



Sun Enterprise™ 10000 Dynamic Reconfiguration User Guide

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

Part No. 806-7616-10
October 2001, Revision A

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

Copyright 2001 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, CA 94303-4900 U.S.A. All rights reserved.

This product or document is 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.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, Sun Enterprise, Sun Ultra, OpenBoot, Solaris, and UltraSPARC 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.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

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 2001 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, CA 94303-4900 Etats-Unis. Tous droits réservés.

Ce produit ou document est 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.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, Sun Netra, Sun Enterprise, Sun Ultra, 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.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Adobe PostScript

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 in any 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.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

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

Preface xiii

Before You Read This Book xiii

How This Book Is Organized xiii

Using UNIX Commands xiv

Typographic Conventions xv

Shell Prompts xv

Related Documentation xvi

Ordering Sun Documentation xvi

Accessing Sun Documentation Online xvi

Sun Welcomes Your Comments xvii

1. DR on the Sun Enterprise 10000 System 1

Overview of Dynamic Reconfiguration 1

DR Concepts 2

 Detachability 2

 Quiescence 2

 Suspend-Safe and Suspend-Unsafe Devices 3

DR Models 4

 DR Model 2.0 5

	Requirements for Multipathing	5
	DR Model 3.0	5
	Requirements for Multipathing	6
	Switching DR Models	6
	▼ To Switch DR Models	6
2.	DR 2.0 Model	9
	DR 2.0 Operations	9
	Memory	10
	Pageable and Nonpageable Memory	10
	To Determine Whether Memory is Pageable or Nonpageable	11
	Target Memory Constraints	11
	Correctable Memory Errors	11
	▼ To Re-Enable Dump Detection	12
	DR 2.0 and IDNs	12
	RPC Time-Out or Loss of Connection	13
	DR 2.0 Procedures	13
	Attaching a System Board	14
	Init Attach	14
	Complete Attach	15
	Attach Buttons	15
	▼ To Attach a Board With Hostview	16
	▼ To Attach a Board Through the Command Line Interface (CLI)	19
	Detaching a System Board	23
	Drain	23
	Complete Detach	24
	Hostview Detach Buttons	25
	▼ To Detach a Board With Hostview	26

▼	To Detach a Board Through the Command Line Interface (CLI)	29
	Viewing Domain Information	31
▼	To View Domain Information With Hostview	32
▼	To Specify How Windows Are Updated	32
▼	To View DR CPU Configuration Information	33
▼	To View DR Memory Configuration Information	34
▼	To View DR Device Configuration Information	36
▼	To View DR Device Detailed Information	37
▼	To View DR OBP Configuration Information	38
▼	To View the DR-Unsafe Devices	39
3.	DR 3.0 Model	41
	Automatic DR	41
	Enhanced System Availability	42
	DR and I/O Boards	42
	Sun Enterprise 10000 Domains	42
	DR 3.0 Procedures	43
	Showing Platform Information	43
▼	To Show Platform Information	43
	Showing Device Information	44
▼	To Show Device Information	44
	Adding Boards	45
▼	To Add a Board to a Domain	45
	Deleting Boards	45
▼	To Delete a Board From a Domain	46
	Moving Boards	46
▼	To Move a Board	47
	Replacing System Boards	47

▼ To Physically Replace a System Board 47

4. DR Error Messages 49

Searching This Appendix 49

Error-Type Links 49

SSP Errors 50

Domain Errors 50

SSP Error Messages 50

Protocol and Communication Error Messages 51

Attach-Related Error Messages 56

Detach-Related Error Messages 57

Auto-Configuration Error Messages 62

Domain Error Messages 64

DR Daemon Start-Up Error Messages 64

DR Driver Error Messages 65

Memory Allocation Error Messages 68

Platform Specific Module (PSM) Error Messages 75

DR General Domain Error Messages 77

OpenBoot PROM Error Messages 79

DR Domain Exploration Error Messages 82

Unsafe-Device Query Error Messages 94

Alternate Pathing (AP) Error Messages 96

DCS Error Messages 98

DR Driver Error Messages 101

Plugin Error Messages 107

Figures

FIGURE 2-1	Attach Window	17
FIGURE 2-2	DR Window With the Init Attach Button	18
FIGURE 2-3	DR Window With the Complete Button	19
FIGURE 2-4	DR Detach Window	27
FIGURE 2-5	DR Window With the Drain Button	27
FIGURE 2-6	System Information Buttons	32
FIGURE 2-7	DR Properties Window	32
FIGURE 2-8	DR CPU Configuration Window	33
FIGURE 2-9	DR Memory Configuration Window	35
FIGURE 2-10	DR Device Configuration Window	36
FIGURE 2-11	DR Detail Device Window	37
FIGURE 2-12	DR OBP Configuration Window	38
FIGURE 2-13	DR Unsafe Devices Window	39

Tables

TABLE 2-1	Hostview Buttons	25
TABLE 2-2	DR CPU Configuration Information	34
TABLE 2-3	DR Memory Configuration Information	35
TABLE 4-1	Protocol and Communication Failure Error Messages	51
TABLE 4-2	Attach-Related Failure Error Messages	56
TABLE 4-3	Detach-Related Failure Error Messages	57
TABLE 4-4	Auto-Configuration Error Messages	62
TABLE 4-5	DR Daemon Start-Up Error Messages	64
TABLE 4-6	DR Driver Error Messages	66
TABLE 4-7	Memory Allocation Error Messages	68
TABLE 4-8	PSM Error Messages	75
TABLE 4-9	DR General Domain Error Messages	77
TABLE 4-10	OpenBoot PROM Error Messages	79
TABLE 4-11	DR Domain Exploration Error Messages	82
TABLE 4-12	Unsafe-Device Query Error Messages	94
TABLE 4-13	AP-related Error Messages	96
TABLE 4-14	DCS Error Messages	98
TABLE 4-15	DR Driver Error Messages	101
TABLE 4-16	Plugin Error Messages	107

Preface

This book describes the dynamic reconfiguration (DR) feature. The DR feature enables you to logically attach and detach system boards to and from a dynamic system domain while the operating system continues to run.

Before You Read This Book

This book is intended for the DR 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 pdf. 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 feature.

Chapter 2 contains information and procedures on the DR 2.0 model.

Chapter 3 contains information and procedures on the DR 3.0 model.

Appendix 4 lists error messages that are generated either by the SSP or 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.

See one or more of the following for this information:

- AnswerBook™ 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 <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output	% su Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type <code>rm filename</code> .

Shell Prompts

Shell	Prompt
C shell	<i>machine_name%</i>
C shell superuser	<i>machine_name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Related Documentation

Application	Title	Part Number
User information	<i>Sun Enterprise 10000 DR Configuration Guide</i>	816-2095
Command reference	<i>Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual</i>	806-7617
Installation instructions	<i>Sun Enterprise 10000 SSP 3.5 Installation Guide and Release Notes</i>	806-7615

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 Fatbrain.com at:

<http://www.fatbrain.com/documentation/sun>

Accessing Sun Documentation Online

A broad selection of Sun system documentation is located at:

<http://www.sun.com/products-n-solutions/hardware/docs>

A complete set of Solaris documentation and many other titles are located at:

<http://docs.sun.com>

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at:

`docfeedback@sun.com`

Please include the part number (806-7616-10) of your document in the subject line of your email.

DR on the Sun Enterprise 10000 System

This chapter describes what dynamic reconfiguration (DR) is and what it can do for you. Then it describes the two models of dynamic reconfiguration (DR) that are available on the Sun Enterprise™ 10000 system. It also contains procedures for switching between the DR 2.0 and DR 3.0 models.

Overview of Dynamic Reconfiguration

DR software is part of the Solaris operating environment. With the DR software you can dynamically reconfigure system boards to safely remove them or install them into a system. You perform DR operations while the Solaris operating environment is running, and with minimum disruption to user processes that are running in the dynamic system domain (referred to simply as a domain in this document).

With DR you can:

- Minimize the interruption of system applications while installing or removing a board
- Disable a failing device by removing it from the domain, before the failure can crash the operating system
- Display the operational status of boards in a domain
- Reconfigure a domain while the Solaris operating environment continues to run in the domain

DR Concepts

This section contains descriptions of general DR concepts that pertain to the Sun Enterprise 10000.

Detachability

For a device to be detachable, it must conform to the following items:

- The device driver must support `DDI_DETACH`.
- Critical resources must be redundant or accessible through an alternate pathway. CPUs and memory banks can be redundant critical resources. Disk drives are examples of critical resources that can be accessible through an alternate pathway.

Some boards cannot be detached because their resources cannot be moved. For example, if a domain has only one board, that board cannot be detached. A board is not detachable if it controls the boot drive.

If there is no alternate pathway for a board, you can:

- Put the disk chain on a separate board. The secondary board can then be detached.
- Add a second path to the device through a second board so that the board can be detached without losing access to the secondary disk chain.

Quiescence

During the unconfigure operation on a system board with permanent memory (OpenBoot™ PROM or kernel memory), the operating environment is briefly paused, which is known as operating environment *quiescence*. All operating environment and device activity on the centerplane must cease during this critical phase of the unconfigure operation.

Before it can achieve quiescence, the operating environment must temporarily suspend all processes, CPUs, and device activities. If the operating environment cannot achieve quiescence, it displays the reasons, which may include the following:

- An execution thread did not suspend.
- Real-time processes are running.
- A device exists that cannot be paused by the operating environment.

The conditions that cause processes to fail to suspend are generally temporary. Examine the reasons for the failure. If the operating environment encountered a transient condition—a failure to suspend a process—you can try the operation again.

Suspend-Safe and Suspend-Unsafe Devices

When DR suspends the operating environment, all of the device drivers that are attached to the operating environment must also be suspended. If a driver cannot be suspended (or subsequently resumed), the DR operation fails.

A *suspend-safe* device does not access memory or interrupt the system while the operating environment is in quiescence. A driver is suspend-safe if it supports operating environment quiescence (suspend/resume). A suspend-safe driver also guarantees that when a suspend request is successfully completed, the device that the driver manages will not attempt to access memory, even if the device is open when the suspend request is made. A *suspend-unsafe* device allows a memory access or a system interruption to occur while the operating environment is in quiescence.

DR uses an *unsafe driver list* in the `dr.conf` (for DR model 2.0) and `ngdr.conf` (for DR model 3.0) files to prevent unsafe devices from accessing memory or interrupting the operating environment during a DR operation. The unsafe driver list is a property in both the `dr.conf` and `ngdr.conf` files, with the following format:

```
unsupported-io-drivers="driver1" ,"driver2" ,"driver3" ;
```

DR reads this list when it prepares to suspend the operating environment so that it can unconfigure a memory component. If DR finds an active driver in the unsafe driver list, it aborts the DR operation and returns an error message. The message includes the identity of the active, unsafe driver. You must manually remove the usage of the device by performing one, or more, of the following tasks.

- Killing the processes using the device
- Unloading the driver by using the `modunload(1M)` command
- Depending on the device, disconnecting the cables

You can retry the DR operation after you have removed the usage of the device.

Note – It is a good idea to list all unsafe drivers in both the `dr.conf` and `ngdr.conf` files, whether or not you initially intend to switch between DR models. If you later decide to switch DR models, both configuration files will include references to all unsafe drivers.

DR Models

There are two models of DR available for the Sun Enterprise 10000 system. DR model 2.0 is sometimes referred to as “legacy DR,” and DR model 3.0 is referred to as “next generation DR.” The DR model is determined by the configuration of the Solaris operating environment on the domain.

DR model 3.0 is available starting with the Solaris 8 10/01 operating environment, but requires that the SSP be running version 3.5 of the SSP software. Either DR model 2.0 or 3.0 can be enabled on the Solaris 8 10/01 operating environment, whereas only DR 2.0 exists in versions of Solaris prior to Solaris 8 10/01.

Only one model of DR can run within a domain at a time. To check the version of DR that is running, use the following command: `domain_status -m`. (The `-m` flag is available only with version 3.5 of the SSP software.)

The Sun Enterprise 10000 system can switch from one model of DR to the other. See “Switching DR Models” on page 6 for information about switching between DR models 2.0 and 3.0.

Make sure to use the `domain_status(1M)` command with its `-m` option to verify the DR model before you execute any DR commands. The following is an example of the `domain_status(1M)` output. The `DR-MODEL` column indicates which model is enabled.

```
# domain_status -m
```

DOMAIN	TYPE	PLATFORM	DR-MODEL	OS	SYSBDS
A	Ultra-Enterprise-10000	all-A	2.0	5.8	2
B	Ultra-Enterprise-10000	all-A	3.0	5.8	3 4
C	Ultra-Enterprise-10000	all-A	2.0	5.7	5 6

According to this output, domain A is running Solaris version 8 software (OS 5.8) with DR model 2.0 enabled; domain B is running Solaris version 8 software with DR model 3.0 enabled; and domain C is running Solaris version 7 software (OS 5.7) with DR model 2.0 enabled. (Only DR version 2.0 is available on Solaris version 7 software.)

Only certain commands are available in each model, and if you execute a command that is not supported, an error message appears on the console.

DR Model 2.0

Versions 2.0 and 3.0 of dynamic reconfiguration (DR) enable you to logically attach and detach system boards to and from the operating system without causing machine downtime. DR is used in conjunction with hot swapping, which is the process of physically removing or inserting a system board. You can use DR to add a new system board, reinstall a repaired system board, or modify the domain configuration on the Sun Enterprise 10000 system.

If a system board is being used by a domain, you must detach it before you can power it off and remove it. After a new or upgraded system board is inserted and powered on, you can attach it to the domain.

You can perform DR operations from the system service processor (SSP) by using the Automated DR (ADR) commands: `addboard(1M)`, `moveboard(1M)`, `deleteboard(1M)`, and `showusage(1M)`.

For more information about using DR 2.0, see the Section “DR 2.0 Procedures” on page 13.

Requirements for Multipathing

To use multipathing on DR model 2.0 domains, run the Sun Enterprise Server Alternate Pathing software. Any version of the alternate pathing software is compatible with DR model 2.0. For further AP information, refer to the Sun Enterprise Server Alternate Pathing User Guide.

DR Model 3.0

The DR 3.0 model offers the following enhancements to DR 2.0:

- DR 3.0 has a framework that offers better integration with applications, through the Reconfiguration Coordination Manager.
- DR 3.0 supports network multipathing using IPMP.

You can execute DR operations from the system service processor (SSP) by using the SSP commands—`addboard(1M)`, `moveboard(1M)`, `deleteboard(1M)`, `rcfgadm(1M)`, and `showdevices(1M)`.



Caution – Before you switch to DR 3.0 in a domain that is running the Solaris 8 10/01 operating environment, you must upgrade the SSP software to version 3.5. Previous versions of SSP do not support DR 3.0 operations.

For more information about DR 3.0, see the Section “DR 3.0 Procedures” on page 43.

Requirements for Multipathing

To use multipathing on DR model 3.0 domains, run IPMP (the IP multi-pathing software provided with the Solaris 8 operating environment) and MPxIO software, included in Solaris Kernel Update Patches 111412-02, 111413-02, 111095-02, 111096-02, and 111097-02.

Switching DR Models

By default, DR 2.0 is enabled when you install the Solaris 8 10/01 operating environment. However, you can switch the model by using the following procedures. The steps in both of the following procedures are executed on the domain.



Caution – Before you switch to DR 3.0 in a domain that is running the Solaris 8 10/01 operating environment, you must upgrade the SSP software to version 3.5. Previous versions of SSP do not support DR 3.0 operations.

▼ To Switch DR Models

1. Change to the following directory:

```
% cd /platform/SUNW,Ultra-Enterprise-10000/kernel/drv
```

2. To change DR models, modify the value assigned to the `legacy-dr-model` variable in the `ngdr.conf` file. (The default value of `legacy-dr-model` is 1.)
3. To switch to DR 3.0, set `legacy-dr-model` to 0 (zero).

```
% legacy-dr-model=0;
```

Reboot the domain. After the reboot, you can verify that DR 3.0 is enabled by viewing the `/var/adm/messages` file for the following message.

```
NOTICE: Next Generation DR Model (DR 3.0) is enabled
```


4. To switch to DR 2.0, set `legacy-dr-model` to 1.

```
% legacy-dr-model=1;
```

Reboot the domain. After the reboot, you can verify that the 2.0 (legacy) model is enabled by viewing the `/var/adm/messages` file for the following message.

```
NOTICE: Legacy DR Model (DR 2.0) is enabled
```

5. To ensure that the SSP recognized the model switch, execute the following command on the SSP: `domain_status -m`.

DR 2.0 Model

This chapter contains the introductory material found in the DR user guides in past SSP releases. In addition, the chapter contains the procedures on using DR 2.0 that were described in past DR user guides.

The DR 2.0 model is based on the use of the DR daemon, `dr_daemon(1M)`, to control DR operations. Model 2.0 includes the Automated DR (ADR) commands, such as `addboard(1M)`, `deleteboard(1M)`, and `moveboard(1M)`. Using ADR commands is described in Section “DR 3.0 Procedures” on page 43 .

DR 2.0 still supports DR commands that are executed in the `drshell(1M)` and on the Hostview DR menu.

