



Sun Fire™ 15K Dynamic Reconfiguration (DR) User Guide

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

Part No. 806-6808-11
October 2001, Revision A

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

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 Fire, OpenBoot, Sun Management Center, Sun RSM Array, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

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 Fire, OpenBoot, Sun Management Center, Sun RSM Array, 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.



Contents

Preface vii

Before You Read This Book vii

How This Book Is Organized vii

Using UNIX Commands viii

Typographic Conventions viii

Shell Prompts ix

Related Documentation ix

Ordering Sun Documentation ix

Accessing Sun Documentation Online x

Sun Welcomes Your Comments x

1. Restrictions on Using Sun Fire 15K Dynamic Reconfiguration (DR) 1

DR Limitations 1

2. Introduction to DR on the Sun Fire 15K Server 3

What Is DR? 3

Command Line Interface (CLI) 3

Graphical User Interface (GUI) 4

Automatic DR 4

Enhanced System Availability 4

DR Concepts	5
Detachability	5
Quiescence	5
Suspend-Safe and Suspend-Unsafe Devices	6
Attachment Points	6
Conditions and States	7
DR Operations	7
Hot-Plug Hardware	8
Sun Fire 15K Domains	8
Component Types	9
DR on I/O Boards	10
Problems With I/O Devices	10
Nonpermanent and Permanent Memory	11
Target Memory Constraints	11
Correctable Memory Errors	11
An Illustration of DR Concepts	12
3. DR State and Condition Models	15
Board States and Conditions	15
Board Receptacle States	15
Board Occupant States	16
Board Conditions	17
Component States and Conditions	17
Component Receptacle States	17
Component Occupant States	17
Component Conditions	18
4. DR Operations and Software Components on the Domain	19
DR Operations	19

Connect Operation	20
Configure Operation	21
Disconnect Operation	22
Unconfigure Operation	22
Nonpermanent Memory	23
Permanent Memory	23
Software Components	25
Domain Configuration Server	25
DR Driver	25
Reconfiguration Coordination Manager	25
System Events Framework	26
5. DR User Interfaces on the Domain	27
DR Commands and Options on the Domain	27
State Change Functions	28
Availability Change Functions	28
Condition Change Functions	28
Options and Operands	29
6. DR Domain Procedures	31
Attachment Points	31
Displaying Board Status	32
Basic Status Display	32
Detailed Status Display	32
Removing a Board	33
▼ To Remove a Board	33
▼ To Remove an I/O Board	33
Adding a Board	34
▼ To Install a Board	35

Preface

This book describes the dynamic reconfiguration (DR) feature on the Sun Fire™ 15K system. DR enables you to attach system boards to, and detach them from, Sun Fire 15K domains while the operating environment continues to run.

Before You Read This Book

This book is intended for the Sun Fire 15K system administrator who has a working knowledge of UNIX® systems, particularly those based on the Solaris™ operating environment. If you do not have such knowledge, first read the Solaris user and system administrator books in AnswerBook2™ format provided with this system and consider UNIX system administration training.

How This Book Is Organized

This book contains the following chapters:

Chapter 1 “Restrictions on Using Sun Fire 15K Dynamic Reconfiguration (DR)”

Chapter 2 “Introduction to DR on the Sun Fire 15K Server”

Chapter 3 “DR State and Condition Models”

Chapter 4 “DR Operations and Software Components on the Domain”

Chapter 5 “DR User Interfaces on the Domain”

Chapter 6 “DR Domain Procedures”

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:

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

Typographic Conventions

Typeface or Symbol	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <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>System Management Services (SMS) 1.0 Dynamic Reconfiguration User Guide</i>	806-6809
Reference	<i>System Management Services (SMS) 1.0 Reference Manual</i>	806-6141

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 (806-6808-11) of your document in the subject line of your email.

