# Sun Enterprise™ 10000 Domain Error Messages

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

Part No. 806-4120-10
February 2000, Revision 01

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

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X / Open Company, Ltd. For Netscape Communicator™, the following notice applies: (c) Copyright 1995 Netscape Communications Corporation. All rights reserved.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, Sun Enterprise, OpenBoot, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

**RESTRICTED RIGHTS:** Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6 / 87) and FAR 52.227-19(6 / 87), or DFAR 252.227-7015(b)(6 / 95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X / Open Company, Ltd. La notice suivante est applicable à Netscape Communicator™: (c) Copyright 1995 Netscape Communications Corporation. Tous droits réservés.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, Sun Enterprise, OpenBoot, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

---

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 DR and IDN error messages.

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

Chapter 3 contains the IDN 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.<br>Use `ls -a` to list all files.<br>`% You have mail.` |
| **`AaBbCc123`** | What you type, when contrasted with on-screen computer output | `% su`<br>`Password:` |
| *AaBbCc123* | Book titles, new words or terms, words to be emphasized | Read Chapter 6 in the *User's Guide.*<br>These are called *class* options.<br>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 3.3 User Guide* | 806-2887 |
| | *Sun Enterprise 10000 Domain Configuration Guide* | 806-4121 |
| | *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* | 806-4122 |
| Reference | *Sun Enterprise 10000 SSP 3.3 Reference Manual* | 806-2888 |
| | *Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual* | 806-4123 |
| Release Notes | *Solaris 8 5/00 Release Notes Supplement for Sun Hardware* | Printed in Media Kit. |

# 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://www1.fatrain.com/documentation/sun`

## 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`

## 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 (806-4120-10) of your document in the subject line of your email.

CHAPTER 1

# Domain Errors Introduction

This chapter contains an introduction to the DR and IDN 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* or the *Sun Enterprise 10000 InterDomain Networks User Guide* in the SSP 3.3 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 or “IDN Error Messages on the Domain” on page 3 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
- *domain_name_a*, *domain_name_b*, *domain_name_c* for the names of the domains used with the IDN commands
- *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.

“DR Daemon Start-Up Errors” on page 5

“Memory Allocation Error Messages” on page 7

“DR Driver Failures” on page 14

“PSM Error Messages” on page 16

# IDN Error Messages on the Domain

This book contains the IDN messages that occur on the domain from which the IDN command was executed. TABLE 3-1 contains the name, number, and description of IDN `errno` numbers. This table describes some of the message conventions that are used in the message tables.

Tables B-2 through B-5 contain the common InterDomain Network errors, notifications, and panics that can occur on the domain. These tables contain the text of the error, description of the possible cause of the error, and suggested action. For notifications, the list contains the text of the notice and a possible cause of the message.

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`<br><br>`DR: Error: complete_attach: ioctl failed`<br><br>`DR: Error: abort_attach: ioctl failed`<br><br>`DR: Error: get_cpu_info: ioctl failed`<br><br>`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.<br><br>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<br><br>DR: Error: get_mem_drain: ioctl failed<br><br>DR: Error: update_attach: ioctl failed<br><br>DR: Error: ioctl failed, error draining resources<br><br>DR: Error: detach_board: UNCONFIGURE ioctl failed<br><br>DR: Error: detach_board: DISCONNECT ioctl failed<br><br>DR: Error: abort_detach: CANCEL ioctl failed<br><br>DR: Error: abort_detach: CONFIGURE ioctl failed<br><br>DR: Error: get_dr_state: ioctl failed<br><br>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*.<br><br>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*.<br><br>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*`)`<br><br>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. | 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. | 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*<br><br>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 |

CHAPTER 3

# IDN Error Messages, Notifications, and Panics on the Domain

This chapter contains IDN error numbers that appear on the domain and IDN error messages, notifications, and panics that occur on the domain

The destination of these messages depends entirely on the location of the individual error or failure. For some errors, both locations must be used to diagnose the error or failure.

## Domain IDN Messages

IDN messages that occur on the domain are sent to the following locations:

- `netcon`(1M) console window
- `/var/adm/messages`
- `$SSPLOGGER/`*domain_name*`/messages`

The following table contains the IDN error numbers that are specific to the domain.

**TABLE 3-1** IDN Domain-Specific Error Numbers

| Message | Number | Description |
|---|---|---|
| `IDNKERR_DRV_DISABLED` | `0x100` | The IDN driver is disabled. |
| `IDNKERR_DATA_LEN` | `0x101` | The IDN region in the signature block (BBSRAM) is misaligned between the IDN driver and the SSP (`P0` represents the length). |
| `IDNKERR_INFO_FAILED` | `0x102` | SSI_INFO command failed (only in engineering). |

**TABLE 3-1** IDN Domain-Specific Error Numbers *(Continued)*

| Message | Number | Description |
|---|---|---|
| IDNKERR_INVALID_DOMAIN | 0x103 | An invalid domain ID was specified (P0 represents the domain ID, and P1 represents the CPU ID). |
| IDNKERR_INVALID_FORCE | 0x104 | An invalid force option was passed (P0 represents the force option). |
| IDNKERR_INVALID_CMD | 0x105 | An invalid IDN command was requested (P0 represents command). |
| IDNKERR_INVALID_WTIME | 0x106 | An invalid wait time was specified for the IDN operation (P0 represents the wait time). |
| IDNKERR_SMR_CORRUPTED | 0x107 | SMR memory was found corrupted (P0 represents the domain ID, against which the corruption was found). |
| IDNKERR_CPU_CONFIG | 0x108 | Domain ID is not configured properly for an IDN. Each system board that hosts memory must have at least one CPU (P0 represents the domain ID). |
| IDNKERR_HW_ERROR | 0x109 | The domain was unable to properly program the hardware to support an IDN connection to domain ID (P0 represents the domain ID) |
| IDNKERR_SIGBINTR_LOCKED | 0x10a | The signature block interrupt lock on the host is currently locked. |
| IDNKERR_SIGBINTR_BUSY | 0x10b | The signature block interrupt handler thread is currently active. |
| IDNKERR_SIGBINTR_NOTRDY | 0x10c | The signature block interrupt handler thread has not been initialized. |
| IDNKERR_CONFIG_FATAL | 0x10d | An error occurred during the exchange of configuration information with domain ID, specifically it was missing information (P0 represents the domain ID). |
| IDNKERR_CONFIG_MULTIPLE | 0x10e | Multiple conflicts were found between the configuration parameters exchanged during the connection establishment (P0 represents the domain ID). |
| IDNKERR_CONFIG_MTU | 0x10f | The MTU sizes of the domains do not match (P0 represents the domain ID; P1 represents the expected value, and P2 represents the actual value). |
| IDNKERR_CONFIG_BUF | 0x110 | The values of the idn_smr_bufsize variable conflict among the domains (P0 represents the domain ID; P1 presents the expected value, and P2 represents the actual value). |
| IDNKERR_CONFIG_SLAB | 0x111 | The values of the SMR slab sizes conflict among the domains (P0 represents the domain ID; P1 represents the expected value, and P2 represents the actual value). |