DR 2.0 Operations

You can perform DR operations from the SSP by using the Hostview GUI or the `dr(1M)` shell application (refer to the `hostview(1M)` and `dr(1M)` man pages for more information). DR supports the following operations:

- **DR Attach** — Logically attaches a system board to the operating system running in a domain. A system board is logically attached when its resources (processors, memory, and I/O adapters) are configured into a domain and are available to the Solaris operating environment. The system board must already be present in the system, powered on, and not be a member of a domain. Normally, you attach a system board after it is inserted and powered on by your service provider or after it is detached from another domain.
- **DR Detach** — Logically detaches a system board from a domain. A system board is logically detached when its resources (processors, memory, and I/O adapters) are removed from the domain configuration and are no longer available to the domain. Normally, you detach a system board to either move it to another domain or prepare it for removal.

While DR operations are being performed within a domain, the `dr_daemon(1M)` (refer to the *Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual*) and the operating environment write messages regarding the status or exceptions of DR requests to the domain syslog message buffer (`/var/adm/messages`) and the SSP message files (`$$SSPOPT/adm/domainName/messages` and `$$SSPOPT/adm/messages`). In addition to the status and exception information displayed by Hostview and the `dr(1M)` shell application, the `dr_daemon(1M)` and operating environment messages are useful for determining the status of DR requests.

Note – Only one DR operation per platform can be active at any one time. A partially completed DR operation must be finished before a subsequent DR operation can run in the same domain. A DR operation that is partially completed and then dismissed within one domain does not prevent a subsequent DR operation from being started in a different domain.

Memory

If you use memory interleaving between system boards, those system boards cannot be detached because DR does not yet support interboard interleaving. By default, `hpost(1M)` does not set up boards with interleaved memory. Look for the following line in the `hpost(1M)` file `.postrc` (`postrc(4)`):

```
mem_board_interleave_ok
```

If `mem_board_interleave_ok` is displayed, you may not be able to detach a board that uses memory interleaving.

Pageable and Nonpageable Memory

Before you can detach a board, the operating system must vacate the memory on that board. Vacating a board means flushing its pageable memory to swap space and copying its nonpageable (that is, kernel and OpenBoot™ ROM (OBP) memory) to another memory board. To relocate nonpageable memory, the operating environment on a domain must be temporarily suspended, or quiesced. The length of the suspension depends on the domain I/O configuration and the running workloads. Detaching a board with nonpageable memory is the only time when the operating environment is suspended; therefore, you should know where nonpageable memory resides, so you can avoid significantly impacting the operation of the domain. When permanent memory is on the board, the operating environment must find other memory to receive the copy.

To Determine Whether Memory is Pageable or Nonpageable

To determine whether the memory on a board is pageable or nonpageable use the `drshow(1M)` command:

```
% dr
dr> drshow board_number mem
```

Similarly, you can determine whether the memory on a board is pageable by looking at the DR Memory Configuration window, which is available when you perform a detach operation within Hostview. The DR Memory Configuration window is described in the *Sun Enterprise 10000 DR Configuration Guide* in the Solaris 8 10/01 version of the Sun Hardware Answerbook2 Collection.

Target Memory Constraints

When permanent memory is detached, DR chooses a target memory area to receive a copy of the memory. The DR software automatically checks for total adherence. It does not allow the DR memory operation to continue if it cannot verify total adherence. A DR memory operation can be disallowed for the following reasons:

- The domain is not large enough to hold a copy of the nonpageable memory.
- The domain is interleaved with memory on other boards.

In Solaris 7 and subsequent compatible releases of the Solaris operating environment, if no target board is found, the detach operation is refused, and DR displays an error message. (See Appendix A, “DR Error Messages” on page 49 for more information about DR error messages.)

Correctable Memory Errors

Correctable memory errors indicate that the memory on a system board (that is, one or more of its dual inline memory modules (DIMMs), or portions of the hardware interconnect) may be faulty and need replacement. When the SSP detects correctable memory errors, it initiates a record-stop dump to save the diagnostic data, which can interfere with a DR Detach operation. When a record-stop occurs from a correctable memory error, Sun Microsystems™ suggests that you allow the record-stop dump to complete its process before you initiate a DR Detach operation.

If the faulty component causes repeated reporting of correctable memory errors, the SSP performs multiple record-stop dumps. If this happens, you should temporarily disable the dump-detection mechanism on the SSP, allow the current dump to finish, then initiate the DR Detach operation.

Note – After the detach operation finishes, you should re-enable the dump detection.

▼ To Re-Enable Dump Detection

1. **Log into the SSP as the user `ssp`.**
2. **Disable record-stop dump detection:**

```
ssp% edd_cmd -x stop
```

This command suspends all event detection on all domains.

3. **Monitor the in-progress record-stop dump:**

```
ssp% ps -ef | grep hpost
```

In the `grep(1)` output, the `-D` option of `hpost` indicates that a record-stop dump is in progress.

4. **Perform the DR Detach operation.**
5. **Enable event detection:**

```
ssp% edd_cmd -x start
```

DR 2.0 and IDNs

The IDN feature enables domains to communicate with to each other over the interconnect by using standard TCP/IP protocols. To provide this capability, the IDN feature maintains detailed information about the hardware configuration and is dependent on the hardware configuration of the member domains.

The DR feature enables the user to reconfigure the hardware while the operating system is running. DR makes an IDN aware of any changes so that the IDN can maintain consistent, up-to-date information about the hardware. DR accomplishes this requirement by unlinking the domain from the IDN, reconfiguring the hardware, and relinking the domain to the IDN.

The unlinking and relinking of the domain occurs during the complete attach or complete detach phase of the DR operation. DR determines whether the domain is a member of an IDN, and it performs the unlinking and relinking of the domain during the complete phase.

No interaction is needed by the user. However, if a member domain is in an unknown state, the unlink operation does not succeed, especially if the domain is in a non-responsive state. If one or more domains are in an unknown state when you attempt to perform a DR operation, you must unlink all those domains within the IDN in a single step (that is, use the `domain_unlink(1M)` command with all of the names of those domains).

During the time period in which the domain is not linked to the IDN, no transmission to or from the domain is allowed. The domain remains a member of the IDN as defined in the `domain_config(4)` file on the SSP, and the domain continues to be listed as a member of the IDN when you use the `domain_status(1M)` command.

Note – Due to the interaction between the DR and IDN features, only one DR or IDN operation is allowed at any given time within a single Sun Enterprise 10000 system.

Certain conditions may require you to use the `force` option. In the context of a DR operation, you can use the DR `force` option, which is passed to the `domain_unlink(1M)` command. When used on a domain that is a member of an IDN, the `force` option should be used with extreme care. Refer to the *Sun Enterprise 10000 InterDomain Networks User Guide* for more information about the `force` option and its use.

RPC Time-Out or Loss of Connection

The `dr_daemon(1M)`, which runs in each domain, communicates with Hostview and the `dr(1M)` shell application (both of which run on the SSP) by way of remote procedure calls (RPCs).

For more information about RPC time-outs and loss of connection failures, refer to the *Sun Enterprise 10000 DR Configuration Guide*.

DR 2.0 Procedures

This section contains information and procedures for using the DR 2.0 features.

Note – For information and procedures on how to configure and reconfigure the domain before and after a DR operation, refer to the *Sun Enterprise 10000 DR Configuration Guide* in the Solaris 8 10/01 release in the Sun Hardware Answerbook2™ Collection.

Attaching a System Board

This section gives a broad overview of the actions that occur when you execute DR Attach. For step-by-step instructions using the GUI, see “To Attach a Board With Hostview” on page 16. Or see “To Attach a Board Through the Command Line Interface (CLI)” on page 19 .

You can attach system boards that are present in the machine, powered on, and not part of an active domain (that is, not being used by an operating environment). These unattached boards may have been hot-swapped into the domain after the domain was booted, blacklisted when the domain was booted, or detached from another domain.

Note – If the system board has been hot-swapped into the domain, you must use the `thermal_config(1M)` command immediately after the board has been powered on.

Prior to attaching a board, diagnostics are run on the board. At least one processor must be present on the board, and it must not be blacklisted. After you have selected an eligible board and a target domain, the DR Attach operation proceeds through two operations: init attach and complete attach.

Init Attach

During the init attach phase, DR diagnoses and configures the selected board, preparing it and its devices for attachment to the operating environment. During this phase, DR performs the following tasks:

- Adds the board to the board list of the target domain in the `domain_config(4)` file on the SSP.
- Runs `hpost -H` on the board to configure it. `hpost(1M)` isolates the board on the Sun Enterprise 10000 system centerplane by placing it into a single-board hardware domain (refer to the `hpost(1M)` man page).

Note – Depending on the presence of memory and I/O devices on the board, the `hpost` testing level can be different for subsequent attach operations.

- Runs `obp_helper -H` which loads `download_helper` to the board, and takes the processors on the board out of reset mode, enabling them to spin in `download_helper`.
- Reconfigures the centerplane and board domain mask registers, placing the board in the target hardware domain.

DR displays the output of these `hpost(1M)` and `obp_helper(1M)` operations, including the steps that succeeded and those that caused exceptions.

If `hpost(1M)` and `obp_helper(1M)` succeed, the operating system is notified and requests OBP to probe the board. The operating environment then scans the OBP device tree and adds the devices to its configuration, but the drivers are not loaded.

After the init attach phase has completed, the OBP board configuration can be displayed to confirm which devices are present on the board. You can then enter the complete attach phase, or you can abort the operation.

If you abort the operation, DR removes the board configuration from the operating environment data structures and removes the board from the `domain_config(4)` file, leaving the board in a state where it is not assigned to any domain. The board can then be hot swapped out of the system, left in the system unattached, or attached at a later time.

Complete Attach

During the complete attach phase, DR attempts to complete the attach operation by making the resources that are hosted by the new system board available to the operating environment. If a problem occurs that prevents the attachment of any device on the board, the `dr_daemon(1M)` logs that problem in the system message buffer. To determine which devices were successfully attached, display and check the domain configuration for the board.

After a board is successfully attached, you have the option of reconfiguring the I/O devices. Refer to the *Sun Enterprise 10000 DR Configuration Guide* for more information. This operation can take several minutes to complete.

Attach Buttons

When you perform an attach operation using the Hostview GUI (which transparently calls a separate executable `drview(1M)`), the following buttons appear at various times during the attach process:

- **Init attach**—Begins the attach operation (see “Init Attach” on page 14). After the operation has completed successfully, the label on this button changes to complete.
- **Complete**—Completes the attach operation (see “Complete Attach” on page 15).
- **Reconfig**—Automatically reconfigures the device directories in the domain. You may want to run the reconfiguration operation after attaching a board (refer to the *Sun Enterprise 10000 DR Configuration Guide*).
- **Abort**—Cancels the attach operation. This button is enabled after the init attach operation has been successfully completed (see “Init Attach” on page 14).
- **Dismiss**—Terminates the step that is currently in progress, but leaves the board in its current state (Present, init attach, In Use). You can remove the DR Attach window by choosing dismiss at any point during the attach operation. The dismiss button terminates any work being done on the SSP for the attach operation. For example, if `hpost(1M)` is running when you click dismiss, that `hpost(1M)` process is terminated. Note that dismiss does not terminate work being done on the host by way of RPCs to the `dr_daemon(1M)`. After an RPC is initiated, the host completes the RPC regardless of whether or not the calling program is waiting for the RPC to finish. The host `dr_daemon(1M)` keeps track of the progress of the attach operation. The state of the init attach operation is retained after the process completes successfully. Therefore, you can dismiss the window, then return to the DR operation later and complete or abort the attach.
- **Help**—Accesses online information regarding DR Attach operations.

▼ To Attach a Board With Hostview

Note – Before you perform the following steps, you should read “Attaching a System Board” on page 14.

1. **From the Hostview window, use the View menu to select the proper view of the system.**
2. **From Hostview, Choose the view that contains the board you want to attach, and select the board.**
3. **From Hostview, choose Configuration > Board > Attach.**

The attach Board and Domain Selection window is displayed. (FIGURE 2-1)

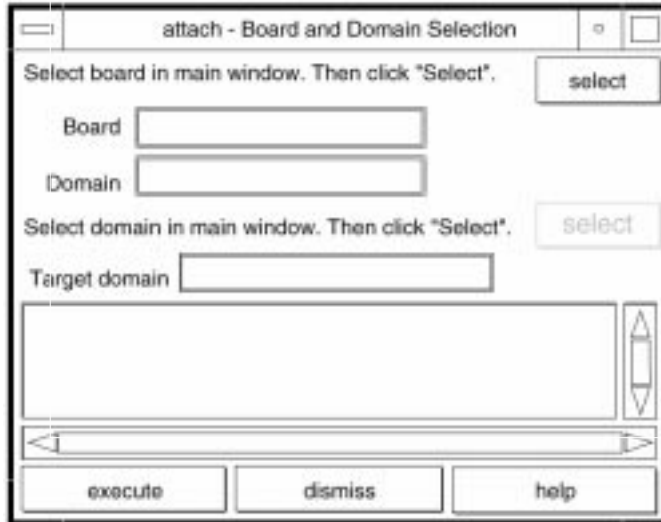


FIGURE 2-1 Attach Window

4. Click the select button at the top.

The Board field is automatically filled in for you. If the board is part of a domain, the Domain field is also filled in. (You can edit these fields manually as well.)

5. In the main Hostview window, use the View menu to select the domain to which you want to attach the board.

6. Click the select button at the bottom.

The Target Domain field is automatically filled in. (You can edit the field manually as well.)

7. Click the execute button.

If errors are detected, the error messages appear in the main Hostview window. Otherwise, the Dynamic Reconfiguration window is displayed with the init attach button visible. (FIGURE 2-2)

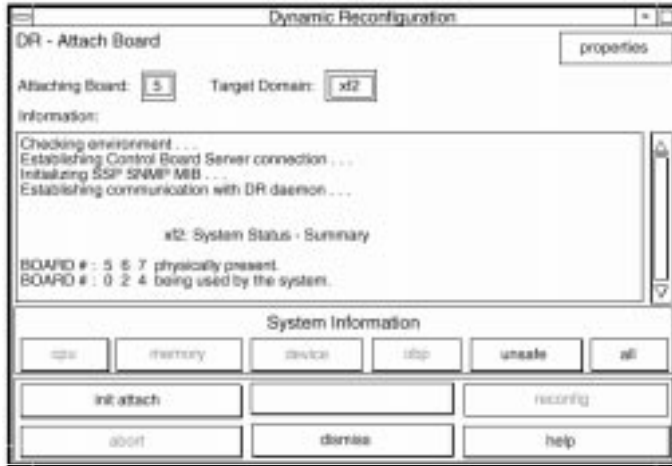


FIGURE 2-2 DR Window With the Init Attach Button

8. Click the init attach button.

Clicking on the init attach button begins the first phase of the board attach process. First, the system updates the `SSP domain.config(4)` file by adding the system board to the board list of the target domain. Next, the system uses `hpost(1M)` to self-test the system board. After the self-test is complete, the board is made visible to the running target domain by merging it into the hardware domain by modifying the centerplane and the system board hardware registers. Finally, during the conclusion of the init attach, OBP probes the new board to discover what CPU, I/O, and memory resources are present on the board. When this phase is finished, the caption on the button changes to complete. Before you click the complete button, however, view the domain information before proceeding, as described in “Viewing Domain Information” on page 31.

Typically, the init attach operation can take a few minutes to complete. Output from the `hpost(1M)` command is directed to the Information pane of the Dynamic Reconfiguration window.

If the init attach fails, look for the cause in the output in the Information pane. After determining the cause, you can choose init attach again.

If the init attach operation completes successfully, the window changes with the Complete button enabled.

9. Click the complete button. (FIGURE 2-3)

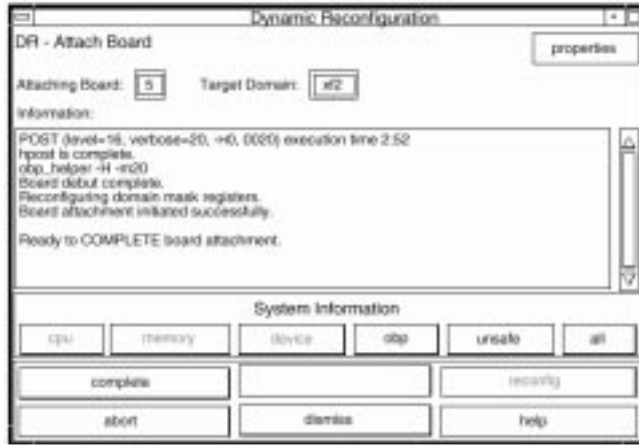


FIGURE 2-3 DR Window With the Complete Button

The complete operation normally takes less than one minute to finish. After the complete operation has executed successfully, DR displays the following message:

```
Board attachment completed successfully
```

The system board resources—processors, memory, and I/O devices—are now available to the operating system.

You can view the domain information about the newly attached board by using the buttons (CPU, Memory, Device, and so forth), as described in “Viewing Domain Information” on page 31.

Caution – Before you choose the reconfig button, refer to the *Sun Enterprise 10000 DR Configuration Guide*.