Restrictions on Using Sun Fire 15K Dynamic Reconfiguration (DR)

This pre-release version of the Dynamic Reconfiguration (DR) software on Sun Fire 15K systems has the following functional limitations:

DR Limitations

- When you want to use DR to reconfigure domains on a Sun Fire 15K system, make sure there are no more than three active domains on the system. To check the status of domains, enter the following command: `showboards | grep Active`. Active boards are indicated by “Active.” All three domains can be reconfigured using DR.
- In order to use DR on a Sun Fire 15K domain, the domain can contain up to five CPU/memory (SBx) boards and five I/O (IOx) boards. You can use either the `cfgadm -l` command or the `showboards(1M)` command to check the current domain configuration.
- DR is not supported on I/O boards. However, you can hot-plug hPCI cards on I/O boards to reconfigure I/O capacity dynamically. Do not use the `psradm(1m)` command concurrently with a hot-swap operation on the same domain.
- Do not perform multiple DR operations simultaneously on the same Sun Fire 15K system. This includes DR operations on CPU/memory boards and hot-plug operations on hPCI cards.
- Do not attempt to perform DR operations in a domain that includes a Max CPU system board.
- Do not attempt to perform DR operations on a CPU/memory board in slot 0 of any expander when slot 1 of the same expander is occupied by a Max CPU board. This restriction applies whether both boards reside in the same domain.

Introduction to DR on the Sun Fire 15K Server

This chapter contains descriptions about general concepts that pertain to the dynamic reconfiguration (DR) feature on the Sun Fire™ 15K server.

What Is DR?

The dynamic reconfiguration (DR) feature on the Sun Fire 15K server enables you to perform hardware configuration changes to a live domain that is running the Solaris Operating Environment, without causing machine downtime. You can also use DR in conjunction with hot-swap to remove or to add boards physically to the server.

You can execute DR operations from the Sun Fire 15K system controller (SC) by using the system management services (SMS) commands—`addboard(1M)`, `moveboard(1M)`, `deleteboard(1M)`, and `rcfgadm(1M)`, or from the domain by using the `cfgadm(1M)` command.

Command Line Interface (CLI)

The DR software has a command line interface using the `cfgadm` command, which is the configuration administration program. The DR agent also provides a remote interface to the Sun Management Center 3.0 software.

Graphical User Interface (GUI)

The optional Sun Management Center 3.0 Update 1 software, which is designed for these systems, provides features such as domain management, as well as a graphical user interface (GUI) to the `cfgadm` DR command line interface (CLI). If you prefer to use a graphical user interface instead of a command line interface, you can use the Sun Management Center 3.0 software instead of the command line interfaces of the system controller software and the DR software.

To use the Sun Management Center 3.0 software, you must attach the System Controller board to a network. With a network connection, you can view both the command line interface and the graphical user interface. For instructions on how to use the Sun Management Center 3.0 software, refer to the *Sun Management Center 3.0 User's Guide*, shipped with the Sun Management Center 3.0 software. For instructions on how to connect the system controller to a network connection on the System Controller board, see your systems installation documentation.

Automatic DR

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

Enhanced System Availability

The DR feature enables you to hot-swap system boards without bringing the server down. It is used to unconfigure the resources on a faulty system board from a domain so that the system board can be removed from the server. The repaired or replacement board can be inserted into the domain while the Solaris Operating Environment is running. The DR command then configures the resources on the board into the domain. If you use the DR feature to add or remove a system board or component, DR always leaves the board or component in a known configuration state (see Chapter 3 “DR State and Condition Models” for more information about configuration states for system board and components).

DR Concepts

This section contains descriptions of general DR concepts that pertain to Sun Fire 15K domains. For more information about DR concepts on the SC, refer to the *System Management Services (SMS) 1.0 Dynamic Reconfiguration User Guide*.

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 CPU board, that CPU board cannot be detached. An I/O board is not detachable if it controls the boot drive.

If there is no alternate pathway for an I/O board, you can:

- Put the disk chain on a separate I/O board.
The secondary I/O board can then be detached.
- Add a second path to the device through a second I/O board so that the I/O 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 a critical phase of the 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` file to prevent unsafe devices from accessing memory or interrupting the operating environment during a DR operation. The unsafe driver list is a property in the `dr.conf` 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.