**TABLE 3-1** IDN Domain-Specific Error Numbers *(Continued)*

| Message | Number | Description |
|---|---|---|
| IDNKERR_CONFIG_NWR | 0x112 | The values of the idn_nwr_size variables conflict among the domains (P0 represents the domain ID; P1 represents the expected value, and P2 represents the actual value). |
| IDNKERR_CONFIG_NETS | 0x113 | The values of the idn_max_nets variables conflict between domains (P0 represents the domain ID; P1 represents the expected value, and P2 represents the actual value). |
| IDNKERR_CONFIG_MBOX | 0x114 | The values of the idn_mbox_per_nets variables conflict between domains (P0 represents the domain ID; P1 represents the expected value, and P2 represents the actual value). |
| IDNKERR_CONFIG_NMCADR | 0x115 | The number of MCADRs received does not match the number of MCADRs that the domain had reported to exist (P0 represents the domain ID; P1 represents the expected value, and P2 represents the actual value). |
| IDNKERR_CONFIG_MCADR | 0x116 | Received an MCADR for a board that the remote domain did not report to exist (P0 represents the domain ID; P1 represents the expected value, and P2 represents the actual value). |
| IDNKERR_CONFIG_CKSUM | 0x117 | The values of the idn_checksum parameter is not consistent among the domains (P0 represents the domain ID; P1 represents the expected value, and P2 represents the actual value). |
| IDNKERR_CONFIG_SMR | 0x118 | The master domain SMR is too large for the slave domain (P0 represents the domain ID; P1 represents the expected value, and P2 represents the actual value). |

The following table contains the errors, notices, and panics that are specific to the domain.

**TABLE 3-2** IDN Domain-Specific Messages, 100 through 142

| Error | Description | Recovery |
|---|---|---|
| WARNING: IDN: 100: sigblock area misaligned (*bytes*) != exp (*bytes*) | This message indicates a mismatch between the version of the sigblock data structure in the IDN driver and the region of the signature block that is reserved for the IDN. | Unlink the domains, then recreate the IDN. |
| WARNING:IDN:101: not in expected OFFLINE state for DDI_RESUME | The IDN driver is not in the expected state for the DR driver to perform the DDI_RESUME operation. | Unlink the domain, then relink it. |

**TABLE 3-2** IDN Domain-Specific Messages, 100 through 142 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `NOTICE: IDN: 102 driver disabled - check OBP environment (idn-smr-size)` | The IDN driver was not initialized. | Check the OpenBoot PROM (OBP) variable `idn-smr-size` to ensure that it is set properly. Refer to the *Sun Enterprise 10000 Domain Configuration Guide* for more information about this variable. |
| `WARNING: IDN: 103: unable to reference sigblock area` | The `sgnblk_poll_reference()` routine failed to initialize. The IDN driver may have been loaded too early in the boot sequence. | Reload the module after the operating system boots. |
| `WARNING: IDN: 104: cannot suspend while active (state =` *GSTATE*`)` | The IDN driver cannot be suspended while it is in use. | Wait for the driver to complete the current transmission, or unlink the domain from the IDN before you initiate a DR operation. |
| `WARNING: IDN: 105: driver parameter (`*parameter*`) specified (`*number*`) out of range [`*low_value* `-` *high_value*`]` | The value for the specified parameter is outside of the range of values that can be used for the IDN. | Reset the parameter with a value that is within the allowed range. |
| `WARNING: IDN: 106: idn_nwr_size (`*Mbytes*`) > idn_smr_size(`*Mbytes*`) - Limiting to` *number* `MB` | The value of `idn_nwr_size` variable is greater than the value of `idn_smr_size` variable. The value of `idn_nwr_size` cannot be larger than the value of `idn_smr_size`. | The IDN driver reduces the size of the network region (NWR) to the total size of the shared memory region (SMR). If the IDN must have a NWR that is larger than the current size of the SMR, increase the size of the SMR so that the size of the NWR can be increased to the appropriate value. |
| `WARNING: IDN: 107: memory region(`*bytes*`) < slab size(`*bytes*`)` | The value of the `idn_nwr_size` variable is less than the size of one of the slabs within the SMR. | Increase the value of `idn_smr_size` or `idn_nwr_size` to a value that is larger than the smallest buffer size in the SMR. Or, reset other tunables, such as `idn_slab_bufcount`, until the size of each slab within the IDN is smaller than the value of `idn_nwr_size`. |
| `WARNING: IDN: 108: idn_lowat(`*bytes*`) >= idn_hiwat(`*bytes*`)` | The specified values for the low-water and high-water marks for IDN STREAMS are not set properly. | Lower the value of `idn_lowat` or increase the value of `idn_hiwat` as appropriate. Refer to the *Sun Enterprise 10000 Domain Configuration Guide* for more information about the appropriate values for these parameters. |