10. Click the dismiss button.

The DR Attach operation is complete.

▼ **To Attach a Board Through the Command Line Interface (CLI)**

Before you perform the following steps, read “Attaching a System Board” on page 14. The process of attaching a board is very similar whether you use Hostview or `dr(1M)`. The basic concepts are not repeated in this section. A quick reference guide is available in the `dr(1M)` application by using the `help` command.

1. Set `SUNW_HOSTNAME` to the appropriate domain by using the `domain_switch(1M)` command.

```
% domain_switch domain_name
```

2. Use the `dr(1M)` command in an SSP Window to bring up the `dr(1M)` prompt.
In the following example, the target domain is called `xf3`.

```
% dr
Checking environment...
Establishing Control Board Server connection...
Initializing SSP SNMP MIB...
Establishing communication with DR daemon...

      xf3: Domain Status - Summary

BOARD #: 0 1 2 5 6 8 9 10 11 13 physically present.
BOARD #: 4 7 being used by the domain.
dr>
```

3. Begin the `init_attach(1M)` operation for the designated board.

In this example, board 6 is being attached to `xf3` domain.

```
dr> init_attach 6
Initiate attaching board 6 to domain xf3.
Adding board 6 to domain_config file.
/opt/SUNWssp/bin/hpost -H40,28
Opening SNMP server library...

Significant contents of /export/home/ssp/.postrc:
blacklist_file ./bf
redlist_file ./rf
Reading centerplane asics to obtain bus configuration...
Bus configuration established as 3F.
phase cplane_isolate: CP domain cluster mask clear...
...
phase final_config: Final configuration...
Configuring in 3F, FOM = 2048.00: 4 procs, 4 SCards, 1024 MBytes.
Creating OBP handoff structures...
Configured in 3F with 4 processors, 4 SBus cards, 1024 MBytes
memory.
Interconnect frequency is 83.294 MHz, from SNMP MIB.
Processor    frequency is 166.631 MHz, from SNMP MIB.
Boot processor is 6.0 = 24
POST (level=16, verbose=20, -H28,0040) execution time 3:07
hpost is complete.
obp_helper -H -m24
Board debut complete.
Reconfiguring domain mask registers.
Board attachment initiated successfully.

Ready to COMPLETE board attachment.
```

4. Abort or complete the attach operation.

- After the system successfully completes the `init_attach(1M)` operation, you can use the `drshow(1M)` OBP display to see an inventory of the board resources.

```
dr> drshow board_number OBP
```

- If you wish to abort the attach operation, use the `abort_attach(1M)` command.

```
dr> abort_attach board_number
```

- If you wish to complete the board attach operation, use the `complete_attach(1M)` command.

```
dr> complete_attach 6
Completing attach for board 6.
...Checking IDN state of domain_name_a : UP
Issuing IDN UNLINK (domain_name_a)
Verifying IDN UNLINK...
IDN (XM) UNLINK succeeded (domain_name)
...Checking IDN state of domain_name_a : UP
...Checking IDN state of domain_name_b : UP
Initiating IDN LINK...
IDN LINK succeeded (domain_name_a + domain_name_b)
Board attachment completed successfully.
dr>
```

After you successfully attach the board, all of the `drshow(1M)` displays become available.

5. Use the `drshow(1M)` to display the I/O information for the newly attached board.

```
dr> drshow 6 IO

      SBus Controllers and Devices for Board 6

----- Sbus 0 : Slot 0 : SUNW,pln0 -----

device      opens   name                               usage
-----
ssd0         0      /dev/dsk/clt0d0s0
ssd16        0      /dev/dsk/clt1d0s0
ssd32        0      /dev/dsk/clt2d0s0
ssd48        0      /dev/dsk/clt3d0s0
ssd64        0      /dev/dsk/clt4d0s0
ssd80        0      /dev/dsk/clt5d0s0

----- Sbus 0 : Slot 1 : SUNW,pln2 -----

device      opens   name                               usage
-----
ssd96        0      /dev/dsk/c2t0d0s0
ssd97        0      /dev/dsk/c2t0d1s0
...
```


6. Type `exit` to terminate this `dr(1M)` session.

```
dr> exit
%
```

The SSP login shell prompt is again displayed.

Detaching a System Board

This section gives a broad overview of the actions that occur when you execute DR Detach. For step-by-step instructions using the GUI, see “To Detach a Board With Hostview” on page 26. Or see “To Detach a Board Through the Command Line Interface (CLI)” on page 29.

System boards that are currently being used by the operating environment can be detached if they meet the requirements covered in the *Sun Enterprise 10000 DR Configuration Guide*. After you select an eligible board, you can detach that board by performing two operations: drain and complete detach.

Drain

The primary function of the drain operation is to determine how the board’s memory is to be vacated by the operating environment and, if required, to select a target memory area for copying the nonpageable memory on a board. If a suitable target memory area is not available when the drain operation is requested, the request is denied. If the drain is rejected for this reason, you can continue to retry until target memory is available. Refer to the *Sun Enterprise 10000 DR Configuration Guide*.

After the drain operation is started, the pageable memory on the board is flushed to a disk, which removes it from use by the domain. Whenever a page of memory becomes free, that page is locked from further use. The drain has no noticeable impact on the processes using the CPU and I/O resources on the board. However, less memory is available to the domain.

Note – After memory is drained, enough memory and swap space must remain in the domain to accommodate the current workloads.

During the drain period, Hostview and `dr(1M)` are available to monitor the detach progress. You can view the current status of the drain operation, including the number of memory pages remaining to be drained, and device usage on the board. With this information, you can prepare the domain for detaching the remaining board devices.

If you decide not to proceed with the detach operation, you can abort the operation and return the memory on the board to regular usage. You can also abort the operation during the drain process or after the drain has been completed. If extreme memory pressure exists during the drain, you will see little, or no, progression in the percentage of drained pages. You may want to abort the drain and wait until the workload on the domain has decreased, enabling it to accommodate the reduction in memory.

The drain operation is complete when all memory pages are free from usage. You can then complete the detach operation.

Complete Detach

Before the detach operation can be completed, you must terminate all usage of board resources (processors, memory, and I/O devices). DR terminates the use of memory, processors, and network devices automatically; however, you must terminate the use of all non-network I/O devices.

Note – To identify components on the board that are to be detached, use `drshow(1M)`, which is an option of the `dr(1M)` command, or use the display windows in Hostview (select the Configuration menu and then choose the Board pull-down menu and the Detach menu item). Another somewhat less informative way to identify the components is to use the `prtdiag(1M)` command on the domain.

After all board usage is terminated, you can perform the complete detach operation. If a device is still in use at this time, the detach operation fails and the device in use is reported. After the problem is resolved, perform the complete detach operation again.

If the board to be detached contains nonpageable memory, the complete detach operation may also fail due to quiescence problems, which are described in the *Sun Enterprise 10000 DR Configuration Guide*. After the problem with quiescence is resolved, perform the complete detach operation.

If you do not want to proceed with the detach operation at this time, you can abort the detach operation. The memory on a board is returned to normal usage and detached board devices are reattached. If the system configuration was modified to

remove the board from use (that is, file systems were unmounted and networks were unplumbed), you must undo these modifications and return the devices to normal operation.

After the board is successfully detached from the operating environment, it is isolated from the centerplane. In addition, the board list is automatically updated in the SSP `domain_config(4)` file.

You can now attach the board to another domain, or leave it unattached.

Hostview Detach Buttons

The Hostview detach window displays the following buttons at various times during a detach operation:

TABLE 2-1 Hostview Buttons

Button	Description
drain	Drains the memory (see “Drain” on page 23). After the drain operation is finished, the drain button becomes the complete button.
complete	Completes the detach operation after the board has been fully drained (see “Complete Detach” on page 24).
force	Permits you to complete the detach operation by forcibly quiescing the domain (refer to the <i>Sun Enterprise 10000 DR Configuration Guide</i>). If the complete detach operation fails due to a forcible quiesce condition, the force button is enabled.
reconfig	Reconfigures device directories in a domain automatically. You may want to run <code>reconfig</code> after permanently detaching a board. Use <code>reconfig</code> with extreme caution (refer to the <i>Sun Enterprise 10000 DR Configuration Guide</i> for more information).

TABLE 2-1 Hostview Buttons (*Continued*)

Button	Description
abort	Cancels the DR operation, and returns the board to normal operation. This button is enabled after the drain operation starts and remains enabled until the complete detach operation starts. To stop the draining of memory and cancel the detach, choose abort (see “Detaching a System Board” on page 23).
dismiss	Cancels any step that is in progress, and leaves the board in its current state (In Use, Drain, Present). At any point during the DR Detach operation you can close the DR Detach window by choosing dismiss, which terminates any work being done on the SSP for the detach operation. Note that dismiss does not terminate work being done on the host through RPC calls to the <code>dr_daemon(1M)</code> . After an RPC call is initiated, the host completes the RPC call regardless of whether Hostview is waiting for the RPC call to finish. The host <code>dr_daemon(1M)</code> keeps track of the progress of the detach operation after the drain is started. Therefore, you can dismiss the window and then return later to either complete or abort the detach operation.
help	Accesses online information regarding DR Detach operations.

▼ To Detach a Board With Hostview

Note – Before you execute the following steps, read “Detaching a System Board” on page 23.

1. **From the Hostview window, use the View menu to select the domain in which the board is attached, and click the icon of the board to detach.**
2. **From the Hostview menu, choose Configuration > Board > Detach.**

The detach - Board and Domain Selection window is displayed. (FIGURE 2-4)

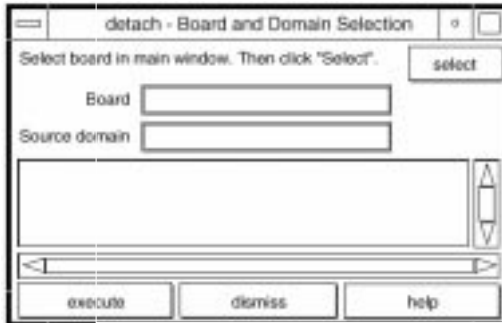


FIGURE 2-4 DR Detach Window

3. Click the select button.

The Board and Source domain fields are automatically filled in. (You can edit these fields manually as well.)

4. Click the execute button.

If the target domain is not currently booted, the detach operation simply manipulates the domain configuration file on the SSP. However, if the domain is running, the following window is displayed. (FIGURE 2-5)

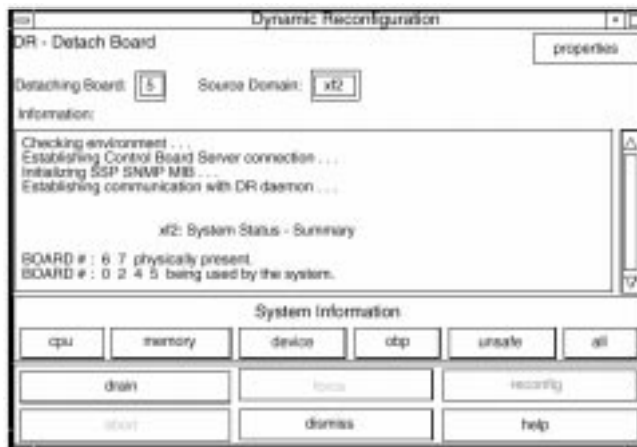


FIGURE 2-5 DR Window With the Drain Button

5. Click the drain button.

Hostview begins draining memory. The memory information is displayed and enables you to monitor the progress of the drain operation.

Memory drain statistics are automatically updated at periodic intervals if the Auto Update Domain Information Displays option in the DR Properties window is enabled. (See “Viewing Domain Information” on page 31.)

If the drain operation fails, an explanatory message appears in the Information pane. After you have determined the cause, and corrected it, you can choose drain again.

You can proceed to the next step without waiting for the drain operation to complete.

6. To determine which devices are active on the board, click the device button.

The DR Device Configuration window is displayed and is periodically updated, providing you with a current snapshot of device usage.

7. Terminate all usage of board-resident I/O devices.

For more information, see “Complete Detach” on page 24.

When the complete button is displayed, DR is finished draining the memory, and you can proceed to the next step.

8. Click the complete button.

This operation may take several minutes to complete, particularly if an operating environment quiescence is necessary. When the complete operation is finished, the board devices are detached from the operating system.

If your attempt to complete the detach fails, it may be due to any of the following reasons:

- All online processors in the domain are on the board being detached.
- The board you want to detach contains the last processor in the “default” processor set. You must add an additional processor from another system board before you retry the detach operation.
- Primary network interfaces are on the board being detached. You must stop all usage of these networks manually (see “Complete Detach” on page 24).
- All usage of the I/O devices on the board you want to detach has not been stopped. The Information pane identifies the device on which the error was encountered (see “Complete Detach” on page 24).
- The operating environment quiescence failed. You must determine and resolve the cause of the error (refer to the *Sun Enterprise 10000 DR Configuration Guide*).

After resolving the cause of the failure, you can select either the complete or force process to complete the detach operation. If no further problems occur, the board is detached and reset. When the board is successfully detached, the following message is displayed:

```
Board detachment completed successfully.
```



Caution – Before you choose the reconfig button, refer to the *Sun Enterprise 10000 DR Configuration Guide*.

You can now either reconfigure the device directories or dismiss the detach window. The board can be powered off and removed by hot-swapping; attached to another domain; left in the system unattached; or reattached at a later time.

▼ To Detach a Board Through the Command Line Interface (CLI)

Before you execute the following steps, read “Detaching a System Board” on page 23. The process of detaching a board is very similar with either Hostview or `dr(1M)`.

1. **Set `SUNW_HOSTNAME` to the appropriate domain using the `domain_switch(1M)` command.**
2. **Use the `dr(1M)` command in an SSP Window to bring up the `dr(1M)` prompt.**
In the following example, the target domain is called `xf3`.

```
% dr
Checking environment...
Establishing Control Board Server connection...
Initializing SSP SNMP MIB...
Establishing communication with DR daemon...

xf3: Domain Status - Summary

BOARD #: 0 1 2 5 6 8 9 10 11 13 physically present.
BOARD #: 4 7 being used by the domain.
dr>
```

3. Use drain(1M) to drain the board.

```
dr> drain 6
Removing board 6 from domain_config file.
Start draining board 6
Board drain started. Retrieving Domain Info...

    Bound Processes for Board 6

cpu    user  sys  procs
----  -
24     0    1
25     0    1
26     0    1
27     0    1

    Active Devices for Board 6

device    opens  name                usage
-----  -
ssd384    0     /dev/rdsk/c5t0d0s4  AP database

    Memory Drain for Board 6 - IN PROGRESS

Reduction= 1024 MBytes
Remaining in Domain= 1024 MBytes
Percent Complete= 99% (5696 KBytes remaining)

Drain operation started at Wed Oct 09 18:06:00 1996
Current time                Wed Oct 09 18:06:34 1996
Memory Drain is in progress. When Drain has finished,
you may COMPLETE the board detach.

dr>
```

The drain(1M) command initiates the drain operation and returns to the shell prompt immediately. You can monitor the progress of the drain operation with the following command:

```
dr> drshow board_number drain
```

Note – In addition, you can initiate the drain with the `wait` option of the `drain(1M)` command, which does not return to the shell prompt until after the drain has completed. Refer to `drain(1M)` for more information regarding the `wait` option.

4. After the drain operation has finished successfully, use the `complete_detach(1M)` command to complete the detach.

```
dr> complete_detach 6
Completing detach of board 6
...Checking IDN state of domain_name_a : UP
Issuing IDN UNLINK (domain_name_a)
Verifying IDN UNLINK...
IDN (XM) UNLINK succeeded (domain_name)
Operating System has detached the board.
Reconfiguring domain mask registers.
...Checking IDN state of domain_name_a : UP
...Checking IDN state of domain_name_b : UP
Initiating IDN LINK...
IDN LINK succeeded (domain_name_a + domain_name_b)
Board 6 placed into loopback.
Board detachment completed successfully.
dr>
```

If the `complete_detach` fails with the message `Operating system failed to quiesce due to forcible conditions`, and if you have determined the root cause of the quiescent failure, you can retry the `complete_detach` with the `force` option. (You can see the console messages to help determine the cause of the quiescent failure.) Refer to `complete_detach(1M)` for more information.

You can abort the detach operation, rather than complete it. To do so, use the command `abort_detach board_number`, instead of the `complete_detach` command shown above.

Viewing Domain Information

Both `dr(1M)` and `Hostview` enable you to display information about the suspend-unsafe devices as well as information about the board selected during DR operations. For `dr(1M)`, this information is accessible by using the `drshow(1M)` command. From `Hostview`, this information is available by clicking the `cpu`, `memory`, `device`, `obp`, and `unsafe` buttons in the attach or detach windows.

Note – View and use the domain information *before* you attempt to drain the memory on the board.

The informational content is the same for both `dr(1M)` and Hostview. Note that the cpu, memory, and device displays are only enabled when the board is attached to the operating environment. When the cpu, memory, and device displays are available, they always contain accurate information. The obp display shows the information that is available to OBP, but it is not as detailed as the other three displays. This section shows how to use the displays.