Attachment Points

An attachment point is a collective term for a board and its slot. DR can display the status of the slot, the board, and the attachment point. The DR definition of a board also includes the devices connected to it, so the term “occupant” refers to the combination of board and attached devices.

- A slot (also called a *receptacle*) has the ability to electrically isolate the occupant from the host machine. That is, the software can put a single slot into low-power mode.
- Receptacles can be named according to slot numbers or can be anonymous (for example, a SCSI chain). To obtain a list of all available logical attachment points, use the `-l` option with the `cfgadm(1M)` command.
- An occupant I/O board includes any external storage devices connected by interface cables.

There are two types of names for attachment points:

- A physical attachment point describes the software driver and location of the slot. An example of a physical attachment point name is:

```
/devices/pseudo/dr@0:slotx.y
```

Where, `x` represents the expander number (0 to 17) for a particular board, and `y` represents the slot number (0 or 1).

- A logical attachment point is an abbreviated name created by the system to refer to the physical attachment point. Logical attachment points take one of the following two forms:

```
SBx (for CPUs and memory)
```

```
IOx (for I/O devices)
```

Conditions and States

A state is the operational status of either a receptacle (slot) or an occupant (board). A condition is the operational status of an attachment point. The `cfgadm(1M)` command can display nine types of states and conditions. See Chapter 3 “DR State and Condition Models” for descriptions of the conditions and states for system boards and components.

DR Operations

There are four main types of operations related to boards: connection, configuration, unconfiguration, and disconnection.

During the connect operation, the slot provides power to the board and begins monitoring the board temperature. For I/O boards, the connection operation is included in the configuration operation.

During the configure operation, the operating environment assigns functional roles to a board and loads device drivers for the board and for devices attached to the board.

During the unconfigure operation, the system detaches a board logically from the operating environment and takes the associated device drivers offline. Environmental monitoring continues, but devices on the board are not available for system use.

During the disconnect operation, the system stops monitoring the board and power to the slot is turned off.

If a system board is in use, stop its use and disconnect it from the domain before you power it off. After a new or upgraded system board is inserted and powered on, connect its attachment point and configure it for use by the operating environment.

The `cfgadm(1M)` command can connect and configure (or unconfigure and disconnect) in a single command, but if necessary, each operation (connection, configuration, unconfiguration, or disconnection) can be performed separately.

Hot-Plug Hardware

Hot-plug boards and modules have special connectors that supply electrical power to the board or module before the data pins make contact. Boards and devices that do not have hot-plug connectors cannot be inserted or removed while the system is running.

I/O boards and CPU/memory boards used in the Sun Fire 15K server are hot-plug devices. Some devices, such as the peripheral power supply, are not hot-plug modules and cannot be removed while the system is running.

Sun Fire 15K Domains

The Sun Fire 15K server can be divided into dynamic system domains, referred to as domains in this document. These domains are based on system board slots that are assigned to the domains. Each domain is electrically isolated into hardware partitions, which ensures that an arbitrary stop in one domain does not affect the other domains in the server.

Sun Fire 15K domain configuration is determined by the domain configuration table in the platform configuration database (PCD), which resides on the SC. The domain table controls how the system board slots are logically partitioned into domains. The domain configuration represents the intended domain configuration. Thus, the configuration can include empty slots and populated slots.

The number of slots available to a given domain is controlled by an available component list that is maintained on the system controller (refer to the *System Management Services (SMS) 1.0 Administrator Guide* for more information about the available component list. After a slot has been assigned to a domain, it becomes visible to that domain and unavailable and invisible to any other domain. Conversely, you must disconnect and unassign a slot from its domain before you can connect and assign it to another domain.

The logical domain is the set of slots that belong to the domain. The physical domain is the set of boards that are physically interconnected. A slot can be a member of a logical domain without having to be part of a physical domain. After the domain is booted, the system boards and the empty slot can be assigned to or unassigned from a logical domain; however, they are not allowed to become a part of the physical domain until the operating environment requests it. System boards or slots that are not assigned to a domain are available to all domains if the board is in the available component list for each domain. These boards can be assigned to a domain by the platform administrator; however, an available component list can be set up on the system controller to allow users with appropriate privileges to assign available boards to a domain.