**TABLE 3-2** IDN Domain-Specific Messages, 100 through 142 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| WARNING: IDN: 109: mailbox area(*bytes*) + slab size(*bytes*) > nwr region(*bytes*) | The specified values for the mailbox area, idn_mbox_size, and the slab size, idn_slab_size, variables are not set properly. | Increase the size of the NWR or the SMR so that the size of the NWR can be increased. You can also decrease the size of other variables, such as the number of mailboxes per channel, idn_mbox_per_channel, or the number of I/O buffers per slab, idn_slab_bufcount. |
| WARNING: IDN: 110: maximum number of slabs(*number*) < minimum required(*number*) | The value of the idn_slab_maxtotal variable is less than the required minimum. The value of this variable is calculated by the IDN driver. The driver returns this warning if the value is less than the minimum value for nominal usage of the IDN. | Lower the value of the idn_slab_mintotal variable, or increase the value of the SMR so that it can hold more slabs. You may need to adjust the value of other driver parameters, such as idn_slab_bufcount, to recover from this error. |
| WARNING: IDN: 111: idn_smr_bufsize(*bytes*) not on a 64 byte boundary | The value of the idn_smr_bufsize variable must be a multiple of 64. | Reset the idn_smr_bufsize variable to a multiple of 64. |
| WARNING: IDN: 112: idn_smr_bufsize(*bytes*) not a power of 2 | The value of the idn_smr_buffsize variable must be set to a value that is a power of two. | Reset the idn_smr_bufsize variable to a power of two. |
| WARNING: IDN: 113: idn_mbox_per_net(*number*) must be an odd number | For proper hashing, the value of the idn_mbox_per_net variable must be an odd number. | Reset the idn_mbox_per_net variable to an odd number. |
| WARNING: IDN: 115: idn_netsvr_wait_min (*seconds*) cannot be greater than idn_netsvr_wait_max (*seconds*) | The minimum wait time for the IDN net server cannot be greater than the maximum wait time. | Decrease the value of idn_netsvr_wait_min, or increase the value of idn_netsvr_wait_max. |
| WARNING: IDN: 116: failed rmalloc(kernelmap, *number* pages) | The operating system failed to allocate pages *number* of the virtual address space for the mapping of the SMR. | Reboot the domain, then attempt to link the domain to the IDN. |

**TABLE 3-2** IDN Domain-Specific Messages, 100 through 142 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 117: IDN not enabled` | The IDN driver failed to initialize the IDN because the IDN was not enabled (that is, `idn-smr-size` is set to zero). | At the OBP prompt, use the `ndd`(1M) command to check the value of the `idn-smr-size` variable. If it is set to zero, reset it to the appropriate value to enable the IDN and to set the size of the SMR. You must reboot the domain if you change the value of this variable. |
| `WARNING: IDN: 118: hardware config not appropriate` | The hardware configuration within the domain that you are trying to link is not appropriate for an IDN, or the IDN driver could not determine the hardware configuration.<br>The hardware configuration within a domain could cause this error if it consists of one of the following configurations:<br>A system board that hosts memory does not host a CPU. System boards in domains must host at least one CPU for the domain to be considered eligible for linking to an IDN. See option `no_non_proc_boards` in the `post`(4) man page on the SSP for information on how to prevent system boards without a CPU from being included in the domain.<br>The shared memory mask on the CIC is not enabled to allow the IDN driver to manipulate the registers. | Ensure that each system board that contains memory has at least one CPU and that SSP 3.2 is running on the SSP. |
| `WARNING: IDN: 119: failed to initialize` *number* `protocol servers` | The IDN driver failed to start up the *number* protocol servers that manage the IDN connections. Typically, a strain on kernel resources causes this error. | Reload the IDN driver. If this error occurs repeatedly, reduce the memory usage by other applications, if possible. |
| `WARNING: IDN: 120: cannot deinit while active (state =` *GSTATE*`)` | The DR operation tried to deinitialize the IDN driver while it was still in use (that is, not offline). | Ensure that the domain is completely unlinked from all IDNs before you initiate a DR operation on the domain. |

**TABLE 3-2** IDN Domain-Specific Messages, 100 through 142 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `IDN: 121: domain` *domain_ID* `(cpu` *CPUID*`, name "`*host*`", state` *DSTATE*`)` | This error displays the IDN state of each domain connection. This error message occurs in conjunction with messages 104 and 120. | See error messages 104 and 120. |
| `WARNING: IDN: 123: unexpected M_DATA packets for q_stream` *VADDR* | The STREAMS read procedure in the IDN driver received an unexpected data packet on the respective stream queue. The packet is discarded. | If this error persists, report the problem to your Sun Microsystems, Inc., service representative. |
| `WARNING: IDN: 124: sigblk for cpuid` *CPUID* `is NULL` | The CPU in question expected to receive messages from the SSP; however, the signature block (BBSRAM) data structure for that CPU is not mapped properly. | Unload, then reload, the IDN driver. If that does not work, unlink the domain, then reboot it. Relink the domain after it has booted successfully. |
| `WARNING: IDN: 125: op (`*IDNOP*`) failed, returning (`*errno*`/`*IDNERR* `[`*EPARAM0*`, `*EPARAM1*`, `*EPARAM2*`])` | An attempted IDN operation (link, unlink, or info) failed. In the message, *errno* equals the error number, *IDNERR* equals the IDN error, and *EPARAM2* represents the parameters that are dependent on this type of IDN error. | Ensure that you used the correct parameters. If not, retry the operation with the correct parameters. |
| `WARNING: IDN: 126: sighandler thread already exists (`*VADDR*`)` | The IDN driver attempted to create an unnecessary, duplicate, sigblock-interrupt-handler thread. | You can safely ignore this message. |
| `WARNING: IDN: 127: cannot change` *parameter* `while IDN connected` | The IDN tunable parameter that you tried to change cannot be changed while the domain is linked to an IDN. | Unlink the domain before you attempt to change the parameter settings for the specified parameter (*parameter*). |
| `WARNING: IDN: 128: cannot change` *parameter* `while DLPI attached` | The IDN tunable parameter that you tried to change cannot be changed while the IDN interface (`idn`*X*) is plumbed under TCP/IP. | Unplumb all of the IDN interfaces that are under TCP/IP control before you attempt to change the parameter settings for the specified parameter (*parameter*). |
| `WARNING: IDN: 129:` *IDNOP* `operation timed out` | An IDN operation (link, unlink, or info) exceeded the specified wait-time before it successfully completed. | Check for AWOL domains in the IDN. If present, unlink the AWOL domain before you attempt to unlink the local domain. You can use the force option, `-f` or `-F`; however, you must use the force option with caution. |