▼ To View Domain Information With Hostview

1. Click on any of the System Info buttons during the DR operation.

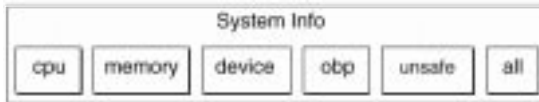


FIGURE 2-6 System Information Buttons

When you click any of these buttons, a window is displayed, and it remains open until you click the dismiss button within that window.

If you click the All button, all currently enabled windows are displayed.

▼ To Specify How Windows Are Updated

1. Click the Properties button in the Dynamic Reconfiguration window.
2. To ensure that displays are updated regularly, set the Auto Update Domain Information Displays to On (the default).

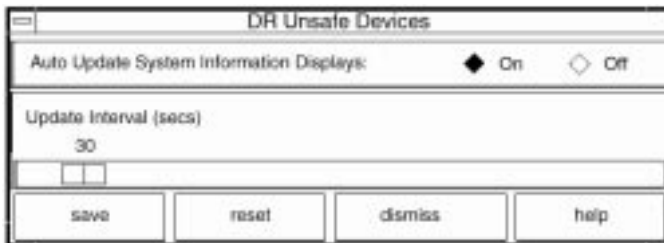


FIGURE 2-7 DR Properties Window

3. The Update Interval value (in seconds) determines how frequently the display is updated.

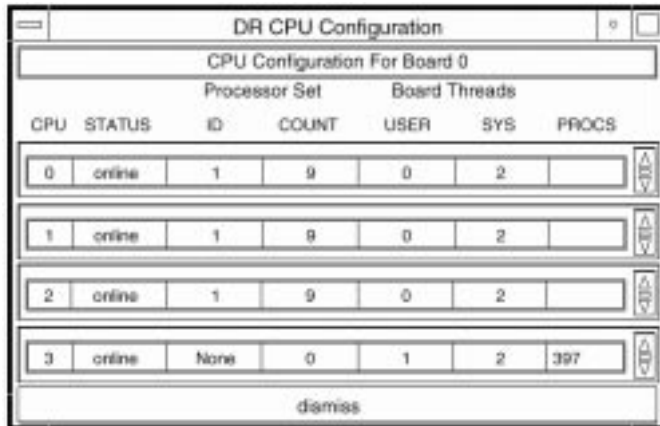
If you set Auto Update Domain Information Displays to Off, the displays are not updated; each display is a snapshot taken when the button was clicked.

4. Click the Save button to save the settings between Hostview invocations.

Note – When the update interval is set to a low value, such as 10 seconds, and several information windows are displayed, responsiveness of the DR windows may be degraded. This is especially true when the device detail windows are displayed. Each time an information window is updated, an RPC is issued to the `dr_daemon(1M)` running on the domain. The `dr_daemon` is an iterative RPC server, so each RPC request is run sequentially.

▼ To View DR CPU Configuration Information

1. Click the `cpu` button. The DR CPU Configuration window is displayed. (FIGURE 2-8)



The screenshot shows a window titled "DR CPU Configuration" with a subtitle "CPU Configuration For Board 0". Below the subtitle is a table with columns for "Processor Set" and "Board Threads". The table has a header row with columns: CPU, STATUS, ID, COUNT, USER, SYS, and PROCS. There are four data rows. The first three rows have ID 1, COUNT 9, USER 0, and SYS 2. The fourth row has ID None, COUNT 0, USER 1, and SYS 2, with a PROCS value of 397. A "dismiss" button is at the bottom.

Processor Set		Board Threads				
CPU	STATUS	ID	COUNT	USER	SYS	PROCS
0	online	1	9	0	2	
1	online	1	9	0	2	
2	online	1	9	0	2	
3	online	None	0	1	2	397

FIGURE 2-8 DR CPU Configuration Window

The DR CPU Configuration window shows specific information about each processor on the selected board.

TABLE 2-2 DR CPU Configuration Information

Heading	Description
CPU ID	Displays the ID number of the selected board.
STATUS	Displays the status of the selected board (whether the board is online or offline).
Processor Set	ID—Displays the ID number of the processor set to which the processor belongs. If the processor belongs to the default set, the word <code>none</code> is displayed in the box. COUNT—Displays the number of CPUs in the processor set.
Bound Threads	Displays the number of user and system bound threads and the process IDs of the bound threads. Some operating system device drivers may bind threads to processors to provide better servicing of a device. Threads may be bound to a processor by use of the <code>pbind(1M)</code> command.
PROCS	Displays the process IDs of the user processes that are bound to a CPU.

▼ To View DR Memory Configuration Information

1. Click the memory button.

The DR Memory Configuration window is displayed, showing three panels. (FIGURE 2-9)

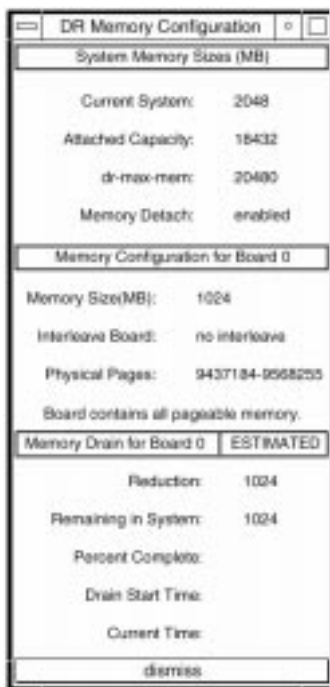


FIGURE 2-9 DR Memory Configuration Window

The DR Memory Configuration window is divided into three panels:

TABLE 2-3 DR Memory Configuration Information

System Memory Sizes (Domain Memory Information)

Current System:	Total size of memory in the domain from all boards
Attach Capacity:	Amount of memory that can be added by using the DR Attach operation
dr-max-mem:	Current value of the OBP variable dr-max-mem (for more information, refer to the <i>Sun Enterprise 10000 DR Configuration Guide</i>)

Memory Detach:

Memory Configuration for Board 0 (Board-Level Information)

Memory Size (MB):	Amount of memory on the selected board
Interleave Board:	Board that the selected board is interleaved with

TABLE 2-3 DR Memory Configuration Information (*Continued*)

Physical Pages:	Highest and lowest physical pages that are occupied by the memory on a board (Small memory areas in the middle of this range may not be used by the board. Note that DR is not able to detach boards that have interleaved memory.)
Memory Drain Information	
Reduction:	Amount of memory to be removed from domain usage when the board is detached
Remaining in System:	Domain memory size after the board is detached
Percent Complete:	How far the drain operation has progressed. Note that the time required to drain each memory page is not equal. Some memory pages take longer to drain than others.
Drain Start Time:	The time the drain operation was started.
Current Time:	The current time, which can be compared to the drain start time to see how long the drain operation has been in progress.

▼ To View DR Device Configuration Information

1. Click the device button.

The DR Device Configuration window is displayed. (FIGURE 2-10)

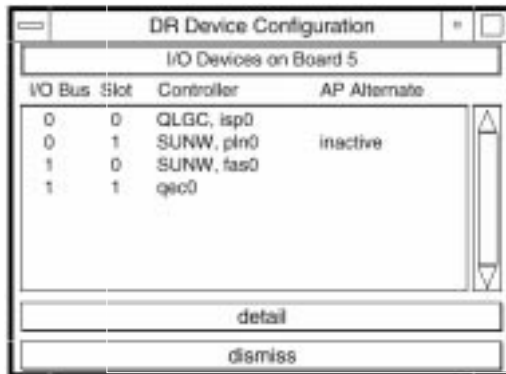


FIGURE 2-10 DR Device Configuration Window

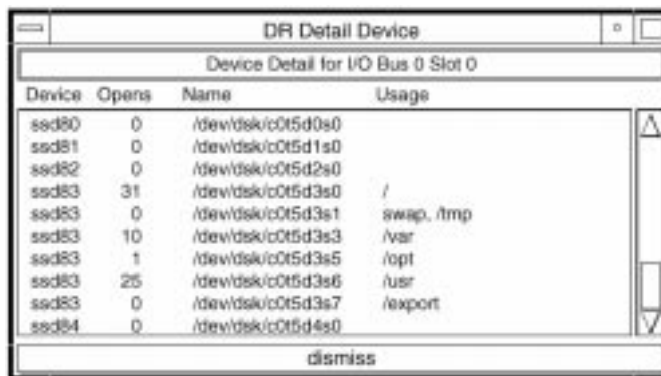
The controllers or devices in each slot are listed. The controller and device names are a concatenation of their device name and their operating environment instance number (for example, `sd31`).

Note – The DR Device Configuration window may not show all of the devices that are physically present on the board. For example, controllers whose drivers are unattached do not appear in the list. The device display that is available by using the obp button lists the cards on the board that were successfully probed and identified.

▼ To View DR Device Detailed Information

1. Highlight one or more controller(s).
2. Choose Detail.

The following window is displayed for each selected controller. (FIGURE 2-11)



The screenshot shows a window titled "DR Detail Device" with a subtitle "Device Detail for I/O Bus 0 Slot 0". It contains a table with four columns: "Device", "Opens", "Name", and "Usage". The table lists several devices with their respective open counts, names, and usage information. A "dismiss" button is located at the bottom of the window.

Device	Opens	Name	Usage
sd80	0	/dev/dsk/c0t5d0s0	
sd81	0	/dev/dsk/c0t5d1s0	
sd82	0	/dev/dsk/c0t5d2s0	
sd83	31	/dev/dsk/c0t5d3s0	/
sd83	0	/dev/dsk/c0t5d3s1	swap, /tmp
sd83	10	/dev/dsk/c0t5d3s3	/var
sd83	1	/dev/dsk/c0t5d3s5	/opt
sd83	25	/dev/dsk/c0t5d3s6	/usr
sd83	0	/dev/dsk/c0t5d3s7	/export
sd84	0	/dev/dsk/c0t5d4s0	

FIGURE 2-11 DR Detail Device Window

The current usage information for each device is shown. The window includes an open count (if available) and the common name (for example, a disk partition, a metadvice, or an interface name) by which the device is known. Additional usage information is also provided, including the partition mount points, network interface configuration, swap space usage, and metadvice usage.

Note – Some device usage, such as disk partitions used for Sun Solstice DiskSuite™ databases, Alternate Pathing (AP) databases, and Sun Enterprise Volume Manager™ usage, may not be reported.

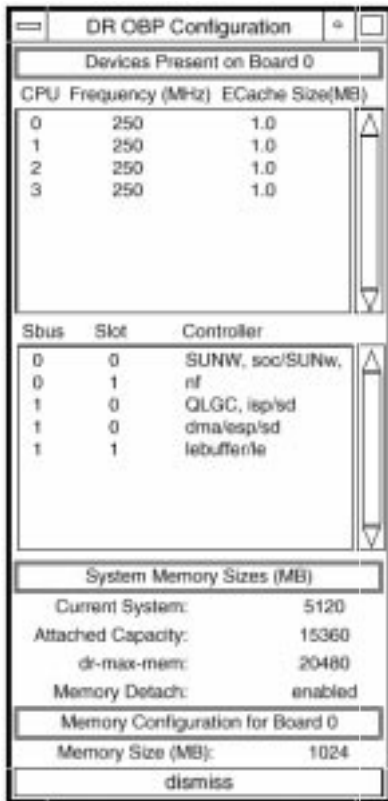
If a controller or network interface is part of the AP database, the window indicates that it is active or that it is an AP alternate. For active AP alternates, the usage of the AP metadvice is displayed.

▼ To View DR OBP Configuration Information

Note – The information in the DR OBP Configuration window is derived from the OBP device tree, and is less detailed than the information that is available from the other windows described in this section. For example, in the init attach state, only the I/O adapters are known—not the devices attached to those controllers nor the memory interleave configuration. This window is usually used when a board is in the init attach state.

1. Click the obp button.

The DR OBP Configuration window is displayed. (FIGURE 2-12)



The screenshot shows a window titled "DR OBP Configuration" with a scrollable list of hardware information for "Board 0". The window is divided into several sections:

- Devices Present on Board 0:** A table with columns for CPU ID, Frequency (MHz), and ECache Size (MB).

CPU	Frequency (MHz)	ECache Size(MB)
0	250	1.0
1	250	1.0
2	250	1.0
3	250	1.0
- Controller Information:** A table with columns for Sbus, Slot, and Controller.

Sbus	Slot	Controller
0	0	SUNW, soc/SUNw,
0	1	nf
1	0	QLGC, isp/sd
1	0	dma/esp/sd
1	1	lebuffer/le
- System Memory Sizes (MB):** A list of memory-related values.

Current System:	5120
Attached Capacity:	15360
dr-max-mem:	20480
Memory Detach:	enabled
- Memory Configuration for Board 0:** A single row showing memory size.

Memory Size (MB):	1024
-------------------	------
- Dismiss:** A button at the bottom of the window.

FIGURE 2-12 DR OBP Configuration Window

▼ To View the DR-Unsafe Devices

1. Click the unsafe button.

The DR Unsafe Devices window is displayed. (FIGURE 2-13)



FIGURE 2-13 DR Unsafe Devices Window

The DR Unsafe Devices window shows the suspend-unsafe devices that are open across the entire domain, not just those that are resident on the selected system board. This information is useful for determining the cause of operating environment quiescence errors due to unsafe devices being open.

DR 3.0 Model

This chapter contains information about the DR 3.0 model of dynamic reconfiguration (DR) on the Sun Enterprise™ 10000 system.

The DR 3.0 model is based on the use of the domain configuration server, `dcs(1M)`, to control DR operations. This model includes the Automated DR (ADR) commands, such as `addboard(1M)`, `deleteboard(1M)`, and `moveboard(1M)`. The `showusage(1M)` command is no longer supported. DR 3.0 includes three new commands:

- `showdevices(1M)`—displays the usage of devices (see Section “Showing Device Information” on page 44 for more information).
- `rcfgadm(1M)`—displays the status of attachment points on the domain.
- `cfgadm(1M)`—lists the status of the dynamically reconfigurable hardware resources on the domain (see also `cfgadm_sbd(1M)` for more information).

Automatic DR

Automatic DR enables an application to perform DR operations without requiring user interaction. This ability is provided by an enhanced DR framework that includes a reconfiguration coordination manager (RCM) and a system event facility called `sysevent`. The RCM enables application-specific loadable modules to register callbacks with the RCM. The callbacks perform preparatory tasks before a DR operation, an error recovery during a DR operation, or a clean-up after a DR operation. The `sysevent` facility enables applications to register for notification system events. The automatic DR framework interfaces with the RCM and with `sysevent` to notify applications to give up resources automatically prior to unconfiguring them, and to capture new resources as they are configured into the domain.

Note – Automatic DR is a different feature from Automated DR (ADR)

For more information about RCM, refer to the *Solaris 8 System Administration Supplement* (part number 806-7502-10) in the Solaris 8 10/01 Update Collection.

Enhanced System Availability

The DR feature enables you to hot-swap system boards without bringing the system down. DR is used to unconfigure the resources on a faulty system board from a domain so that the system board can be removed from the system. The repaired, or replacement, board can be inserted into the domain while the Solaris operating environment is running. DR then configures the resources on the board into the domain.

DR and I/O Boards

You must use caution when you add or remove system boards with I/O devices. Before you can remove a board with I/O devices, all of its devices must be closed and all its file systems must be unmounted.

If you need to remove a board with I/O devices from a domain temporarily and then re-add it before any other boards with I/O devices are added, reconfiguration unnecessary. In this case, device paths to the board devices remain unchanged. However, if you add another board with I/O devices after the first was removed, then re-add the first board; reconfiguration is required because the paths to the devices on the first board have changed.

Sun Enterprise 10000 Domains

The Sun Enterprise 10000 system can be divided into domains that contain system boards and I/O boards; and the components such as CPUs, memory chips, and CompactPCI cards that are connected to the boards. Each domain is electrically isolated into hardware partitions, which ensures that a failure in one domain does not affect the other domains in the system.

DR 3.0 Procedures

This section contains procedures that describe how to use the DR 3.0 commands. The following procedures are included:

- “Showing Platform Information” on page 43
- “Showing Device Information” on page 44
- “Adding Boards” on page 45
- “Deleting Boards” on page 45
- “Moving Boards” on page 46
- “Replacing System Boards” on page 47

Showing Platform Information

Before you attempt to add, move, or delete a board to or from a specific domain, use the `domain_status(1M)` command to determine the domain name and board number.

▼ To Show Platform Information

1. Use the `domain_status(1M)` command to obtain the domain information.

```
% domain_status -m
```

Using the `domain_status` with the `-m` option command displays the domain name, the DR model, and the number of the boards in the domain, as in the following example.

```
% domain_status -m
```

DOMAIN	TYPE	PLATFORM	DR-MODEL	OS	SYSBDS
A	Ultra-Enterprise-10000	all-A	2.0	5.8	2
B	Ultra-Enterprise-10000	all-A	3.0	5.8	3 4
C	Ultra-Enterprise-10000	all-A	2.0	5.7	5 6

Showing Device Information

Before you attempt to perform any DR operation, use the `showdevices(1M)` command to display the device information, especially when removing devices.