Component Types

You can use DR to configure or to unconfigure several types of components. The following table contains the name and description of the component types.

Name	Description
cpu	An individual CPU
memory	All of the memory on the board
pci	Any I/O device, controller, or bus

DR on 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 of 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 is not necessary and need not be performed. In this case, device paths to the board devices will remain unchanged.

Problems With I/O Devices

All I/O devices must be closed before they are unconfigured. If you encounter a problem with an I/O device, the following list can help you to overcome the problem.

- Use the `fuser(1M)` command to see which processes have the device open.
- Run `showdevices(1M)` on the SC to determine the state and usage of the device.
- If disk mirroring is being used to access a device connected to the board, reconfigure the device so that it is accessible by way of controllers on other system boards.
- Unmount file systems.
- Remove multipathing databases from board-resident partitions. The location of multipathing databases is explicitly chosen by the user and can be changed.

Refer to the *Solaris 8 5/01 on Sun Hardware Release Notes Supplement* for special instructions for I/O devices.

- Remove any private regions used by volume managers. By default, volume managers use a private region on each device that they controls. Such devices must be removed from volume manager control before they can be detached.
- Take any RSM 2000 controllers offline by using the `rm6` or `rdacutil` commands.
- Remove disk partitions from the swap configuration.
- If a detach-unsafe device is present on the board, close all instances of the device and use `modunload(1M)` to unload the driver.



Caution – Unmounting file systems may affect NFS client systems.

- Either kill any process that directly opens a device or raw partition, or direct it to close the open device on the board.

Note – If you use the `ndd(1M)` command to set the configuration parameters for network drivers, the parameters may not persist after a DR operation. Use the `/etc/system` file or the `driver.conf` file for a specific driver to set the parameters permanently.

Nonpermanent and Permanent Memory

Before you can delete a board, the environment must vacate the memory on that board. Vacating a board means flushing its nonpermanent memory to swap space and copying its permanent (that is, kernel and OpenBoot™ PROM memory) to another memory board. To relocate permanent 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 permanent memory is the only time when the operating environment is suspended; therefore, you should know where permanent memory resides so that you can avoid significantly impacting the operation of the domain. You can display the permanent memory by using the `cfgadm(1M)` command with the `-v` option. When permanent memory is on the board, the operating environment must find other memory component of adequate size to receive the permanent memory.

Target Memory Constraints

When permanent memory is removed, 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 because the domain is not large enough to hold the permanent memory.

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 SC detects correctable memory errors, it initiates a record-stop dump to save the diagnostic data, which can interfere with a DR operation. Therefore, Sun Microsystems suggests that when a record-stop occurs from a correctable memory error, you allow the record-stop dump to complete its process before you initiate a DR operation.

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

An Illustration of DR Concepts

DR lets you disconnect and then re-connect system circuit boards without bringing the system down. You can use DR to add or remove system resources while the system continues to operate.

To illustrate reconfiguration of system resources consider the following Sun Fire system configuration, as depicted in the diagram that follows: Domain A contains system boards 0 and 2, and I/O board 2. Domain B contains system boards 1 and 3, and I/O boards 1, 3, and 4.