**TABLE 3-2** IDN Domain-Specific Messages, 100 through 142 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 130: IDN DMV handler already initialized` | The IDN driver attempted to initialize the interrupt handler too many times. | Reboot the domain. If this error persists, contact your Sun service representative. |
| `WARNING: IDN: 131; unable to allocate data area for DMV handler` | The IDN driver was unable to allocate a data area for control structures that are used by the DMV handler. | This error typically occurs when memory usage levels are too high. Retry the IDN command when the usage level decreases, or remove some of the system activity that is causing the high usage level. |
| `WARNING: IDN: 132: failed to add IDN DMV handler` | The IDN driver failed to register its internal interrupt handler with the DMV-based kernel subsystem. | Reboot the domain, or unload the IDN driver and retry the operation. If this error persists, report the problem to your Sun service representative. |
| `WARNING: IDN: 133: sigblock event area missing` | The signature block event area that is used between the IDN driver and the SSP may be missing or corrupted. | Reboot the domain, or unload the IDN driver and retry the operation. If this error persists, report the problem to your Sun service representative. |
| `IDN: 134: unable to mark boardset (`*BMASK*`) AWOL` | This error returns a 16-bit board mask that indicates which boards in the domain could not be marked as AWOL so that they could be handled by the SSP. | Unlink the domain(s) in question, then relink them to the domain. You may need to reboot the domain or to unload the IDN driver before you retry the operation. |
| `IDN: 135: idn: kstat_create failed` | Unable to create the `kstat` structures. Thus, global kernel statistics for the IDN are not maintained. | Reboot the domain, and retry the command, if necessary. |
| `WARNING: IDN: 136: "`*property*`" property not found, disabling IDN` | The IDN driver failed to initialize because it did not find the specified OBP property. | Ensure that the SSP 3.2 software is on the SSP. |
| `WARNING: IDN: 137: SMR size is 0, disabling IDN` | The IDN driver failed to initialize because the OBP variable `idn-smr-size` is set to zero (0). | Reset the `idn-smr-size` variable to the appropriate value. The value must be the same for all of the domains in an IDN. Reboot the domain after you reset the value of the variable. |
| `WARNING: IDN: 138: SMR size (`*number*`MB) is too big (max = `*number*`MB), disabling IDN` | The IDN driver failed to initialize because the value of the OBP variable `idn-smr-size` is too large. | Reset the `idn-smr-size` variable to the appropriate value. The value must be the same for all of the domains in an IDN. Reboot the domain after you reset the value of the variable. |

**TABLE 3-2** IDN Domain-Specific Messages, 100 through 142 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 139: OBP region for SMR is 0 length` | The IDN driver failed to initialize because the OBP variable `idn-smr-size` is set to zero (`0`). | Reset the `idn-smr-size` variable to the appropriate value. The value must be the same for all of the domains in an IDN. Reboot the domain after you reset the value of the variable. |
| `WARNING: IDN: 140: OPB region (`*bytes* `B) smaller than requested size (`*bytes* `B)` | The SMR region allocated by the OBP is smaller than the value of the OBP variable `idn-smr-size`. | Ensure that the SSP 3.2 software is running on the SSP, then reset the `idn-smr-size` variable to the appropriate value. The value must be the same for all of the domains in an IDN. Reboot the domain after you reset the value of the variable. |
| `WARNING: IDN: 141: OPB region (`*PADDR*`) not on (`*hex_number*`) boundary` | The SMR region allocated by OBP is not on the appropriate memory address boundary (64-Kbytes). | Ensure that the SSP 3.2 software is running on the SSP, then reboot the domain and retry the operation. |
| `NOTICE: IDN: 142: link (domain `*domain_ID*`, cpu `*CPUID*`) connected` | The domain has been linked with domain *domain_ID* that hosts CPU *CPUID*. | Notification only |

**TABLE 3-3** IDN Domain-Specific Messages, 200 through 242

| Error | Description | Recovery |
|---|---|---|
| `NOTICE: IDN: 200: link (domain `*domain_ID*`, cpu `*CPUID*`) disconnected` | The domain has been unlinked with domain *domain_ID* that hosts CPU *CPUID*. | Notification only |
| `WARNING: IDN: 201: (`*IDNOP*`) invalid cpu-id (`*CPUID*`)` | During the IDN operation, the IDN driver specified an invalid ID number for the CPU. | Retry the IDN operation. |
| `WARNING: IDN: 202: (`*IDNOP*`) invalid time-out value (`*seconds*`)` | An invalid time out value was passed to the IDN operation (*IDNOP* = LINK/UNLINK). | Retry the IDN operation. |
| `WARNING: IDN: 203: (`*IDNOP*`) invalid domain-id (`*domain_ID*`)` | An invalid domain ID was passed to the IDN operation (*IDNOP* = LINK/UNLINK). | Retry the IDN operation. |
| `WARNING: IDN: 204: domain `*domain_ID* `state (`*DSTATE*`) inappropriate` | The domain *domain_ID* was not in the closed state when the IDN operation was executed. | Retry the IDN operation. If this error persists, unlink the domain. |