▼ To Show Device Information

1. Use the `showdevices(1M)` command to display the device information for a domain.

```
% showdevices -v -d A
```

The above command displays the device information for all of the CPUs in domain A. Refer to the `showdevices(1M)` man page to learn how to display device-specific information.

```
CPU
----
domain  board  id   state  speed  ecache  usage
A       C1     40  online  400    4
A       C1     41  online  400    4
A       C1     42  online  400    4
A       C1     43  online  400    4
A       C2     55  online  400    4
A       C2     56  online  400    4
A       C2     57  online  400    4
A       C2     58  online  400    4
```

The following output is an example of the memory output for the `showdevices(1M)` command.

```
Memory
drain in progress:
-----
domain  board  board  perm  base  domain  target  deleted  remaining
        board  mem MB  mem MB  addr  mem MB  board  MB      MB
A       C1     2048  933   0x600000 4096  C2     250    1500
A       C2     2048  0     0x200000 4096
```

The following is an example of the I/O devices output for the `showdevices(1M)` command.

```
IO Devices
-----
domain  board  device  resource          usage
A       I01    sd0
A       I01    sd1
A       I01    sd2
A       I01    sd3      /dev/dsk/c0t3d0s0  mounted filesystem "/"
A       I01    sd3      /dev/dsk/c0t3d0s1  dump device (swap)
A       I01    sd3      /dev/dsk/c0t3d0s1  swap area
A       I01    sd3      /dev/dsk/c0t3d0s3  mounted filesystem "/var"
A       I01    sd3      /var/run           mounted filesystem "/var/run"
A       I01    sd4
A       I01    sd5
```

Refer to the `showdevices(1M)` man page for a complete list of the options and arguments for this command.

Adding Boards

Adding a board to a domain moves the board through several state changes. First the board is connected to the domain, and then it is configured into the Solaris operating environment. After the board is connected, it is considered to be part of the physical domain and available for use by the operating system.

▼ To Add a Board to a Domain

1. Use the `addboard(1M)` command to add the board to the domain.

The following example shows how the `addboard(1M)` command adds system board 2 to the domain specified by `domain_id`. Two retries are performed, if necessary, with a wait time of 10 minutes between retries.

```
% addboard -d domain_id -r 2 -t 600 SB2
```

Deleting Boards

Deleting a board from a domain removes the board from the domain.

Always check the usage of the components on a board before you delete it from a domain. If the board hosts permanent memory, the memory is moved to another board within the same domain before the board is deleted from the domain. Likewise, if any busy devices are present, you must wait or ensure that the device is no longer being used by the system before you attempt to remove the board.



Caution – You must use the `power` command to power off the board before physically removing it from the system. The `deleteboard(1M)` command does not power off the board. Refer to the `power(1M)` man page for information about the `power` command. Or see the section “To Physically Replace a System Board” on page 47. In addition, the *Sun Enterprise 10000 Systems Service Manual* contains complete information about physically removing and replacing boards.

▼ To Delete a Board From a Domain

1. Use the `deleteboard(1M)` command to delete the board from the domain.

The following example of the `deleteboard(1M)` command deletes system board 2 from its current domain. Two retries are performed, if necessary, with a wait time of 15 minutes between retries.

```
% deleteboard -r 2 -t 900 SB2
```

Moving Boards

Moving a board from one domain to another domain involves removing the board from the first domain; and then connecting and configuring it into the target domain.

Always check memory usage on a board and the devices that are connected to it before moving it out of a domain. If the board hosts permanent memory, the memory must be moved to another board within the same domain before the board can be moved to another domain. Likewise, if any busy devices are present, you must wait until the device is no longer being used by the system before you attempt to move the board.

▼ To Move a Board

1. Use the `moveboard(1M)` command to move the board from one domain to another domain.

The following example of the `moveboard(1M)` command moves system board 2 from its current domain to the domain specified by `domain_id`. Two retries are performed, if necessary, with a wait time of 15 minutes between retries.

```
% moveboard -d domain_id -r 2 -t 900 SB2
```

Replacing System Boards

This section describes how to physically replace a board in a domain by using the commands described in this chapter.

▼ To Physically Replace a System Board

In the following steps, system board 2 is removed from its current domain and replaced by system board 3.

1. Delete the board from the domain.

```
% deleteboard -r 2 -t 900 SB2
```

2. Power off system board 2.

Refer to the `power(1M)` man page for information about the `power` command.

```
% power -off -sb 2
```



Caution – For complete information about physically removing and replacing boards, refer to the *Sun Enterprise 10000 Systems Service Manual*. Failure to follow the stated procedures can result in damage to system boards and other components.

3. Physically remove system board 2 and replace it with system board 3.
4. Power on system board 3.

```
% power -on -sb 3
```

5. Add system board 3 to the domain.

```
% addboard -r 2 -t 900 SB3
```

DR Error Messages

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

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

- SSP applications
- System error logs

Searching This Appendix

Before you use this appendix, be sure to read the following list of search tips.

- Search on a specific string of text in the error message.
- Avoid using numeric values. They are treated as replaceable text in this appendix.
- Avoid using text that is replaceable. In this appendix, 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*.
- The tables are presented by the type of error or failure. The contents of the tables are sorted alphabetically in descending order.

Error-Type Links

The following are different types of errors:

SSP Errors

Use one of the following links to start a search of SSP-related error messages:

[“Protocol and Communication Error Messages” on page 51](#)

[“Attach-Related Error Messages” on page 56](#)

[“Detach-Related Error Messages” on page 57](#)

[“Auto-Configuration Error Messages” on page 62](#)

Domain Errors

Use one of the following links to start a search of domain-related error messages:

[“DR Daemon Start-Up Error Messages” on page 64](#)

[“Memory Allocation Error Messages” on page 68](#)

[“DR Driver Error Messages” on page 65](#)

[“Platform Specific Module \(PSM\) Error Messages” on page 75](#)

[“DR General Domain Error Messages” on page 77](#)

[“DR Domain Exploration Error Messages” on page 82](#)

[“OpenBoot PROM Error Messages” on page 79](#)

[“Unsafe-Device Query Error Messages” on page 94](#)

[“Alternate Pathing \(AP\) Error Messages” on page 96](#)

[“DCS Error Messages” on page 98](#)

[“DR Driver Error Messages” on page 101](#)

[“Plugin Error Messages” on page 107](#)

SSP Error Messages

The following sections contain SSP-related error messages:

Protocol and Communication Error Messages

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

TABLE 4-1 Protocol and Communication Failure Error Messages

Error Message	Probable Cause	Suggested Action
NGDR Error: abort_attach_board: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, the error indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.
NGDR Error: abort_detach_board: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.
NGDR Error: attach_finished: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.

TABLE 4-1 Protocol and Communication Failure Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
NGDR Error: complete_attach_board: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.
NGDR Error: cpu0_move_finished: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.
NGDR Error: detach_board: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.

TABLE 4-1 Protocol and Communication Failure Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
NGDR Error: detach_finished: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.
NGDR Error: detachable_board: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.
NGDR Error: drain_board_resources: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.

TABLE 4-1 Protocol and Communication Failure Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
<p>NGDR Error: get_board_config: invalid board number</p>	<p>The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input or catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.</p>	<p>Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.</p>
<p>NGDR Error: get_board_state: invalid board number</p>	<p>The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.</p>	<p>Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.</p>
<p>NGDR Error: get_cpu_info: invalid board number</p>	<p>The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.</p>	<p>Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.</p>

TABLE 4-1 Protocol and Communication Failure Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
NGDR Error: get_obp_board_config: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.
NGDR Error: initiate_attach_board: invalid board number	The RPC is attempting to perform a DR operation on a board number that is not in the range of valid numbers. The DR applications carefully filter the user input to catch out-of-range board numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.
NGDR Error: initiate_attach_board: invalid cpu number	The RPC is attempting to perform a initiate an attach of a board that contains a CPU that is not on the board. The DR applications carefully filter the user input or catch invalid CPU numbers before they send the RPC. Therefore, this error indicates a breakdown on the SSP or in the network connection to the SSP. Or, it indicates an incompatibility between the SSP applications and the DR daemon.	Check the SSP network connection and/or the SSP and DR applications to ensure that they are operating properly.
NGDR Error: Unauthorized RPC call . . . Not owner	The DR daemon received an RPC that failed authentication.	Check the system log for more information about this error. Also, make sure that the version numbers match for the SSP and the DR daemon and that the SSP user and network services are properly configured.

Attach-Related Error Messages

The following table contains attach-related failure error messages that are sent to the system logs and/or the SSP applications.

TABLE 4-2 Attach-Related Failure Error Messages

Error Message	Probable Cause	Suggested Action
NGDR Error: abort_attach_board: invalid board state	The attach operation could not be aborted because the board is not in the init_attach state, waiting to be configured into the domain.	Wait for the board to enter the init_attach state before aborting the attach operation.
NGDR Error: attach_finished: invalid board state	Communication protocol has been breached during the attach operation. The DR driver and daemon disagree with the SSP that the board was waiting for the confirmation of the attach operation from the SSP.	Exit and restart the current DR application, then retry the operation. If this error persists, stop and restart the DR daemon. You may need to reboot the domain to recover from this error.
NGDR Error: Cannot abort attach. Board ineligible for further DR operations.	The board entered the FATAL state after the abort command was issued, causing the abort operation to fail and the board to be lost from the system.	Reboot the domain.
dr_attach: failure executing A3000 hot_add script . . . <i>error message</i>	The Sun StoreEdge™ A3000 hot_add script is executed directly after a DR Attach operation. If the script exists, but it cannot be executed, the error message explains why.	If you are not using, nor plan to use, A3000 devices, you can rename the script so that it will not be found.
initiate_attach_board: already init_attached	You tried to initiate the attach of a board that was already initiated.	Open the Complete Attach window and continue the attach process.
NGDR Error: complete_attach_board: invalid board state	You tried to initiate an attach operation on a board that is not eligible—the board is not in the init_attach state awaiting attachment to the domain.	Wait for the board to enter the init_attach state before aborting the attach operation.
NGDR Error: initiate_attach_board: invalid board state	You tried to initiate an attach operation on a board that is not eligible—the board is not in the PRESENT state awaiting attachment to the domain.	Wait for the board to enter the init_attach state before aborting the attach operation.

TABLE 4-2 Attach-Related Failure Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
<p>NGDR Error: Some devices not attached. Examine the host syslog for details . . . <i>errno_description</i></p>	<p>Some devices were not configured into the domain.</p>	<p>Look at the system logs for more details about what devices were not configured into the domain and why they were not configured. Some devices on the board may not be supported by the operating environment or by the DR feature. Blacklist unsupported devices.</p>

Detach-Related Error Messages

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

TABLE 4-3 Detach-Related Failure Error Messages

Error Message	Probable Cause	Suggested Action
<p>NGDR Error: Cannot detach board <i>board_number</i>. It has <i>interface_name</i> interfaces configured.</p>	<p>The board is not eligible to be detached because it has one or more network interfaces attached to it that are critical to the operation of the domain. The network interfaces can be any mix of primary, SSP, AP, or PBF interfaces.</p>	<p>Use the <code>ifconfig(1M)</code> command to determine the role of the interface(s). If the configured interface is the primary network or the SSP, manually switch the interface to the alternate interface if one exists. For an interface other than the primary and the SSP, unplumbing it may enable the detach operation to succeed. Otherwise, the domain must be shut down, and the interfaces moved to another board.</p>
<p>NGDR Error: <code>cpu0_move_finished: invalid board state</code></p>	<p>Communication protocol has been breached over the eligibility of a CPU. To the SSP, the CPU has been moved off the board. To the DR driver, the move operation is an invalid operation for that board.</p>	<p>None</p>

TABLE 4-3 Detach-Related Failure Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
ifconfig down failed.	The <code>ifconfig(1M)</code> command failed to bring down the network interfaces. The <code>ifconfig(1M)</code> command unplumbs and brings down the network interfaces before the board is detached. One of the network interfaces on the board could be busy, so manual intervention may be needed.	Log in to the domain and, if possible, bring down the network interfaces on the board manually by using the <code>ifconfig(1M)</code> command with the <code>down</code> option. The manual execution of the command may yield more detailed information about the failure.
ifconfig unplumb failed.	The <code>ifconfig(1M)</code> command failed to unplumb the network interfaces. The <code>ifconfig(1M)</code> command unplumbs and brings down the network interfaces before the board is detached. One of the network interfaces on the board could be busy, so manual intervention may be needed.	Log in to the domain and, if possible, unplumb the network interfaces manually by using the <code>ifconfig(1M)</code> command with the <code>unplumb</code> option. The manual execution of the command may yield more detailed information about the failure.
Warning: Error return from /opt/SUNWconn/bin/nf_snmd_kill (<i>return_value</i>)	The command failed. Certain daemons keep network interfaces open continuously. These daemons must be stopped before the devices they control can be detached.	Analyze the <i>return_value</i> to determine why the <code>kill(1)</code> command failed, and try to correct the problem. If necessary, use the <code>ps(1)</code> command to obtain the PID number for the daemons. Use the <code>kill(1)</code> command to stop the daemons manually.
Warning: Error return from /opt/SUNWconn/bin/pf_snmd_kill (<i>return_value</i>)	The <code>kill(1)</code> command failed. The daemons that are used to control certain network devices must be stopped before the devices can be detached because the daemons keep the interfaces open continually.	Analyze the <i>return_value</i> to determine why the <code>kill</code> command failed, and try to correct the problem. If necessary, use the <code>ps(1)</code> command to obtain the PID number for the daemons. Use the <code>kill(1)</code> command to stop the daemons manually.
NGDR Error: abort_detach: board already drained	The <code>cancel ioctl()</code> failed while the DR daemon was trying to abort the detach operation. The failure caused the board to be reported as being in the unreferenced state, indicating that the memory has already been drained.	The board must be completely detached before you can recover from this error. Retry the DR operation after the board has been successfully detached.

TABLE 4-3 Detach-Related Failure Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
<p>NGDR Error: abort_detach_board: invalid board state</p>	<p>Communication protocol has been breached over the eligibility of a board. To the SSP, the board is part of the domain and has been, or is being, drained of its resources. The SSP, therefore, issues the abort command to stop the detach operation. However, to the DR driver and daemon, the board is not part of the domain.</p>	<p>Exit and restart the DR application.</p>
<p>NGDR Error: board configuration query failed.</p>	<p>The DR daemon failed to determine the eligibility of the configuration of the board.</p>	<p>Stop and restart the DR daemon and/or the DR driver. If this error persists, use the <code>modinfo(1M)</code>, <code>modload(1M)</code>, and <code>modunload(1M)</code> commands to work with the driver after you have stopped the DR daemon. Also, check the size of the DR daemon with the <code>ps(1)</code> command. If the size is not between 300 and 400 Kbytes, report this error, providing as much information from the system logs as possible.</p>
<p>NGDR Error: Cannot abort detach. Board detached from OS (detach completed).</p>	<p>This message indicates that the detach operation has completed, and follows the message that is displayed for the NGDR Error: <code>abort_detach: board already drained</code> error message.</p>	<p>See the NGDR Error: <code>abort_detach: board already drained</code> message.</p>

TABLE 4-3 Detach-Related Failure Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
<p>NGDR Error: couldn't query cpu configuration</p>	<p>The complete_detach operation has failed because the DR daemon could not ascertain the CPU configuration just prior to the beginning of the complete_detach operation. After a board is detached, the DR daemon uses the information about the CPU configuration to update the utmp and wtmp entries for each CPU on the board. Although the complete_attach operation does not depend on the updates, if the mechanism through which the CPU configuration is queried is broken, serious problems exist. Do not proceed with the detach operation.</p>	<p>Stop and restart the DR daemon and/or the DR driver. Also, check the size of the DR daemon with the ps(1) command. If it is not between 300 and 400 Kbytes, report this error, providing as much information from the system logs as possible.</p>
<p>NGDR Error: detach_board: invalid board state</p>	<p>Communication protocol has been breached over the eligibility of a board. To the SSP, the board is part of the domain, and its resources have been drained, causing the SSP to attempt to complete the detach operation. However, to the DR driver and daemon, the board is not part of the domain.</p>	<p>Examine the state of the board by using the showdevices(1m) command, and determine the cause of the problem. Retry the drain and/or complete_detach operations to determine if the error is recoverable. Stop and start the DR daemon and driver.</p>
<p>NGDR Error: detach_board: invalid board state</p>	<p>The proper sequence of board states has not been followed, meaning that the board went into the error state or that an earlier failure in the drain-detach sequence of events was not properly reported.</p>	<p>Examine the state of the board by using the showdevices(1m) command, and determine the cause of the problem. Retry the drain and/or complete_detach operations to determine if the error is recoverable. Stop and start the DR daemon and driver.</p>
<p>NGDR Error: detach_finished: invalid board state</p>	<p>Communication protocol has been breached over the eligibility of a board. To the SSP, the board has been detached. However, to the DR driver and daemon, the board has not been detached from the domain.</p>	<p>Examine the state of the board by using the showdevices(1m) command, and determine the cause of the problem. Retry the drain and/or complete_detach operations to determine if the error is recoverable. Stop and start the DR daemon and driver.</p>