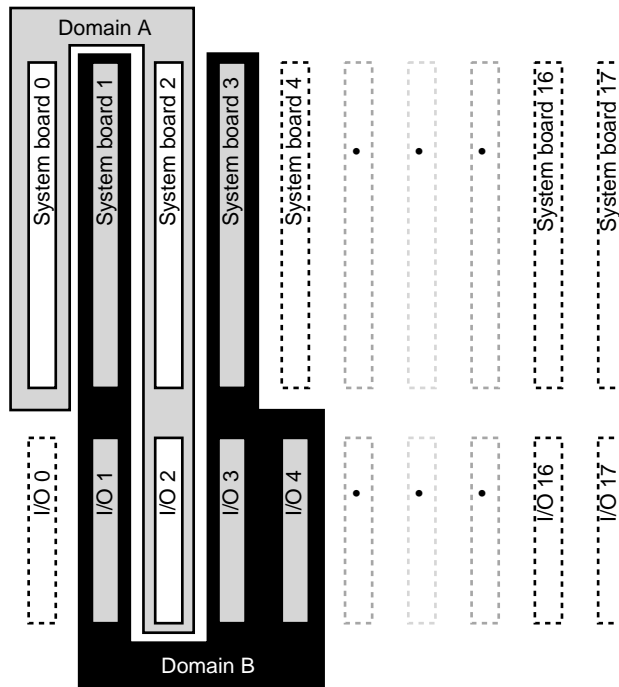


FIGURE 2-1 Domains A & B before Reconfiguration

To assign system board 4 and I/O board 0 to Domain A, and to move I/O board 4 from Domain B to Domain A, you can use the Sun Management Center software's GUI. Or you can perform the following steps manually on the CLI in each domain as follows:

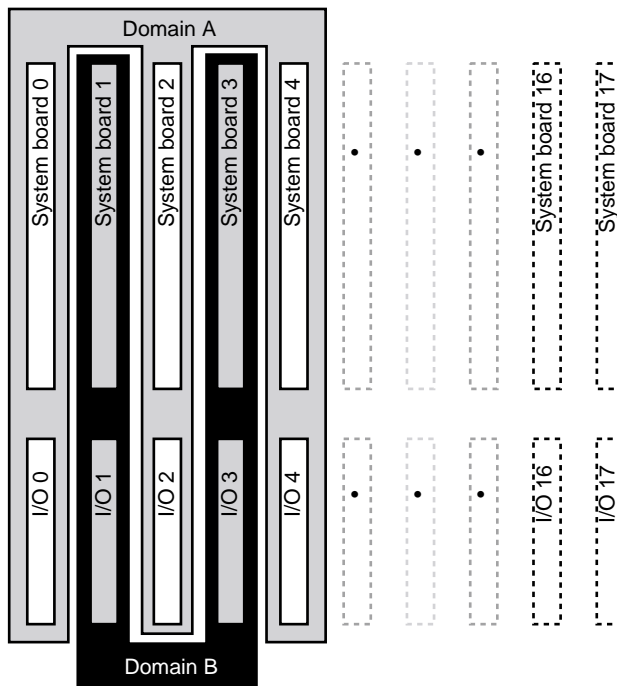
1. Enter the following configuration command on the command line in Domain B to disconnect I/O board 4 from Domain B:

```
% cfgadm -c disconnect -o nopoweroff,unassign IO4
```

2. Then, enter the following single command on the command line in Domain A, which assigns, connects, and configures system board 4 and I/O boards 0 and 4 into Domain A:

```
% cfgadm -configure SB4 IO0 IO4
```

The following system configuration is the result. Notice that only the way in which the boards are connected has changed, but not the physical layout of the boards within the cabinet.



DR State and Condition Models

This chapter contains descriptions of the state and condition models for boards and components. The state models are divided into two categories: receptacle and occupant.

Before you attempt to perform any DR operation on a board or component from the domain, you must determine state and condition. Use the `cfgadm(1M)` command with the `-la` options to display the type, state, and condition of each component and the state and condition of each board slot in the domain. See the Section “Component Types” on page 9 for a list of the component types.

Note – You can use the `prtdiag(1M)` command to display information about board slots and components. The `prtdiag(1M)` command displays board numbers in the format `SBxx` or `IOxx`, with two-digit board numbers that include leading zeroes, for readability.

Board States and Conditions

This section contains descriptions of the states and conditions of system boards (also known as system slots).

Board Receptacle States

A board can have one of three receptacle states: empty, disconnected, or connected. Whenever you insert a board, the receptacle state changes from empty to disconnected, or whenever you remove a board the receptacle state changes from disconnected to empty.



Caution – Physically removing a board that is in the connected state, or that is powered on and in the disconnected state, crashes the operating system and can result in permanent damage to that system board.