**TABLE 3-3** IDN Domain-Specific Messages, 200 through 242 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 205: (`*function*`) failed to open-domain (`*domain_ID*`,`*CPUID*`)` | The internal IDN function *function* failed to open an IDN domain control structure for managing an IDN connection with domain *domain_ID* and CPU *CPUID*. | Wait a few minutes for the connection to be resolved. If the connection cannot be resolved, reboot domain *domain_ID*. |
| `WARNING: IDN: 206: cannot link domains with equal votes (L(`*domain_ID*`), R(`*domain_ID*`),`*hex_number*`)` | The local domain, *domain_ID*, has the same vote ticket, *hex_number*, as the domain, *domain_ID*, to which it is trying to link. The vote tickets are determined internally and must be unique. | Reboot the local domain, then retry the operation. If the error occurs again, reboot domain *domain_ID*, then retry the operation. If the error persists, contact your Sun service representative. |
| `WARNING: IDN: 207: local/remote master-id conflict (`*local_domain_ID*`.lmasterid = `*domain_ID_a*`, `*remote_domain_ID*`.rmasterid = `*domain_ID_b*`)` | The local domain, *local_domain_ID*, has a master ID, *domain_ID_a*, that conflicts with the master ID, *domain_ID_b*, of domain *remote_domain_ID*. | Retry the operation. If the error persists, reboot both of the domains, then retry the operation. |
| `WARNING: IDN: 208: idn_select_master: unknown case (`*number*`)` | This is an internal error. During the selection of the master domain, the IDN driver encountered an unexpected case, *number*. | Retry the operation. If the error persists, reboot both of the domains, then retry the operation. |
| `WARNING: IDN: 209: remote domain (id `*domain_ID*`, cpu `*CPUID*`) reporting master (id `*master_domain_ID*`) without cpuid` | During the IDN operation, the remote domain, *domain_ID*, returned the ID of the master to the local domain, but not the CPU ID. | Retry the operation. If this error persists, reboot the remote domain. |
| `WARNING: IDN: 210: failed to init MASTER context` | This is an internal error. The local domain failed to initialize control structures that the domain needs to be a master domain. | Reboot the local domain, then retry the operation. |
| `WARNING: IDN: 211: disconnect domain `*domain_ID*`, unexpected GSTATE (`*GSTATE*`)` | During the disconnect operation for domain *domain_ID*, the local IDN was in an unexpected global state, *GSTATE*. | Reboot the local domain, then retry the operation. |
| `PANIC: IDN: 212: disconnect domain `*domain_ID*`, bad GSTATE (`*GSTATE*`)` | During the disconnect operation for domain *domain_ID*, the local IDN was in an unexpected global state, *GSTATE*. | Reboot the local domain, then retry the operation. |

**TABLE 3-3** IDN Domain-Specific Messages, 200 through 242 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 213: no destination specified (d=`*domain_ID*`, c=`*channel*`, n=0x`*NE*`)` | The local domain attempted to send a data packet to an inappropriate destination, domain ID *domain_ID*, on channel *channel* with the network ID *NE*. | Retry the operation. If this error persists, unlink the local domain, then relink it and retry the operation. |
| `WARNING: IDN: 214: received message (`*MSG*`[0x`*M_number*`]) from self (domid` *domain_ID*`)` | The local domain received an IDN connection protocol message, *MSG*[0x*M_number*], from itself, domain ID *domain_ID*. | Reboot the local domain, then retry the operation. |
| `WARNING: IDN: 215: invalid cookie (`*cookie*`) for message (`*M_number*`) from domain` *domain_ID* | The local domain received an IDN connection protocol message, *M_number*, from domain *domain_ID* with an invalid or stale cookie, *cookie*. | Retry the operation. If this error persists, unlink the local domain and/or remote domain, then relink the local and/or remote domain. |
| `WARNING: IDN: 216: (`*M_number*`)msgtype/ (`*A_number*`)acktype rcvd from domain` *domain_ID* | The local domain received an invalid IDN connection protocol message or acknowledgement (*M_number*/*A_number*) from domain *domain_ID*. | Retry the operation. If this error persists, unlink the local and/or remote domain, then relink the local and/or remote domain. |
| `WARNING: IDN: 217: unknown CFGARG type (`*type*`) from domain` *domain_ID* | The local domain received an unexpected type, *type*, of configuration message from the remote domain, *domain_ID*. | Retry the operation. If this error persists, unlink the local and/or remote domain, then relink the local and/or remote domain. |
| `WARNING: IDN: 218: missing some required config items from domain` *domain_ID* | During the connection operation, the local domain did not receive all of the configuration information it expected from domain *domain_ID*. | Retry the operation. If this error persists, unlink the local and/or remote domain, then relink the local and/or remote domain. |
| `WARNING: IDN: 219: remote domain` *domain_ID* `MTU (`*bytes*`) invalid (local.mtu =` *bytes*`)` | The MTU size, *bytes*, received from domain *domain_ID* is not compatible with the MTU size of the local domain, *bytes*. The value of the IDN tunable parameter `idn_smr_bufsize` must be the same for all of the domains in an IDN. | Reset the value of the `idn_smr_bufsize` parameter on the local domain or the remote domain so that the values are the same. |

**TABLE 3-3** IDN Domain-Specific Messages, 200 through 242 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 220: remote domain` *domain_ID* `BUFSIZE (`*bytes*`) invalid (local.bufsize =` *bytes*`)` | The local domain received an SMR buffer size, *bytes*, from the remote domain that is not compatible with the SMR buffer size of the local domain. The value of the IDN tunable parameter `idn_smr_bufsize` must be the same for all of the domains in an IDN. | Reset the value of the `idn_smr_bufsize` parameter on the local domain or the remote domain so that the values are the same. |
| `WARNING: IDN: 221: remote domain` *domain_ID* `SLABSIZE (`*bytes*`) invalid (local.slabsize =` *bytes*`)` | The local domain received an SMR slab size, *bytes*, from domain *domain_ID* that is not compatible with the SMR slab size of the local domain. The value of the IDN tunable parameters `idn_slab_bufcount` and `idn_smr_bufsize` must be the same for all of the domains in an IDN. | Reset the slab size parameters on the local and/or remote domain. |
| `NOTICE: 222: no IDN linkage found (b=`*BMASK_a*`, i=`*BMASK_b*`) upgrading unlink` *FTYPE* `-> FORCE_HARD` | The SSP requested that the local domain be unlinked from the remote domain with boardmask *FTYPE*; however, the SSP was unable to find a hardware link in the IDN hardware register board mask *BMASK_b*. The specified soft force option, `-f`, was upgraded to the hard force option, `-F`, to unlink the domain. | None |
| `WARNING: IDN: 223: remote domain` *domain_ID* `NWRSIZE (`*Mbytes*`) invalid (local.nwrsize =` *Mbytes*`)` | The local domain received a value of the `idn_nwr_size` variable from the remote domain that does not match the value of `idn_nwr_size` on the local domain. The value of the `idn_nwr_size` variable must be the same for all of the domains in the IDN. | Reset the value of the `idn_nwr_size` variable for the remote domain. |