TABLE 4-3 Detach-Related Failure Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
<p>NGDR Error: detachable_board: invalid board state</p>	<p>Communication protocol has been breached over the eligibility of a board. To the SSP, the board is part of the domain, so the SSP attempts to drain the resources. However, to the DR driver and daemon, the board is not part of the domain.</p>	<p>Examine the state of the board by using the <code>showdevices(1m)</code> command, and determine the cause of the problem. Retry the drain and/or <code>complete_detach</code> operations to determine if the error is recoverable. Stop and start the DR daemon and driver.</p>
<p>NGDR Error: detaching board would leave no online CPUs</p>	<p>The detach operation failed because no CPUs would be left online after the board is detached.</p>	<p>Bring more CPUs on other boards online in the domain, or add more boards with online CPUs to the domain, so that the domain has enough online CPUs after the board is detached.</p>
<p>NGDR Error: drain_board_resources: invalid board state</p>	<p>Communication protocol has been breached over the eligibility of a board. To the SSP, the board is part of the domain, so the SSP attempts to drain the resources. However, to the DR driver and daemon, the board is not part of the domain.</p>	<p>Examine the state of the board by using the <code>showdevices(1m)</code> command, and determine the cause of the problem. Retry the drain and/or <code>complete_detach</code> operations to determine if the error is recoverable. Stop and start the DR daemon and driver.</p>
<p>NGDR Error: Remaining system memory (<i>memory_size</i> mb) below minimum threshold (<i>minimum_memory_size</i> mb)Not enough space</p>	<p>The domain must have enough memory to accommodate the memory of the board that is being detached. The detach operation failed because the domain does not have enough memory to detach the board.</p>	<p>Attach as many boards as necessary so that the memory in the domain can hold the memory of the board being detached.</p>
<p>NGDR Error: Some devices not re-attached. Examine the host syslog for details <i>errno_description</i></p>	<p>Devices could not be reattached to the operating environment during an abort detach operation. Errors were encountered while the DR daemon tried to communicate with the device drivers for one or more devices on the board.</p>	<p>Examine the system logs to determine which devices were not reattached. If possible, fix the problem and then issue the <code>complete_attach(1M)</code> command again to fully configure the board. If this action fails, the failure may be caused by an unsupported device for which a state cannot be resolved until the domain is rebooted.</p>

TABLE 4-3 Detach-Related Failure Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
NGDR Error: sysconf failed (_SC_NPROCESSORS_ONLN) . . . <i>errno_description</i>	The <code>sysconf(3c)</code> system call failed to return the total number of online CPUs in the domain. Thus, the DR daemon cannot determine if the domain would be left with any online CPUs after the board is detached.	See the <code>sysconf(3c)</code> man page for more details about this error. Use those details and the <i>errno_description</i> to diagnose and solve the error. Retry the DR operation after you have solved the error. If no fix is apparent, stop and restart the DR daemon, then retry the DR operation.

Auto-Configuration Error Messages

The following table contains the list of auto-configuration error messages that are sent to the system logs and/or to the SSP applications.

TABLE 4-4 Auto-Configuration Error Messages

Error Message	Probable Cause	Suggested Action
NGDR Error: Complete pending DR operation prior to running <code>autoconfig</code> . . . Invalid argument	The <code>autoconfig(1M)</code> command failed because a DR operation was still pending (that is, the board was not fully detached or attached before you issued the <code>autoconfig(1M)</code> command to reconfigure the operating environment).	Use the <code>showdevices(1M)</code> command to determine the state of the board. Decide to abort or complete the pending operation before you try to use the <code>autoconfig(1M)</code> command to reconfigure the operating environment.
NGDR Error: Could not get /tmp/AdDrEm.lck lock . . . <i>errno_description</i>	The DR daemon failed to get the lock it needs to reconfigure the operating environment.	Check the additional <i>errno_description</i> and/or error number that is sent with the error message to determine why the lock could not be acquired.
NGDR Error: Could not unlock /tmp/AdDrEm.lck lock . . . <i>errno_description</i>	The DR daemon could not release the lock.	Check the additional <i>errno_description</i> and/or error number that is sent with the error message to determine why the lock was not released.

TABLE 4-4 Auto-Configuration Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
NGDR Error: devlinks cmd failed . . . <i>error descriptions</i>	The devlinks(1M) command failed to reconfigure the operating environment.	Check the additional <i>error descriptions</i> and/or error number that is sent with the error message to determine why the command failed. Manually run the command on the domain.
NGDR Error: disks cmd failed . . . <i>error descriptions</i>	The disks(1M) command failed to reconfigure the operating environment.	Check the additional <i>error descriptions</i> and/or error number that is sent with the error message to determine why the command failed. Manually run the command on the domain.
NGDR Error: drvconfig cmd failed. . . <i>error description</i>	The drvconfig(1M) command failed to reconfigure the operating environment.	Check the additional <i>error description</i> and/or error number that is sent with the error message to determine why the command failed. Manually run the command on the domain.
NGDR Error: ports cmd failed . . . <i>error description</i>	The ports(1M) command failed to reconfigure the operating environment.	Check the additional <i>error description</i> and/or error number that is sent with the error message to determine why the command failed. Manually run the command on the domain.
NGDR Error: sync cmd failed . . . <i>error description</i>	The sync(1M) command failed to reconfigure the operating environment.	Check the additional <i>error description</i> and/or error number that is sent with the error message to determine why the command failed. Manually run the command on the domain.
NGDR Error: tapes cmd failed . . . <i>error descriptions</i>	The tapes(1M) command failed to reconfigure the operating environment.	Check the additional <i>error description</i> and/or error number that is sent with the error message to determine why the command failed. Manually run the command on the domain.

Domain Error Messages

DR Daemon Start-Up Error Messages

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

TABLE 4-5 DR Daemon Start-Up Error Messages

Error Message	Probable Cause	Suggested Action
Cannot create server handle	The DR daemon could not start 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 <code>inetd.conf</code> entry for the DR daemon.
Cannot fork: <i>descriptive message</i>	The DR daemon could not fork a process from which to run its RPC server.	The <i>descriptive error</i> message corresponds to an <i>errno_value</i> 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.
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 <code>inetd.conf</code> entry for the DR daemon.

TABLE 4-5 DR Daemon Start-Up Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
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 <code>inetd.conf</code> entry for the DR daemon.

DR Driver Error Messages

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 4-6 DR Driver Error Messages

Error Message	Probable Cause	Suggested Action
<p>DR: Error: initiate_attach: ioctl failed</p> <p>DR: Error: complete_attach: ioctl failed</p> <p>DR: Error: abort_attach: ioctl failed</p> <p>DR: Error: get_cpu_info: ioctl failed</p> <p>DR: Error: get_mem_config: ioctl failed</p>	<p>An <code>ioctl()</code> 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.</p> <p>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 <code>/devices/pseudo</code> directory could be missing, or the file permissions could be wrong. The DR daemon could also be experiencing memory corruption or resource limitations. The <code>ioctl()</code> failure message is followed by a message in the form: <i>Daemon (errno) #error_number): error description.</i></p>	<p>The context of the <code>ioctl()</code> failure (that is, which function precedes the <code>ioctl()</code> 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 <code>ioctl(2)</code> man page. You can also use the <code>/usr/include/errno.h</code> header file if the <code>ioctl(2)</code> man page does not have a specific reference for the error number.</p>

TABLE 4-6 DR Driver Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
DR: Error: get_mem_cost : 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): <i>errno_description</i> .	The context of the ioctl() failure (that is, which function proceeds 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.
DR: Error: get_mem_drain: ioctl failed		
DR: Error: update_attach: ioctl failed		
DR: Error: ioctl failed, error draining resources	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): <i>errno_description</i> .	
DR: Error: detach_board : UNCONFIGURE ioctl failed		
DR: Error: detach_board : DISCONNECT ioctl failed		
DR: Error: abort_detach : CANCEL ioctl failed		
DR: Error: abort_detach : CONFIGURE ioctl failed		
DR: Error: get_dr_state : ioctl failed	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.	
DR: Error: get_dr_status: ioctl failed		

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 describes one of two possible errors: ENOMEM or EAGAIN. All of the ENOMEM errors have the same suggested action, as do the EAGAIN errors.

TABLE 4-7 Memory Allocation Error Messages

Error Message	Probable Cause	Suggested Action
NGDR Error: malloc failed (add notnet ap info) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.
NGDR Error: malloc failed (alias_namelen) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.

TABLE 4-7 Memory Allocation Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
NGDR Error: malloc failed (AP ctrl_t array) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.
NGDR Error: malloc failed (ap_controller) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.
NGDR Error: malloc failed (board_cpu_config_t) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.

TABLE 4-7 Memory Allocation Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
NGDR Error: malloc failed (board_mem_config_t) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.
NGDR Error: malloc failed (board_mem_cost_t) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.
NGDR Error: malloc failed (board_mem_drain_t) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.

TABLE 4-7 Memory Allocation Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
<p>NGDR Error: malloc failed (dr_io) <i>errno_description</i></p>	<p>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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.</p>	<p>First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.</p>
<p>NGDR Error: malloc failed (leaf array) <i>errno_description</i></p>	<p>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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.</p>	<p>First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.</p>
<p>NGDR Error: malloc failed (leaf) <i>errno_description</i></p>	<p>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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.</p>	<p>First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.</p>

TABLE 4-7 Memory Allocation Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
<p>NGDR Error: malloc failed (net_leaf_array) <i>errno_description</i></p>	<p>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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.</p>	<p>First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.</p>
<p>NGDR Error: malloc failed (sbus_cntrl_t) <i>errno_description</i></p>	<p>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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.</p>	<p>First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.</p>
<p>NGDR Error: malloc failed (sbus_config) <i>errno_description</i></p>	<p>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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.</p>	<p>First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.</p>

TABLE 4-7 Memory Allocation Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
NGDR Error: malloc failed (sbus_device_t) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.
NGDR Error: malloc failed (sbus_usage_t) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.
NGDR Error: malloc failed (struct devnm) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.

TABLE 4-7 Memory Allocation Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
NGDR Error: malloc failed (swap name entries) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.
NGDR Error: malloc failed (swaptbl) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.
NGDR Error: malloc failed (unsafe_devs) <i>errno_description</i>	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 <i>errno_description</i> usually describes an ENOMEM or EAGAIN error.	First, check the size of the daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon is larger than 400 Kbytes, there may be a memory leak. If there is, 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. Retry the operation, which may succeed eventually, or stop and restart the daemon.

Platform Specific Module (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 4-8 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 <code>kill(1)</code> command.
6 SFDR_ERR_RTTHREAD	Realtime thread could not be stopped.	Retry the operation. If this error persists, try stopping the process with the <code>kill(1)</code> command.
7 SFDR_ERR_KTHREAD	Kernel thread could not be stopped.	Retry the operation. If this error persists, try stopping the process with the <code>kill(1)</code> command.
8 SFDR_ERR_OSFAILURE	The kernel is not processing DR operations properly for the DR driver.	None
9 SFDR_ERR_OUTSTANDING	The <code>ioctl()</code> 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 run.	Check the <code>/etc/system</code> file to ensure that memory detach is enabled.
12 SFDR_ERR_NOMEM	Not enough memory	None
13 SFDR_ERR_PROTO	Protocol failure	None

TABLE 4-8 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 example, 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_INVALID	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 bring 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 Error Messages

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 4-9 DR General Domain Error Messages

Error Message	Probable Cause	Suggested Action
NGDR Error: Cannot fork() process . . . <i>errno_description</i>	The DR daemon could not fork off a process for the command to run in. A message in the form <code>running command</code> appears in the system logs prior to this error message, or any other error message about failed commands.	The <i>errno_description</i> 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: <i>command</i> has continued	While the DR daemon was running external commands, one of the commands failed or exited abnormally. DR executes external commands (for example, <code>drvconf</code>) 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: <i>command</i> stopped by signal <i>signal_number</i>	While the DR daemon was running external commands, one of the commands failed or exited abnormally. DR executes external commands (for example, <code>drvconf</code>) 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: <i>command</i> terminated due to signal <i>signal_number</i>	While the DR daemon was running external commands, one of the commands failed or exited abnormally. DR executes external commands (for example, <code>drvconf</code>) 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: <i>command</i> terminated due to signal <i>signal_number</i> . Core dumped.	While the DR daemon was running external commands, one of the commands failed or exited abnormally. DR executes external commands (for example, <code>drvconf</code>) 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.
NGDR Error: <code>dr_issue_ioctl:</code> failed closing driver . . . <i>errno_description</i>	The DR daemon encountered a failure while it tried to close a DR driver entry point. A detailed explanation of this failure accompanies the error message.	Use the <code>close(2)</code> man page and the <i>errno_description</i> to determine what caused this error and how to solve it.

TABLE 4-9 DR General Domain Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
Cannot exec command (errno = <i>errno_value</i>).	The DR daemon could not execute the external command. A detailed explanation of this failure accompanies the error message.	Check the system logs to determine which command failed. See the <code>exec(2)</code> man page for more details about the specified <i>errno_value</i> . 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.	Gather as much information about this problem as possible from the system logs to determine the cause of the failure. Try stopping and restarting 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 occurred, causing it to call its internal routines with incorrect values.	Gather as much information about this problem as possible from the system logs to 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 <i>board_number</i> '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.

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 4-10 OpenBoot PROM 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 4-10 OpenBoot PROM 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.
NGDR Error: cannot open /dev/openprom. . . <i>errno_description</i>	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 <code>open(2)</code> man page and the <i>errno_description</i> . 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 to 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 4-10 OpenBoot PROM Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
NGDR 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 preceded this error message. Fix the error if possible.
NGDR 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 <code>ps(1M)</code> command. Stop any processes that are keeping the entry point open.
NGDR 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
NGDR 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.
NGDR Error: OPROMCHILD. . . <i>errno_description</i>	An <code>ioctlr()</code> performed on the OBP driver entry point failed, specifically the <code>ioctlr()</code> used to traverse 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 <i>errno_value</i> or the <i>errno_description</i> that accompanies this error message. Fix the error if possible.
NGDR Error: OPROMGETPROP. . . <i>errno_description</i>	An <code>ioct1()</code> performed on the OBP driver entry point failed, specifically the <code>ioct1()</code> 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 <code>ioct1(2)</code> man page and the <i>errno_description</i> that accompanies this error message. Fix the error if possible.

TABLE 4-10 OpenBoot PROM Error Messages (*Continued*)

Error Message	Probable Cause	Suggested Action
NGDR Error: OPROMNEXT. . . <i>errno_description</i>	An <code>ioctl()</code> performed on the OBP driver entry point failed, specifically the <code>ioctr()</code> used to traverse 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 <code>ioctl(2)</code> man page and the <i>errno_description</i> that accompanies this error message. Fix the error if possible.
NGDR Error: System architecture does not support this option of this command.	An unsupported option was given to the DR daemon as it traversed 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

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 4-11 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 . . . <i>errno_description</i>	The DR daemon made an incorrect decision about the detachability and usage of devices in the domain. It is a non-fatal error.	Analyze this error by using the <i>errno_description</i> , 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 (<i>errno=errno_value</i>)	The DR daemon does not allow a detachability test to pass if the <code>mnttab</code> 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 this error by using the <i>errno_value</i> , and try to correct the error. 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.

TABLE 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
<p>Cannot open socket (errno=<i>errno_value</i>)</p> <p>This error message is sent only to the system logs.</p>	<p>The DR daemon could not open a network device. All network devices are opened to test their usage.</p>	<p>Analyze this error by using the <i>errno_value</i>. 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 greater than 400 Kbytes, 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.</p>
<p>get_cpu_bindings: can't access /proc filesystem [non-fatal].</p>	<p>The /proc file system cannot be opened. When the DR daemon explores the domain to determine CPU information for a board, the /proc file system 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.</p>	<p>Check to see why the /proc file system 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.</p>
<p>get_mem_config: couldn't determine total system memory size; only 1 board counted [non-fatal].</p>	<p>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 <code>drshow board_number mem</code> command is inaccurate. The inaccuracy also negatively affects the eligibility of a board for a detach operation. If the total memory cannot be calculated, then the effects of removing a board from the domain cannot be calculated as well.</p>	<p>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 occurred over time. Check the size of the DR daemon by using the <code>ps(1)</code> 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.</p>
<p>get_net_config_info: interface_name no address (errno=<i>errno_value</i>)</p>	<p>The DR daemon encountered a failure while trying to obtain information about a network interface that was configured by using the <code>ifconfig(1M)</code> command.</p>	<p>Determine what caused this error by using the <i>errno_value</i>, then correct the error.</p>

TABLE 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
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 <i>interface_name</i> not found (h_errno= <i>errno_value</i>)	The file needed to test each active network device may not exist, or 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 <i>errno_value</i> to determine if the file exists or if it is corrupted, and correct the error as necessary. The file is named <code>/etc/hostname.<i>interface_name</i></code> , where <i>interface_name</i> is the interface named in the error message.
Host address field for <i>interface_name</i> is null!!	The IP address for the primary interface (<i>interface_name</i>) 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 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
Host address for <i>interface_name</i> must be internet address.	The file 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 <code>/etc/hostname.<i>interface_name</i></code> , where <i>interface_name</i> is the interface named in the error message.
I/O bus device tree not built.	This error message provides information about the NGDR Error: device tree not built error message, in which the libdevinfo API failed to build the device tree for the system board.	See the NGDR Error: device tree not built error message.
minor_walk: failed to build net leaf.	This error message provides information about the NGDR 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 NGDR Error: device tree not built error message.
minor_walk: failed to build non-net leaf.	This error message provides 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 NGDR Error: I/O bus device tree not built error message.
Partition <i>partition_name</i> 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 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
Recursive symlink found ' <i>symbolic_link_name</i> '. Please remove it.	The DR daemon found a symbolic link as it traversed 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= <i>errno_value</i>)	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 <i>errno_value</i> , and try to correct it. Use the swapctl(2) man page and the <i>errno_value</i> 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 <i>errno_value</i>	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 <i>errno_value</i> , then correct the error.