The following table contains the name and description of the receptacle states for boards.

Name	Description
empty	A board is not present.
disconnected	The board is disconnected from the system bus. A board can be in the disconnected state without being powered off. However, a board must be powered off and in the disconnected state before you remove it from the slot.
connected	The board is powered on and connected to the system bus. You can view the components on a board only after it is in the connected state.

Board Occupant States

A board can have one of two occupant states: configured or unconfigured. The occupant state of a disconnected board is always unconfigured. The following table contains the name and description of the occupant states for boards.

Name	Description
configured	At least one component on the board is configured.
unconfigured	All of the components on the board are unconfigured.

Board Conditions

A board can be in one of four conditions: unknown, ok, failed, or unusable. The following table contains the name and description of the conditions for boards.

Name	Description
unknown	The board has not been tested.
ok	The board is operational.
failed	The board failed testing.
unusable	The board slot is unusable.

Component States and Conditions

This section contains descriptions of the states and conditions for components.

Component Receptacle States

A component cannot be individually connected or disconnected. Thus, components can have only one state: connected.

Component Occupant States

A component can have one of two occupant states: configured or unconfigured. The following table contains the name and description of the occupant states for components.

Name	Description
configured	The component is available for use by the Solaris Operating Environment.
unconfigured	The component is not available for use by the Solaris Operating Environment.

Component Conditions

A component can have one of three conditions: unknown, ok, failed. The following table contains the name and description of the conditions for components.

Name	Description
unknown	The component has not been tested.
ok	The component is operational.
failed	The component failed testing.

DR Operations and Software Components on the Domain

This chapter contains descriptions of the four general DR operations: connect, configure, disconnect, and unconfigure. For more information on how to perform these operations, see Chapter 6 “DR Domain Procedures”.

This chapter also contains information about the various software components that work together to accomplish DR operations. The components that are used during a DR operation depend entirely on the point of initiation of the DR operation. For instance, if you initiate the DR operation from the Sun Fire 15K system controller (SC), the system uses several more software components to accomplish the DR operation than it does if you initiate the DR operation from the domain.

For more information about the software components that reside on the SC, refer to the *System Management Services (SMS) 1.0 Dynamic Reconfiguration User Guide* in the SMS AnswerBook collection.

DR Operations

This section contains descriptions of the four general DR operations: connect, configure, disconnect, and unconfigure. These operations are described from the point of view of the domain. They do not contain information that is specific to the SC.

Connect Operation

During the connect operation, the DR framework attempts to assign the slot to the domain if a system board is available and if it is not part of any logical domain. After the slot has been assigned, the DR framework communicates with the SC to have the SC power on and test the board. After the board has been tested, the DR framework requests the SC to electronically connect the board to the system bus, making the board part of the physical domain. The operating system then probes the components on the board.

The syntax of the `cfgadm(1M)` command to connect a system board is as follows:

```
% cfgadm -c connect SBx
```

where *x* represents the number (0 to 17) of a particular board.

The syntax of the `cfgadm(1M)` command to connect an I/O board is as follows:

```
% cfgadm -c connect IOx
```

where *x* represents the number (0 to 17) of a particular board.

The states and conditions for the attachment point before a board is inserted are:

- Receptacle state—Empty
- Occupant state—Unconfigured
- Condition—Unknown

After a board is physically inserted, the states and conditions are:

- Receptacle state—Disconnected
- Occupant state—Unconfigured
- Condition—Unknown

After the attachment point is logically connected, the states and conditions are:

- Receptacle state—Connected
- Occupant state—Unconfigured
- Condition—OK

Configure Operation

During the configure operation, the DR framework attempts to connect the receptacle if the receptacle state is disconnected. It then walks the tree of devices that was created during the connect operation. The DR framework creates the Solaris device tree nodes and attaches the device drivers if necessary.

The CPUs are added to the CPU list, and the memory is initialized. The memory is then added to the system memory pool. The CPUs and the memory are ready for use after the configure function has been completed successfully.

For I/O devices, you must use the `mount(1M)` and the `ifconfig(1M)` commands before the devices can be used.

