *board_number* `not checked for unsafe devices.`	While the DR daemon was examining the system boards for unsafe devices, the daemon encountered a failure that prevented it from examining one of the system boards (*board_number*). This error message may be indicative of a more serious problem.	You may have to stop and restart the DR daemon to recover the domain from this error. Check the size of the DR daemon. If should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon, then restart it. If you cannot recover the domain from this error, you should report this error to your Sun service representative, providing as much information from the system logs as possible.
`DR Error: unsafe_devices: libdevinfo failed.`	The DR daemon cannot determine the names of unsafe major devices because it cannot use the `libdevinfo` API. This API must be used to search the device tree for the names of all of the unsafe major devices.	Make sure that the domain contains the correct version of the `libdevinfo` API and that the domain does not contain version mismatches between any of the DR daemon's libraries, the operating environment on the domain, or the daemon itself. If you cannot determine the cause of this error, report it to your Sun service representative, providing as much information from the system logs as possible.
`DR Error: create_ctlr_array: count mismatch [internal error]`	Communication protocol was breached over the existence of AP controllers. To the AP librarian, the domain has a certain number of AP controllers. However, to the DR daemon, the domain has a different number of AP controllers.	Check to determine the correct amount of AP controllers in the domain, and correct the error. Also, check the size of the DR daemon. It should be between 300- and 400-Kbytes. If it is not within this range, stop the daemon then restart it.

AP-Related Error Messages

The following table contains the list of Alternate Pathing error message that are sent to the system logs and/or to the SSP applications.

TABLE 2-9 AP-Related Error Messages

Error Message	Probable Cause	Suggested Action
`add_net_ap_info: multiple AP aliases ignored`	An AP device has multiple AP aliases. Only one alias is used. The other aliases were ignored. This is not an error.	If this error persists, remove all but one of the AP aliases.
`AP daemon call failed:` *error_message* `*OR* error =` *error_number*	An attempt to notify and/or query the AP librarian failed.	A descriptive error message may be available to provide specific details about this failure, or an error number may be available. Also, check the `ap_daemon`(1M) man page for more details about this error.
`AP daemon comm init failed:` *error_message* `*OR* error =` *error_number*	The DR daemon encountered a failure when it tried to establish a channel of communication with the AP librarian.	A descriptive error message may be available to provide specific details about this failure, or an error number may be available. Also, check the `ap_daemon`(1M) man page for more details about this error.
`AP daemon query failed:` *error_message* `*OR* error =` *error_number*	The DR daemon could not successfully query the AP librarian on the usage of a specific I/O controller.	A descriptive error message may be available to provide specific details about this failure, or an error number may be available. Also, check the `ap_daemon`(1M) man page for more details about this error.
`AP daemon query failed: length mismatch`	The DR daemon queried the AP librarian about the usage of a specific I/O controller, but the response was incorrect.	A descriptive error message may be available to provide specific details about this failure, or an error number may be available. Also, check the `ap_daemon`(1M) man page for more details about this error.

TABLE 2-9 AP-Related Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`Cannot find physical device for` *AP_alias* This error message is sent only to the system logs.	The physical device name that corresponds with the AP alias could not be found. AP may be confused about the device name, or the `/dev` and `/devices` directories are incomplete.	Make sure that AP works properly. Check to see if all of the device entries are present in the `/dev` and `/devices` directories. If they are not present, add them the to the appropriate directories.
`create_ap_net_leaf: interface instance not found`	The DR daemon tries to match the AP metanetwork interfaces with the physical device they represent. This error indicates that the DR daemon could not successfully match a network interface with the physical device it represents for this board.	Make sure that AP works properly if you observe abnormal behavior regarding the availability of devices during and after DR operations. If this error persists, report it to your Sun service representative with as much information from the system logs as possible.
`dr_ap_notify: unknown state` *state_number*	The DR daemon called one of its internal functions with a bad value. However, this error is indicative of a more serious problem.	Report this error to your Sun service representative with as much information as possible from the system logs.
`dr_daemon operating in NO AP interaction mode`	The AP software is not working, or it is not installed. This message means that the DR daemon will not notify AP about attach and detach operations.	Ignore this error if you do not have AP installed. If it is installed, make sure that it is properly installed and that the AP software version is compatible with the version of the DR daemon that is running in the domain.
`init_ap_rpc: Unable to get hostname`	The `uname`(2) system call returned a null hostname. Consequently, the DR daemon could not establish a connection to the AP librarian.	None