TABLE 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
Unable to get swap entries (<i>errno=errno_value</i>)	The <code>swapctl(2)</code> 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 swap partitions cannot be determined.	Analyze what caused this error by using the <code>swapctl(2)</code> man page and the <i>errno_value</i> , 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 <i>devlink_file</i> <i>errno_value</i>	The <code>lstat(2)</code> system call failed when it encountered the <i>devlink_file</i> , where <i>devlink</i> is the name of the symbolic link in the <code>/dev</code> directory.	Determine what caused this error by using the <code>lstat(2)</code> man page and the <i>errno_value</i> . 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 open <i>hostname_file</i> (<i>errno=errno_value</i>)	The information 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 <code>open(2)</code> man page and the <i>errno_value</i> , and try to correct it. Look for incorrect file permissions or non-existent files. The <i>hostname_file</i> value consists of a file named <code>/etc/hostname.ifname</code> , where <i>ifname</i> is a device name, such as <code>hme0</code> or <code>le0</code> .

TABLE 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
Unable to read host name from <code>hostname_file</code>	The file 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 is not corrupt.
Unable to readlink <code>devlink_file</code> <code>errno_value</code>	The <code>readlink(2)</code> system call failed when it encountered the <code>devlink_file</code> , where <code>devlink</code> is the name of the symbolic link in the <code>/dev</code> directory.	Determine what caused this error by using the <code>readlink(2)</code> man page and the <code>errno_value</code> . 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 restore cwd <code>errno_value</code>	The DR daemon was unable to change back to the original directory after it changed into <code>/dev</code> or <code>/devices</code> directories. The DR daemon changes into the <code>/dev</code> and <code>/devices</code> 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 <code>errno_value</code> .
Unable to set cwd <code>errno_value</code>	The DR daemon could not change into the <code>/dev</code> and <code>/devices</code> directories. The daemon changes into these directories to produce the real path names that correspond to device drivers.	Determine what caused this error by using the <code>chdir(2)</code> man page and the <code>errno_value</code> , then correct the error.

TABLE 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
unknown node type	The device tree was built incorrectly. Several functions create the device tree for a system board by using the <code>libdevinfo</code> API, and search the <code>/dev</code> and <code>/devices</code> directories. After the tree is constructed, it is passed on to the <code>rpc_info()</code> 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 from the system logs as possible.
utssys failed (<i>errno_value</i>) for <i>mount_point</i>	The <code>utssys()</code> 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 <i>errno_value</i> , 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.
walk_dir: dirlist buffer overflow.	As it traversed the <code>/dev</code> and <code>/devices</code> 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 <code>/dev</code> and <code>/devices</code> directories for recursive symbolic links. Remove any recursive symbolic links that you find.

TABLE 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
<p>walk_dir: tpath buffer overflow. <i>target_path</i>, <i>device_name</i></p>	<p>The DR daemon cannot add another directory to the <i>target_path</i>. The daemon traverses the <code>/dev</code> and <code>/devices</code> 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 searching, 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.</p>	<p>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.</p>
<p>WARNING: cannot check for cvc/ssp interface.</p>	<p>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 <code>netcon(1M)</code> sessions are also disabled.</p>	<p>Switch the suspected interface to a redundant network connection on another board. You may have to reboot the domain to recover from this error.</p>
<p>WARNING: Cannot check for primary interface</p>	<p>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.</p>	<p>Determine which board hosts the primary network interface and re-attach the board to the domain. Alternatively, 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.</p>

TABLE 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
WARNING: Cannot determine if <i>interface_name_instance</i> is cvc/ssp interface. SIOCGIFNETMASK errno= <i>errno_value</i>	The DR daemon failed to obtain the necessary information to test an active network interface to determine if it is the SSP connection. 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 (<i>interface_name</i>) to another board. If you cannot correct this error, you may have to reboot the domain.
WARNING: cannot stat <i>device_name</i> errno= <i>errno_value</i>	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 <i>errno_value</i> to determine why the file <i>device_name</i> could not be accessed.
NGDR Error: Bad page size from sysconf . . . <i>errno_description</i>	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 <i>errno_value</i> to determine the cause of the error.
NGDR Error: device tree not built.	The libdevinfo API failed to build the device tree for the system board. Detailed information about this error accompanies the error message.	Make sure that the correct version of the libdevinfo API 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.
NGDR Error: dr_get_partn_cpus: failed to get cpu partition info . . . <i>errno_description</i>	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 <i>errno_description</i> to determine and correct the causes of this error.
NGDR Error: dr_page_to_kb: page size smaller than a KB	A math error occurred, or an incorrect memory value was used in a memory calculation.	Report this error to your Sun service representative.

TABLE 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
NGDR 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 restart the DR application, then retry the operation. If the error persists, use the <code>kill(1M)</code> command to stop the DR daemon, then start the DR daemon and retry the DR operation.
NGDR Error: get_board_config: invalid flag	The SSP passed an invalid or unsupported flag to the DR daemon when the daemon tried to determine 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 <code>ps(1)</code> 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.
NGDR Error: libdevinfo failed.	The initial routine used to open the <code>libdevinfo</code> API failed. Therefore, the DR daemon could not explore the device tree for that board. The <code>libdevinfo</code> 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 <code>libdevinfo</code> is included for 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 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
<p>NGDR Error: build_rpc_info: bad slot number</p>	<p>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, the message 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.</p>	<p>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 larger than 400Kbytes, then an internal error may have occurred within it. 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.</p>
<p>NGDR Error: build_rpc_info: device address format error</p>	<p>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, the message 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.</p>	<p>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 larger than 400Kbytes, then an internal error may have occurred within it. 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.</p>
<p>NGDR Error: build_rpc_info: I/O bus node address format error</p>	<p>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, the message 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.</p>	<p>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 larger than 400Kbytes, then an internal error may have occurred within it. 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.</p>

TABLE 4-11 DR Domain Exploration Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
NGDR 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, the message 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 larger than 400Kbytes, then an internal error may have occurred within it. 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.
NGDR 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, the message 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 larger than 400Kbytes, then an internal error may have occurred within it. 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.

Unsafe-Device Query Error Messages

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 4-12 Unsafe-Device Query Error Messages

Error Message	Probable Cause	Suggested Action
unsafe_devices: couldn't determine name of unsafe device <i>major_number</i>	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, <i>major_number</i> ".	This message notifies the user that the DR daemon was unable to find the name of one of the devices. This is not a correctable error. The daemon can use the major number to identify the drive.

TABLE 4-12 Unsafe-Device Query Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
<p>WARNING: board <i>board_number</i> not checked for unsafe devices.</p>	<p>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 (<i>board_number</i>). This error message may be indicative of a more serious problem.</p>	<p>Check the size of the DR daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon has grown larger than 400Kbytes, then an internal error may have occurred within it. 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.</p>
<p>NGDR Error: unsafe_devices: libdevinfo failed.</p>	<p>The DR daemon cannot determine the names of unsafe devices because it cannot use the <code>libdevinfo</code> API. This API must be used to search the device tree for the names of all unsafe devices.</p>	<p>Check the size of the DR daemon by using the <code>ps(1)</code> command. Normally, the daemon uses about 300 to 400 Kbytes of memory. If the daemon has grown larger than 400Kbytes, then an internal error may have occurred within it. 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.</p>
<p>NGDR Error: create_ctlr_array: count mismatch [internal error]</p>	<p>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.</p>	<p>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.</p>

Alternate Pathing (AP) 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 4-13 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: <i>error_message</i> *OR* error = <i>error_number</i>	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 <code>ap_daemon(1M)</code> man page for more details about this error.
AP daemon comm init failed: <i>error_message</i> *OR* error = <i>error_number</i>	The DR daemon encountered a failure when it tried to establish a channel of communication with the AP librarian.	A descriptive error message may be available to provide specific details about this failure, or an error number may be available. Also, check the <code>ap_daemon(1M)</code> man page for more details about this error.
AP daemon query failed: <i>error_message</i> *OR* error = <i>error_number</i>	The DR daemon could not successfully query the AP librarian on the usage of a specific I/O controller.	A descriptive error message may be available to provide specific details about this failure, or an error number may be available. Also, check the <code>ap_daemon(1M)</code> 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 <code>ap_daemon(1M)</code> man page for more details about this error.

TABLE 4-13 AP-related Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
Cannot find physical device for <i>AP_alias</i> . 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 to the appropriate directories.
create_ap_net_leaf: interface instance not found	The DR daemon tries to match the AP meta-network interfaces with the physical devices 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, or 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 <i>state_number</i>	The DR daemon called one of its internal functions with a bad value. However, this error indicates 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 has not been 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

DCS Error Messages

The following table contains DCS error messages that are sent to the console window, to the `/var/adm/messages` directory, and to the `$$SSPLOGGER/domain_name/messages` directory.

TABLE 4-14 DCS Error Messages

Error Message	Probable Cause	Suggested Action
DCS ERROR: permission denied	Only the superuser of the domain can run the DCS.	Check the <code>inetd.conf</code> file on the domain to ensure that the DCS is started with superuser ID.
DCS ERROR: internal error: operation: <i>error_description</i>	An internal error occurred within the DCS.	Use the <i>error_description</i> , which corresponds with the <i>errno_value</i> , to diagnose the error. The operation field refers to the function call that caused the error.
DCS NOTICE: unrecognized error reported	The DCS reported an unknown error condition.	Use the log file on the domain to help determine what caused the error.
DCS ERROR: network initialization failed	The DCS failed to initialize the network connection used to accept DR requests from the DCA.	Retry the DR operation.
DCS ERROR: failed to acquire reserved port	The DCS uses port 665, which is reserved through <code>sun-dr</code> . The error occurred because another process is using the port.	Determine if another process is still using the port. If so, kill the process, if possible, then retry the DR operation.
DCS ERROR: connection attempt failed	The DCS failed to establish a connection with the DCA.	Retry the DR operation.

TABLE 4-14 DCS Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
DCS ERROR: unable to receive message	The DCS failed to receive a message from the DCA.	Retry the DR operation.
DCS ERROR: unable to send message for <i>operation_name</i> operation	The DCS failed to send a message to the DCA.	Retry the DR operation.
DCS NOTICE: sun-dr service not found, using reserved port 665	The DCS failed to find the sun-dr service in /etc/services.	None
DCS NOTICE: client disconnected	The client unexpectedly disconnected.	None
DCS ERROR: unknown operation requested	The DCA requested an operation that is not recognized by the DCS.	Retry the DR operation.
DCS ERROR: operation failed	The current DCS operation failed to complete. The DR operation can still succeed, if the DCS failed only to send the results to the DCA.	Check the status of the operation manually. If the DR operation did not succeed, retry the operation.
DCS ERROR: invalid session establishment sequence	The session establishment sequence and (the initialization handshake) between the DCA and the DCS failed.	Retry the DR operation.
DCS ERROR: <i>operation_name</i> operation issued before session established	A DR operation was requested before the session was established.	Retry the DR operation.
DCS ERROR: received an invalid message	The DCS received unexpected information in the message.	Retry the DR operation.
DCS NOTICE: confirm callback failed, aborting operation	The DCS was unable to display the confirmation prompt to the user.	None
DCS NOTICE: message callback failed, continuing	The DCS was unable to display a message to the user.	None

TABLE 4-14 DCS Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
DCS NOTICE: retry value invalid (<i>retry_value</i>)	The value given for the <i>retry_value</i> was invalid, so the operation proceeded with the retry value set to zero.	None
DCS NOTICE: timeout value invalid (<i>timeout_value</i>)	The value given for the <i>timeout_value</i> was invalid, so the operation proceeded with the retry value set to zero.	None
DCS INFO: retrying operation, attempt <i>attempt_number</i>	The DCS is retrying the operation. The <i>attempt_number</i> field represents the current attempt.	None
DCS ERROR: failed to start a new session handler	The DCS failed to start a concurrent session handler to process the incoming DR request.	Retry the DR operation.
DCS ERROR: abort attempt of session, <i>session_id</i> , unsuccessful	The DCS failed to abort session, <i>session_id</i> .	Retry the abort request.
DCS ERROR: unsupported message protocol version: <i>version_number</i>	The DCS does not support the reported protocol version, <i>version_number</i> .	Check the DR software on the domain and the SSP. Reinstall the proper version of the software on the domain if they are not compatible.
DCS INFO: session aborted	The current DR operation was aborted by the user.	None
DCS ERROR: illegal option <i>option</i> , exiting	The DCS was passed an illegal option name.	Check the <code>inetd.conf</code> file on the domain and remove the illegal option from the entries for the DCS.

TABLE 4-14 DCS Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
DCS NOTICE: illegal argument to <i>option</i> flag (<i>argument</i>), <i>action</i>	The option <i>option</i> was given the illegal argument <i>argument</i> . The DCS will perform the action specified by <i>action</i> .	Check the <code>inetd.conf</code> file on the domain and fix the entries for the DCS.
DCS ERROR: resource info init error (<i>error_code</i>)	The DCS failed to initialize the module responsible for providing resource usage information.	Retry the operation.

DR Driver Error Messages

The following table contains DR driver error messages that are sent to the console window, to the `/var/adm/messages` directory, and to the `$$SPLOGGER/domain_name/messages` directory.

TABLE 4-15 DR Driver Error Messages

Error Message	Probable Cause	Suggested Action
dr: Internal error: dr.c <i>line_number</i>	An internal error has occurred in the DR driver.	Retry the operation that failed. If the error persists, exit and restart various DR software components, then retry the operation. If the problem still persists, reboot the domain. Check the console or the system logs for additional information.
dr: Insufficient memory: <i>resource</i>	The DR framework was unable to configure or unconfigure resources because a <code>KPHYSM_ERESOURCE</code> error or <code>cpu_configure()/cpu_unconfigure()</code> error with the <code>ENOMEM</code> <i>errno</i> occurred.	This condition might be transient. Retry the DR operation. If the error persists and if the operation that is failing is the unconfigure operation, then try configuring more memory into the domain from a different domain. If the error still persists, reboot the domain.

TABLE 4-15 DR Driver Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
dr: Device busy: <i>resource</i>	Translation of possible EBUSY errno messages from <code>cpu_configure()</code> or <code>cpu_unconfigure()</code> ; or an I/O device cannot be detached because it is busy. This error message is also returned if a CPU to be detached is online when <code>dr_pre_detach_cpu</code> is called. A CPU cannot be detached while a memory drain is in progress.	Use <code>showdevices(1M)</code> on the system controller to find out why the resource is busy. Or, on the domain, use <code>fuser(1M)</code> , <code>psrinfo(1M)</code> , <code>prtdiag(1M)</code> , or similar tools to find out why the device is busy. Also check whether another memory drain is already in progress. Either reconfigure or shut down whatever is consuming the resource, or wait for the previous memory drain to complete, depending on the cause of the error. Then retry the DR operation.
dr: Operation already in progress: <i>resource</i>	Translation of possible EALREADY errno from <code>cpu_configure()</code> or <code>cpu_unconfigure()</code> .	Use <code>showdevices(1M)</code> on the system controller to examine the configuration of the specified resource. Or, on the domain, use <code>cfgadm(1M)</code> , <code>pbind(1M)</code> , <code>psrinfo(1M)</code> , and similar commands to examine the configuration of the resource. Determine what operations are already in progress on this resource, and either wait for them to complete or cancel them. Then, retry the DR operation. The operation already in progress may already have terminated, so retrying to the operation might succeed, or may produce another error.
dr: I/O error: <i>resource</i>	An unexpected error code resulted from a call to <code>kphysm_del_start</code> . A more verbose <code>cmn_err</code> message is also printed.	Check the verbose error message from <code>cmn_err</code> in the system logs, and/or on the console for a more specific condition and suggested action.
dr: Bad address: <i>resource</i>	<code>kphysm_add_memory_dynamic</code> returned <code>KPHYSM_EFAULT</code> .	Retry the DR operation. If this error persists, contact your Sun Service representative.
dr: No device(s) on board: <i>board_path</i>	The board is connected or disconnected with no devices (I/O, memory, or CPU).	If devices were expected to be on the board, then disconnect the board and remove it from the system. The board's components should be reseated by a qualified technician.
dr: Invalid argument: <i>attachment_point</i>	DR was passed an invalid argument.	Retry the DR operation. If this error persists, contact your Sun service representative.