Use the `cfgadm(1M)` command to configure a CPU is as follows:

```
% cfgadm -c configure SBx::cpu
```

where *x* represents the board number (0 to 17) for a particular board and *y* represents the CPU number (0 to 3).

The syntax of the `cfgadm(1M)` command to configure memory is as follows:

```
% cfgadm -c configure SBx::memory
```

where *x* represents the board number (0 to 17) for a particular board. For memory, the command applies to all of the memory on the system board.

The syntax of the `cfgadm(1M)` command to configure an I/O board is as follows:

```
% cfgadm -c configure IOx::pciy
```

where *x* represents the board number (0 to 17) for a particular board and *y* represents the PCI number (0 to 3).

The states and conditions for a configured attachment point are:

- Receptacle state—Connected
- Occupant state—Configured
- Condition—OK

Now the system is also aware of the usable devices that reside on the board, and all devices can be mounted or configured for use.

If the configure operation fails for any reason, the states and conditions will still change to configured. This creates a special situation where the board is partially configured. In this situation, only an unconfigure operation is allowed.

Disconnect Operation

During a disconnect operation, the DR framework communicates with the SC to program the interconnect so that the system board is removed from the physical domain. It then attempts to perform the tasks related to the unconfigure operation.

A board can be in the disconnected state without being powered off. However, the board must be powered off and in the disconnected state before you can remove it from the slot.

The syntax of the `cfgadm(1M)` command to disconnect the board is as follows:

```
% cfgadm -c disconnect SBx
```

where *x* represents the board number (0 to 17) for a particular board.

Before the board is disconnected, the states and conditions are:

- Receptacle state—Connected
- Occupant state—Configured
- Condition—OK

After the board is disconnected, the states and conditions are:

- Receptacle state—Disconnected
- Occupant state—Unconfigured
- Condition—Unknown

Unconfigure Operation

The unconfigure operation consists of two separate operations, depending on the presence of permanent memory. If the system board hosts permanent memory, then the DR framework must move it from the specified board to another board in the domain. See the Section “Nonpermanent and Permanent Memory” on page 11 for more information about boards that host permanent memory.

Nonpermanent Memory

If the reconfiguration coordination manager (RCM) is present, then the DR framework informs the RCM about the DR operation. The RCM informs client applications, and the client applications perform the preparatory tasks such as stopping the usage of devices. The clients communicate their readiness to the RCM, and the RCM communicates its readiness to the DR framework. Depending on the responses, the DR framework continues or aborts the operation, reporting an error to the user.

During the unconfigure function, the DR framework unconfigures the board resources from the Solaris Operating Environment and leaves the board in the disconnected state.

If the board hosts CPUs and/or memory, the DR framework removes them from the Solaris Operating Environment, making them unusable to the operating system. If the board is an I/O board, the DR framework detaches the device drivers.

Permanent Memory

The following paragraphs and examples specifically illustrate the unconfigure operation for permanent memory.

In the following code examples, the permanent memory on board 0 must be moved to another board in the domain. Thus, board 0 is the source, and board 1 is the target.

For brevity, the CPU information has been removed from the code examples. The operation is started with the following command:

```
% cfgadm -c unconfigure -y SB0::memory &
```

First, the memory on board 1 in the same address range as the permanent memory on board 0 must be deleted. During this phase, the source board, the target board, and the memory attachment points are marked as busy. You can display the status with the following command:

```
% cfgadm -a -s cols=ap_id:type:r_state:o_state:busy SB0 SB1
```

Ap_Id	Type	Receptacle	Occupant	Busy
SB0	CPU	connected	configured	y
SB0::memory	memory	connected	configured	y
SB1	CPU	connected	configured	y
SB1::memory	memory	connected	configured	y

After the memory has been deleted on board 1, it is marked as unconfigured. The memory on board 0 remains configured, but it is still marked as busy, as in the following example.