**TABLE 3-3** IDN Domain-Specific Messages, 200 through 242 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 224: remote domain` *domain_ID* `idn_max_nets (`*number*`) invalid (local.maxnets =` *number*`)` | The local domain received a value of the `idn_max_nets` variable from the remote domain that does not match the value of `idn_max_nets` on the local domain. The value of the `idn_max_nets` variable must be the same for all of the domains in the IDN. | Reset the value of the `idn_max_nets` variable for the remote domain. |
| `WARNING: IDN: 225: remote domain` *domain_ID* `MBOX_PER_NET (`*number*`) invalid (local.mboxpernet =` *number*`)` | The local domain received a value for the `idn_mbox_per_net` variable from the remote domain that does not match the value of `idn_mbox_per_net` on the local domain. The value of the `idn_mbox_per_net` variable must be the same for all of the domains in the IDN. | Reset the value of the `idn_mbox_per_net` variable for the remote domain. |
| `WARNING: IDN: 226: remote domain` *domain_ID* `CHECKSUM flag (`*number*`) mismatches local domain's (`*number*`)` | The local domain received a value for the `idn_checksum` variable from the remote domain that does not match the value of `idn_checksum` on the local domain. The value of the `idn_checksum` variable must be the same for all of the domains in the IDN. | Reset the value of the `idn_checksum` variable for the remote domain. |
| `WARNING: IDN: 227: missing some required config items from domain` *domain_ID* | The local domain did not receive all of the expected configuration information from the remote domain, *domain_ID*. | Retry the link operation. If this error persists, reboot the remote domain, then retry the link operation. |
| `WARNING: IDN: 228: master's SMR (`*bytes*`) larger than local's SMR (`*bytes*`)` | The size the SMR for the master domain is larger than the virtual space the local domain has available for the SMR. The OBP variable `idn-smr-size` must be the same for all of the domains in an IDN. | Reset the size of the `idn-smr-size` variable for the local domain and/or the remote domains so that the size is the same. Reboot the domain(s), then retry the link operation. |

**TABLE 3-3** IDN Domain-Specific Messages, 200 through 242 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 229: remote domain` *domain_ID* `boardset (`*BMASK*`) conflicts with MCADR(board` *number*`) [`*MCADR*`]` | The local domain received conflicting information about the MCADR register from domain *domain_ID*. Board *number* is reported to have a MCADR setting, but it is not present in the physical board set *BMASK* of the remote domain. *MCADR* represents the actual MCADR register value. | Reboot domain *domain_ID*, then retry the operation. |
| `WARNING: IDN: 230: remote domain` *domain_ID* `reported number of MCADRs (`*number*`) mixmatches received (`*number*`)` | The local domain received conflicting information about the MCADRs in domain *domain_ID*. The number of MCADRs specified by domain *number* does not match the number of MCADRs reported by that domain. | Reboot domain *domain_ID*, then retry the operation. |
| `WARNING: IDN: 231: domain` *domain_ID* `boardset (`*BMASK*`) conflicts with existing IDN boardset (`*BMASK*`)` | The set of boards in the board mask *BMASK* for domain *domain_ID* overlap the existing boards *BMASK* in the IDN. | Ensure that the local domain has passed the power-on self-test (POST). Unlink the domain, then relink it. |
| `WARNING: IDN: 232: domain` *domain_ID* `cpuset (`*CPUSET*`) conflicts with existing IDN cpuset (`*CPUSET*`)` | The set of CPUs in the CPU mask *CPUSET* for domain *domain_ID* overlap the existing CPUs *CPUSET* in the IDN. | Ensure that the local domain has passed the power-on self-test (POST). Unlink the domain, then relink it. |
| `WARNING: IDN: 233: domain` *domain_ID* `missing cpu per memory boardset (`*BMASK*`), cpu boardset (`*BMASK*`)` | Each system board in the domain *domain_ID* must have at least one CPU if it hosts memory. Board set *BMASK* represents the board that have memory, and board set *BMASK* represents the boards that have at least one CPU. | Ensure that the domain you want to link has system boards that host at least one CPU on each board that hosts memory. |
| `WARNING: IDN: 234: failed to program hardware for domain` *domain_ID* `(boardset =` *BMASK* | The IDN driver was unable to program the hardware for the local domain to allow shared memory access with domain *domain_ID*, which contains the board set *BMASK*. | Do not execute additional IDN operations. When appropriate, halt domain *domain_ID*, and run a full diagnostic test by using the `hpost`(1M) command. |