TABLE 4-15 DR Driver Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
dr: Invalid state transition: <i>attachment_point</i>	A DR operation was sequenced out of order. This could be operator error if the <code>cfgadm(1M)</code> commands were issued out of order. Or, the DR driver could be confused due to some internal error conditions.	Retry the DR operation. If this error persists, stop and restart (or unload and load) DR software components to recover from this error condition. If the error persists, reboot the domain.
dr: Device in fatal state	The device could not be suspended, or it refused to be suspended.	Retry the DR operation. If this error persists, the device could be in suspend-unsafe mode. Check the list of suspend-unsafe devices. If the device is unsafe, use <code>showdevices(1M)</code> or <code>fuser(1M)</code> to show whether the device is in use, and manually reconfigure the resource. Then, manually unload the driver, or if needed, disconnect the cables attached to the device. The device should now be safe to retry the operation. Do not reconnect the cables to the device, reload its driver, or reconfigure its resources before the DR operation has completed successfully.
dr: Device failed to resume: <i>path</i>	A previously suspended device could not be resumed.	
dr: Cannot stop user thread	DR could not stop a user thread(s) in preparing a device to be suspended.	Retry the DR operation. If this error persists, examine the user threads that failed to suspend, and determine why they could not be suspended. You might have to kill the threads to enable the DR operation to proceed.
dr: Cannot quiesce realtime <i>thread</i>	A realtime thread was encountered in an attempt to suspend the operating system. Suspending, or quiescence, of realtime threads is not allowed. All realtime threads must be stopped or changed to non-realtime before a suspend can succeed.	Kill realtime thread(s), or adjust their priority by using the <code>pricntl(1M)</code> command. (You must obtain the PID to adjust the priority of realtime threads.)
dr: Cannot stop kernel thread: <i>name</i>	DR could not stop a kernel thread.	Retry the DR operation. If this error persists, examine the user threads that failed to suspend, and determine why they could not be suspended. Kill the kernel threads, if possible, to enable the DR operation to proceed.

TABLE 4-15 DR Driver Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
dr: Failed to off-line: <i>cpu</i>	A CPU could not be brought off-line, which prevents it from being unconfigured. The CPU might have a thread(s) bound to it. An additional <code>cmn_err</code> message is logged if there are threads bound to the CPU. DR must be able to off-line CPUs and/or to power off CPUs before the board can be disconnected.	Check the console and system log messages to determine if threads are bound to the CPU. If they are, they can be manually unbound or rebound to CPUs on other boards in the domain. If threads are not bound to the CPU, use <code>psrset(1M)</code> , <code>pbind(1M)</code> , and <code>psrinfo(1M)</code> to determine what changes are required to enable DR to off-line the CPU. For example, you might have to add more CPUs to the domain from different boards. Or, you may have to online other CPUs. Finally, you might have to add more CPU boards to take over the CPU workload.
dr: Failed to on-line: <i>cpu</i>	DR could not online a CPU on a newly-connected or previously unconfigured board.	
dr: Failed to start CPU: <i>cpu</i>	DR could not start a CPU on a newly-connected or previously unconfigured board.	
dr: Failed to stop CPU: <i>cpu</i>	DR could not power off a CPU on a board to be unconfigured. All of the CPUs on a board to be unconfigured must be taken offline and powered off before the operation can succeed.	
dr: Kernel cage is disabled: <i>resource</i>	When the kernel cage is disabled, boards hosting permanent memory cannot be detached.	Enable the kernel cage in <code>/etc/system</code> and reboot the domain.
dr: No available memory target: <i>resource</i>	DR could not detach the board because it hosts permanent memory and there is no available target for the memory. Permanent memory must be moved to another memory component within the domain before the DR operation can succeed.	Configure an additional memory component that contains an adequate amount of memory to act as a target for this board. Then, retry the DR operation.
dr: VM viability test failed: <i>resource</i>	Translation of error code returned by <code>kphysm_del_start</code> .	Configure additional memory components into the domain to relieve memory resource pressure. Then, retry the DR operation.

TABLE 4-15 DR Driver Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
dr: kphysm_pre_del failed: <i>resource</i>	Translation of error code returned by kphysm_del_start.	Configure additional memory components into the domain to relieve memory resource pressure. Then, retry the DR operation.
dr: Non-relocatable pages in span: <i>resource</i>		
dr: kphysm_del_cancel: <i>resource</i>		
dr: Memory operation failed: <i>resource</i>	DR failed to attach the memory on a newly attached board.	
dr: Can't unconfig cpu if mem online	DR cannot unconfigure a CPU if the memory on the board is online.	You must bring memory offline before you can unconfigure the board.
ngdrmach: Cannot read property value: Device Node <i>node_address</i> : property <i>property_name</i>	DR could not get the specified property of a particular device node.	
ngdrmach: Cannot determine property length: <i>board::slot:property</i>	DR could not get the length of the specified property for a particular device node.	
ngdrmach: No CPU specified for connect: <i>slot</i>		
ngdrmach: Cannot move SIGB assignment		
ngdrmach: Cannot disconnect CPU; SIGB is currently assigned: <i>slot::board</i>		
ngdrmach: Device driver failure: <i>path</i>		
ngdr: Must specify a CPU on the given board: <i>cpu_id</i>		
ngdrmach: No such device: <i>board::slot</i>		
ngdrmach: Memory configured with inter-board interleaving: <i>board::slot</i>		

TABLE 4-15 DR Driver Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
ngdrmach: Invalid board number: <i>board_number</i>	An invalid board number was specified for the assign board operation.	Use a different board number, or fix the available components list on the system controller for the domain to include the board for which the assign function is failing.
ngdrmach:: Cannot proceed; Board is configured or busy: <i>component_name</i>	DR cannot power off or unassign a board that is configured or busy.	Unconfigure the board, or wait for any previous DR operations on the board to complete. Then, retry the DR operation
ngdrmach: Firmware probe failed: <i>attachment_point</i>	OBP failed to probe the board.	
ngdrmach: Firmware deprobe failed: <i>attachment_point</i>	OBP failed to deprobe the board.	
ngdrmach: Operation not supported	The operation you attempted is not supported.	None
ngdrmach: Unrecognized platform command: <i>command/options</i>	An unrecognized command was passed to DR.	Refer to the <code>cfgadm_sbd(1M)</code> man page to ensure that you use a valid argument. If you used a valid argument and this error persists, contact your Sun service representative.
ngdrmach: drmach parameter is not a valid ID	An invalid <code>drmachid_t</code> value was encountered.	
ngdrmach: drmach parameter is inappropriate for operation	The wrong type of <code>drmachid_t</code> was passed to a function.	
ngdrmach: Unexpected internal condition: <i>drmach.c line_number</i>	An internal <code>drmach</code> error occurred.	Use <code>modunload(1M)</code> and <code>modload(1M)</code> to unload then to load the <code>drmach</code> driver. Then, retry the DR operation. If this error persists, then you must reboot the domain.
ngdrmach: No CPU specified for connect.		
ngdrmach: Firmware move_cpu0 failed: CPU <i>cpu_id</i>		
ngdrmach: Cannot move SIGB assignment		

TABLE 4-15 DR Driver Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
ngdrmach: Cannot disconnect CPU; SIGB is currently assigned		

Plugin Error Messages

The following error messages are generated by the `libcfgadm` system board plugin. The messages are sent to the `netcon(1M)` console window, to the `/var/adm/messages` directory, and to the `$SSPLOGGER/domain_name/messages` directory.

TABLE 4-16 Plugin Error Messages

Error Message	Probable Cause	Suggested Action
Configuration operation cancelled: command <i>ap_id</i>	You did not confirm a configuration operation that requires confirmation.	See the <code>cfgadm(1M)</code> and/or the <code>cfgadm_sbd(1M)</code> man page for more information about which configuration operations require confirmation.
Hardware specific failure: <i>command ap_id</i> : <i>error: resource</i>	A system error occurred during the execution of the command. The error message, <i>error</i> , can be a standard error, or it can be a more specific error message that is returned by the DR driver. (See the DR Driver error messages for more information.) The name of the resource, <i>resource</i> , that is causing the error (for example, a busy device) can also be returned by the DR driver.	For busy devices, identify and stop usage of the device. For other errors, refer to the driver's documentation for recovery options.
Library Error: command invalid: <i>command</i>	The specified command is invalid for system boards.	Refer to the <code>cfgadm_sbd(1M)</code> man page for a list of valid commands.
Library Error: command not supported: command <i>ap_id</i>	The command that was executed is not supported for the attachment point specified by <i>ap_id</i> . For example, the <code>connect</code> operation is not allowed for CPUs.	Refer to the <code>cfgadm_sbd(1M)</code> man page for a list of supported commands.
Library Error: command aborted: <i>command</i>	You aborted the command.	N/A

TABLE 4-16 Plugin Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
Library Error: option invalid: <i>option</i>	The specified option, <i>option</i> , is invalid.	Refer to the <code>cfgadm_sbd(1M)</code> man page for a list of the valid options.
Library Error: option requires value: <i>option</i>	The specified option, <i>option</i> , requires a value.	Refer to the <code>cfgadm_sbd(1M)</code> man page for a list of the option values.
Library Error: option requires no value: <i>option</i>	The specified option, <i>option</i> , does not require a value.	Refer to the <code>cfgadm_sbd(1M)</code> man page for a list of options that do not require values.
Library Error: option value invalid: <i>option value</i>	The specified value, <i>value</i> , for the option, <i>option</i> , is invalid.	Refer to the <code>cfgadm_sbd(1M)</code> man page for a list of valid option values.
Library Error: attachment point invalid: <i>ap_id</i>	The specified attachment point, <i>ap_id</i> , could not be parsed correctly. This error is rare and could indicate an internal error.	Refer to the <code>cfgadm_sbd(1M)</code> man page for a list of valid attachment points. If this error persists, contact your Sun service representative.
Library Error: component invalid: <i>ap_id</i>	The specified component, <i>ap_id</i> , is invalid.	Refer to the <code>cfgadm_sbd(1M)</code> man page for a list of valid dynamic attachment points.
Library Error: sequence invalid: <i>command</i> (<i>rstate</i> <i>ostate</i>) <i>ap_id</i>	The specified command, <i>command</i> , is invalid for the receptacle and/or occupant state of the specified attachment point. For example, trying to connect an empty slot results in an invalid sequence error.	Refer to the <code>cfgadm_sbd(1M)</code> man page for a list of valid operations.
Library Error: offline <i>ap_id</i> (<i>path</i>): <i>error</i>	The Reconfiguration Coordination Manager (RCM) failed to take the resource, <i>ap_id</i> , offline. The error message, <i>error</i> , returned by the RCM will indicate the reason for the failure. Usually, the reason is a busy device.	For busy devices, identify and remove the usage of the device.
Library Error: suspend <i>ap_id</i> (<i>path</i>): <i>error</i>	The Reconfiguration Coordination Manager (RCM) failed to suspend the resource, <i>ap_id</i> . The error message, <i>error</i> , returned by the RCM will indicate the reason for the failure. Usually, the reason is a busy device.	For busy devices, identify and remove the usage of the device.
Library Error: not enough memory	The plugin operation failed due to a lack of memory.	Check the memory usage.
Library Error: change signal disposition failed	The plugin operation failed to set up signals before it started the DR operation.	None

TABLE 4-16 Plugin Error Messages (Continued)

Error Message	Probable Cause	Suggested Action
Library Error: cannot get RCM handle	The Reconfiguration Coordination Manager (RCM) failed to initialize.	None
Library Error: cannot open library: <i>error</i>	The Reconfiguration Coordination Manager (RCM) library, <i>library</i> , was found, but an error occurred when it was opened. The error message, <i>error</i> , is returned by <code>dlopen(3DL)</code> .	Check for proper installation of the RCM.
Library Error: cannot find symbol <i>symbol</i> in <i>library</i>	A required symbol, <i>symbol</i> , was not found in the Reconfiguration Coordination Manager (RCM) library, <i>library</i> .	Check for proper installation of the RCM.
Library Error: cannot stat <i>library</i> : <i>error</i>	The Reconfiguration Coordination Manager (RCM) library, <i>library</i> , exists, but the <code>stat(2)</code> function failed to get the file status. The error message, <i>error</i> , will be returned by the Solaris operating environment.	None

Index

A

- abort button, 16, 26
- active DR operations, only one, 10
- addboard(1M) procedure, 45
- adding boards, 45
- amount of memory attachable, 35
- attach, 9
- attach buttons, 15
- attach, complete, 18
- attachable memory, 35
- attaching with dr(1M), 19
- Automated DR (ADR), 5
 - in DR model 2.0, 9
 - in DR model 3.0, 41
 - on I/O boards, 42
- automatic DR, 41

B

- board attach, 9
- board detach, 9, 23
- board, attach, 17
- buttons
 - abort, 16, 26
 - complete, 16, 18, 25, 28
 - CPU, 33
 - device, 36
 - dismiss, 16, 26
 - drain, 25
 - force, 25

- help, 16, 26
- init attach, 16, 18
- memory, 34
- reconfig, 16, 25
- select, 17

C

- CEs (correctable memory errors) and detach, 11
- communication timeouts affecting Hostview and dr(1M), 13
- complete attach, 15, 18
- complete attach vi dr(1M), 22
- complete button, 16, 18, 25, 28
- complete detach, 24, 28
- complete detach via dr(1M), 31
- connection, loss of, 13
- copying nonpageable memory before detach operation, 10
- correctable memory errors (CEs) and detach, 11
- CPU button, 33
- CPU configuration window, 33
- CPUs
 - detachability, 2
 - suspending, 2

D

- DDI_DETACH, 2
- deleteboard(1M) procedure, 45

- deleting boards, 45
- detach, 9, 23
 - configuring memory for detach, 10
 - correctable memory errors (CEs) and detach, 11
 - interleaved memory and detach, 10
- detach buttons, 25
- detachability, 2
- detaching with dr(1M), 29
- detaching with Hostview, 26
- device button, 36
- dismiss button, 16, 26
- domain_status(1M)
 - output, 43
- domains, description of, 1
- DR
 - concepts, 2
 - DR attach, 9
 - DR detach, 9, 23
 - DR overview, 5
 - dr(1M), attaching via, 19
 - dr.conf file, 3
 - drain, 23
 - drain button, 25
 - drain via dr(1M), 30
 - drain, percent complete, 36
 - drivers
 - unsafe, 3
 - dr-max-mem environment variable, 35
 - drshow, dr(1M) command, 22
 - DR-unsafe device, 3
 - dynamic system domains, 1

E

- environment variable, dr-max-mem, 35
- Error messages, 49
 - domain, 64
 - Alternate Pathing, 96
 - DCS, 98
 - DR Daemon start-up, 64
 - DR domain exploration, 82
 - DR driver, 101
 - DR driver failures, 65
 - DR general domain, 77

- memory allocation, 68
- OpenBoot PROM, 79
- plugin, 107
- PSM, 75
- unsafe-device query, 94
- SSP, 50
 - attach-related, 56
 - auto-configuration, 62
 - detach-related, 57
 - protocol and communication, 51

F

- files
 - .postrc, and memory interleaving, 10
- files, error message log, 49
- files,SSP messages, 10
- force button, 25

H

- hardware partitions, 42
- help button, 16, 26
- Hostview, detaching via, 26
- hot swap, 42

I

- I/O devices
 - detachability, 2
 - reconfiguration of, 42
 - suspending, 2
 - suspend-safe, 3
- init attach, 14, 18
- init attach button, 16, 18
- init attach, with dr(1M), 21
- interleaved memory and detach, 10
- interleaved memory, determining if enabled, 10

L

- loss of connection, 13

M

- memory attach capacity, 35
- memory button, 34
- memory configuration window, 34
- Memory Configuration window (Hostview), and nonpageable memory, 11
- memory draining, detach, 23
- memory interleaving, determining if enabled, 10
- memory reduction, detach operation, 36
- memory remaining in system, 36
- memory, configuring for detach, 10
- memory, pageable and nonpageable, 10
- memory, total size (all boards), 35
- moveboard(1M) procedure, 46
- moving boards, 46
- multipathing, 5, 6

N

- nonpageable and pageable memory, 10
- nonpageable memory and Memory Configuration window (Hostview), 11
- nonpageable memory, copying before detach operation, 10
- nonpageable memory, target board for copying, 11

O

- overview of DR, 5

P

- pageable and nonpageable memory, 10
- percent complete, drain, 36

Q

- quiesce OS during detach, and nonpageable memory, 10
- Quiescence, 2

R

- RCM (reconfiguration coordination manager), 41
- reconfig button, 16, 25
- reconfiguration coordination manager (RCM), 41
- record-stop dumps and detach, 11
- reduction of memory
 - detach operation, 36
- remaining memory, detach operation, 36
- replacing system boards, 47
- RPC timeout, 13

S

- select button, 17
- showdevices(1M)
 - output, 44
 - procedure, 44
- SSP messages file, 10
- suspending OS during detach, and nonpageable memory, 10
- suspend-safe devices, 3
- system board, target for copying nonpageable memory, 11
- system boards, replacing, 47
- system information, viewing, 31

T

- target board for nonpageable memory copy, 11
- target domain, attach, 17
- timeout, RPC, 13
- timeouts affecting Hostview and dr(1M), 13

U

- unsafe devices, 3

V

- viewing system information, 31

W

windows

- CPU configuration, 33

- memory configuration, 34