**TABLE 3-3** IDN Domain-Specific Messages, 200 through 242 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 235: [`*MBXTYPE*`] mailbox (domain` *domain_ID*`, channel` *channel*`) SMR CORRUPTED - RELINK`<br>`IDN: 235: [`*MBXTYPE*`] expected (cookie` *cookie*`, cksum` *hex_number*`), actual (cookie` *cookie*`, cksum` *hex_number*`)`<br>`IDN: 235: [`*MBXTYPE*`] activeptr (`*VADDR*`), readyptr (`*VADDR*`)` | The local domain detected that the send and/or receive mailbox *MBXTYPE* control area is corrupted for domain *domain_ID*. These messages indicate the expected and actual values of the cookie and checksum control information. Depending on the condition, an additional message is displayed containing additional control information that is used for synchronization during data transmissions (`activeptr` and `readyptr`). | Unlink the master domain, then relink it. If this error persists, dismantle the entire IDN, then reassemble it. |
| `WARNING: IDN: 236: domain (`*host*`) [id` *domain_ID*`] not responding to` *IDN_command* `[#`*number*`]`<br>`WARNING: IDN: 236: domain [id` *domain_ID*`, cpu` *CPUID*`] not responding to` *IDN_command* `[#`*number*`]` | The local domain attempted to connect or disconnect domain *host* or domain ID *domain_ID* with CPU ID *CPUID*; however, the domain is not responding. *number* represents the number of detected AWOL messages. | Unlink domain *host*, then retry the link operation. |
| `WARNING: IDN: 237: invalid number (`*number*`) of protocol servers` | The specified number of IDN protocol servers is invalid. The value of the `idn.conf`(4) tunable parameter `idn_protocol_nservers` must be greater than zero (`0`). | Reset the value of the `idn_protocol_nservers` parameter in the `idn.conf`(4) file to an appropriate number. |
| `WARNING: IDN: 238: kmem_cache_create (jobcache) failed` | The kernel failed to create an internal cache for allocating IDN job control data structures. | Reboot the local domain, then retry the link operation. If this error persists, remove unused software from the domain, and retry the link operation. |
| `WARNING: IDN: 239: invalid cpuid (`*CPUID*`) specified for IDN net` *channel* | An invalid CPU ID, *CPUID*, was specified for the local domain. *channel* represents the network interface for the IDN data server thread. | Retry the operation with a valid CPU ID for the local domain. |

**TABLE 3-3** IDN Domain-Specific Messages, 200 through 242 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 240: (channel` *channel*`) SMR CORRUPTED - RELINK`<br>`IDN: 240: (channel` *channel*`) cookie (expected` *cookie*`, actual` *cookie*`)`<br>`IDN: 240: (channel` *channel*`) actv_flg (expected` *hex_number*`, actual` *hex_number*`)`<br>`IDN: 240: (channel` *channel*`) ready_flg (expected` *hex_number*`, actual` *hex_number*`)` | The IDN data server for network interface *channel* encountered corrupted data in the SMR. The expected and actual values for control information that is used by the data server are included (`cookie`, `actv_flg`, and `ready_flg`). Subsequent data transmissions are likely to fail. | Unlink the master domain, then relink it. If this error persists, dismantle the IDN, then reassemble it. |
| `WARNING: IDN: 241: [`*operation*`] (domain` *domain_ID*`, channel` *channel_ID* `SMR CORRUPTED - RELINK)` | The IDN driver attempted to transmit or receive data to or from an IDN mailbox in the SMR; however, the SMR was corrupted. The operation is designated as operation, `send` or `recv`, in the message. Future data transmissions are likely to fail. | Unlink the master domain, then relink relink it. If this error persists, dismantle the IDN, then relink the domains. |
| `WARNING: IDN: 242: maximum channels (`*number*`) already open` | You cannot plumb more network interfaces than the IDN driver is configured to support. | Reset the `idn.conf`(4) tunable `idn_max_nets`, then retry the operation. |

**TABLE 3-4** IDN Domain-Specific Messages, 300 through 307

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 300: no slab allocations without a master` | The slave domain attempted to allocate a slab of memory without a master domain being present. | If this error persists, unlink the local domain, then relink it. |
| `WARNING: IDN: 301: (`*SMROP*`) unknown slab state (`*slab_state*`) for domain` *domain_ID* | This is an internal error, indicating that a SMR slab was in an unexpected state for domain ID *domain_ID*, with respect to slab operation *SMROP*. | Unlink the local domain, then relink it. |
| `WARNING: IDN: 302: no slab free without a master` | The slave domain attempted to free up a slab of memory without the master domain being present. | If this error persists, unlink the local domain, then relink it. |

**TABLE 3-4** IDN Domain-Specific Messages, 300 through 307 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 303: buffer len` *bytes* `> IDN_DATA_SIZE (`*bytes*`)` | The local domain attempted to allocate an SMR buffer with a length greater than the length supported by the IDN configuration. | Unlink the local domain, then relink it. |
| `WARNING: IDN: 304: buffer (`*VADDR*`) from domain` *domain_ID* `not on a` *hex_number* `boundary` | An SMR buffer at kernel virtual address *VADDR* was received from domain ID *domain_ID*; however, it was not aligned on the expected boundary *hex_number* (in bytes). | Unlink the local domain and/or domain *domain_ID*, then relink the local domain and/or domain *domain_ID*. |
| `WARNING: IDN: 305: buffer length (`*bytes*`) from domain` *domain_ID* `greater than IDN_DATA_SIZE (`*bytes*`)` | An SMR buffer of length *bytes* was received from domain ID *domain_ID*; however, the length was greater than the length that is supported by the local domain (*bytes* in bytes). | Unlink the local domain and/or domain *domain_ID*, then relink the local domain and/or domain *domain_ID*. |
| `WARNING: IDN: 306: unknown buffer (`*hex_number*`) from domain` *domain_ID* | The local domain received a request for a domain ID, domain_ID, that was not in the expected range of valid domain IDs (that is, 0 to 15). | If this error persists, unlink the local domain, then relink it. |
| `WARNING: IDN: 307: domain id (`*domain_ID*`) invalid` | The SMR subsystem received a request for a domain ID *domain_ID* that was not in the expected range of valid domain IDs (that is, 0 to 15). | If this error persists, unlink the local domain, then relink it. |

**TABLE 3-5** IDN Domain-Specific Messages, 400 through 450

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 400: corrupted MAC header (exp` *hex_number* `or 0xffff, act` *hex_number*`)` | The MAC header in the SMR data packet contained bad data. | Unlink the local domain, then relink it. If this error persists, dismantle the IDN, and reassemble it. |
| `IDN: 450: idn`*X*`: kstat_create failed` | Unable to create the `kstat` structures. No per-interface kernel statistics will be maintained for the IDN. | Reboot the local domain if you need the per-instance kernel statistics to be maintained. |

**TABLE 3-6** IDN Domain-Specific Messages, 500 through 516

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 500: failed to write sm_bar (lsb/msb) (`*hex_number*`)` | While programming the SMR, the local domain failed to write the (`lsb/msb`) portion of the shared memory base-address-register with value *hex_number*. | Do not perform IDN operations. When appropriate, halt the local domain, and run a full diagnostic test by using the `hpost`(1M) command. |
| `WARNING: IDN: 501: failed to write sm_lar (lsb/msb) (`*hex_number*`)` | While programming access to the SMR, the local domain failed to write the (`lsb/msb`) portion of the shared memory limit-address-register with value *hex_number*. | Do not perform IDN operations. When appropriate, halt the local domain, and run a full diagnostic test by using the `hpost`(1M) command. |
| `WARNING: IDN: 502: unable to store data (`*hex_number*`) to CIC buffer (`*PADDR*`)` | While programming access to the SMR, the local domain failed to write to the CIC (Coherency Interface Controller) *prep* buffer with the data *hex_number*. | Do not perform IDN operations. When appropriate, halt the local domain, and run a full diagnostic test by using the `hpost`(1M) command. |
| `WARNING: IDN: 503: (`*PCPROG*`) failed to update PC madr (expected 0xXXX, actual `*hex_number*`)` | While programming access to the SMR during the *PCPROG* (invalidate or validate) phase, the local domain failed to write a memory address decoding register (MADR) entry to the port controller (PC). | Do not perform IDN operations. When appropriate, halt the local domain, and run a full diagnostic test by using the `hpost`(1M) command. |
| `WARNING: IDN: 504: (`*PCPROG*`) failed to update IOPC madr (expected `*hex_number*`, actual `*hex_number*`)` | While programming access to the SMR during the *PCPROG* (invalidate or validate) phase, the local domain failed to write a memory address decoding register (MADR) entry of the I/O port controller (IOPC). | Do not perform IDN operations. When appropriate, halt the local domain, and run a full diagnostic test by using the `hpost`(1M) command. |
| `WARNING: IDN: 505: board `*number*` mising any valid PCs` | Board *number* does not contain valid port controllers (PCs). | Do not perform IDN operations. When appropriate, halt the local domain to ensure that board *XX* hosts the appropriate hardware. You may need to run a full diagnostic test by using the `hpost`(1M) command. |
| `WARNING: IDN: 506: cic sm_mask is not writable` | The CIC has been programmed by POST to not let the operating system level software to manipulate the shared-memory mask register. | Ensure that the SSP 3.2 software is running on the SSP. Run a full diagnostic test on the board by using the `hpost`(1M) command. |

**TABLE 3-6** IDN Domain-Specific Messages, 500 through 516 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 507: failed to map-in post2obp structure` | The local domain did not successfully map in the POST2OBP data structure to the address space of the kernel. | Halt the local domain, then reboot it. After it reboots, relink it to the IDN. |
| `WARNING: IDN: 508: post2obp checksum invalid` | The POST2OBP data structure that was passed from POST to OBP appeared invalid. The expected checksum value did not match the value that is reported by the data structure. | Halt the local domain, then reboot it. After it reboots, relink it to the IDN. |
| `WARNING: IDN: 509: cpu` *CPUID* `never responded to CIC update` | While programming the SMR for access during the parallel update of the CIC registers phase, CPU *CPUID* did not respond to the update request. | Halt the local domain, then reboot it. After it reboots, relink it to the IDN. |
| `WARNING: IDN: 510: failed write-smregs (bd=`*number*`, bs=`*bus*`, sm(bar=`*bar*`, lar=`*lar*`)` | The local domain failed to update all of its shared memory registers. The specific failure occurred on system board *number*, interconnect bus *bus*, with base / limit-address-register contents of *bar* / *lar*, respectively. | Halt the local domain, then reboot it. After it reboots, relink it to the IDN. |
| `WARNING: IDN: 511: update-one (cpu=`*CPUID*`, bd=`*number*`) time conflict` | A stale IDN-hardware update operation was encountered during the update of the shared memory registers across the system. | Halt the local domain, then reboot it. After it reboots, relink it to the IDN. |
| `WARNING: IDN: 512: failed [`*add/delete*`] write-madr (bd=`*number*`, rbd=`*remote_number*`, madr=`*hex_number*`)` | The local domain failed to update (to add and / or to delete) the PC memory address decoding registers on board *number* with respect to the remote board, *remote_number*. The *hex_number* value represents the targeted register contents. | Halt the local domain, then reboot it. After it reboots, relink it to the IDN. |
| `WARNING: IDN: 513: sm-mask error (expected =` *hex_number*`, actual =` *hex_number*`)` | The local domain encountered inconsistent or unexpected values in the shared memory mask of the CIC. | Halt the local domain, then reboot it. After it reboots, relink it to the IDN. |
| `WARNING: IDN: 514: sm-base error (expected =` *hex_number*`, actual =` *hex_number*`)` | The local domain encountered inconsistent or unexpected values in the shared memory base register of the CIC. | Halt the local domain, then reboot it. After it reboots, relink it to the IDN. |

**TABLE 3-6** IDN Domain-Specific Messages, 500 through 516 *(Continued)*

| Error | Description | Recovery |
|---|---|---|
| `WARNING: IDN: 515: sm-limit error (expected =` *hex_number*`, actual =` *hex_number*`)` | The local domain encountered inconsistent or unexpected values in the shared memory limit register of the CIC. | Halt the local domain, then reboot it. After it reboots, relink it to the IDN. |
| `WARNING: IDN: 516: (`*local/remote*`) board` *number* `has memory, but no cpus - CPU-PER-BOARD REQUIRED` | The local domain detected that a local or remote, *local/remote*, system board, *number*, contains memory, but no CPUs. In an IDN, each system board that hosts memory must also host at least one CPU. | Halt the local or remote domain, then check its hardware configuration. If it does not host a CPU, place it in the blacklist, then relink the domain. |