Ap_Id	Type	Receptacle	Occupant	Busy
SB0	CPU	connected	configured	y
SB0::memory	memory	connected	configured	y
SB1	CPU	connected	configured	y
SB1::memory	memory	connected	unconfigured	n

The memory from board 0 is then copied to board 1. After it has been copied, the occupancy state for the memory is switched. The memory on board 0 becomes unconfigured, and the memory on board 1 becomes configured. At this point in the process, only board 0 remains busy, as in the following example.

Ap_Id	Type	Receptacle	Occupant	Busy
SB0	CPU	connected	configured	y
SB0::memory	memory	connected	unconfigured	n
SB1	CPU	connected	configured	n
SB1::memory	memory	connected	configured	n

After the entire process has been completed, the memory on board 0 remains unconfigured, and the attachment points are not busy, as in the following example.

Ap_Id	Type	Receptacle	Occupant	Busy
SB0	CPU	connected	configured	n
SB0::memory	memory	connected	unconfigured	n
SB1	CPU	connected	configured	n
SB1::memory	memory	connected	configured	n

The permanent memory has been moved, and the memory on board 0 has been unconfigured. At this point, you can initiate a new status change operation on either board.

Software Components

This section contains descriptions of the software components that reside on the domain and make DR operations possible. It does not contain descriptions of all of the DR components on the Sun Fire 15K platform. Refer to the *System Management Services (SMS) 1.0 Dynamic Reconfiguration User Guide* for descriptions of the software components that reside on the Sun Fire 15K system controller (SC).

Domain Configuration Server

The domain configuration server (DCS) is a daemon process that runs on a Sun Fire 15K domain and is started by `inetd(1M)` when the first DR request is received. A single instance of the DCS runs in each domain on the platform. The DCS accepts DR requests from the domain configuration agent (DCA) that runs on the SC. After the DCS accepts a DR operation, it performs the request and returns the results to the DCA. Refer to the *System Management Services (SMS) 1.0 Dynamic Reconfiguration User Guide* for more information about the DCA.

DR Driver

The DR driver consists of a platform independent driver, named `dr`, and a platform specific module, named `drmach`. The DR driver uses standard features of the Solaris Operating Environment whenever possible to control DR operations, and it calls the platform specific module as needed. The DR driver is responsible for creating minor nodes in the file system that are used as attachment points for DR operations.

Reconfiguration Coordination Manager

The reconfiguration coordination manager (RCM) is a daemon process that coordinates DR operations on resources that are present in the domain. The RCM daemon uses generic application program interfaces (APIs) to coordinate DR operations between DR initiators and RCM clients.

The RCM consumers consist of DR initiators, which request DR operations, and DR clients, which react to DR requests. Normally, the DR initiator is the configuration administration command, `cfgadm(1M)`. However, it can also be a GUI such as SunTM Management Center.

The DR clients can be software layers that export high-level resources comprised of one or more hardware devices (for example, multipathing applications), or they can be applications that monitor DR operations (for example, Sun™ Management Center). Finally, DR clients can be entities on a remote system, such as the system controller on a server.

System Events Framework

DR uses the Solaris system events framework to notify other software entities of the occurrence of changes that result from a DR operation. DR accomplishes this by sending DR events to the system event daemon, `syseventd`, which, in turn, sends the events to the subscribers of DR events. For more information about the system events daemon, refer to the `syseventd(1M)` man page.

DR User Interfaces on the Domain

This chapter contains a description of the user interfaces on the domain. The interfaces include the command and options that are available to the user and the files that are of importance.

DR Commands and Options on the Domain

On the domain, the `cfgadm(1M)` command is used to perform DR operations. The operations are passed to the `libcfcadm(3LIB)` library interface, which dynamically loads a hardware-specific library plugin that performs the DR operations.

The `sbd.so.1` hardware-specific plug-in provides dynamic reconfiguration functionality for connecting, configuring, unconfiguring, and disconnecting system boards, enabling you to connect or disconnect a system board from a running system without having to reboot the system.

The `cfgadm(1M)` command resides in `/usr/sbin` (see the `cfgadm(1M)` man page for more information).

Each board slot appears as a single attachment point in the device tree. You can view the type, state, and condition of each component and the state and condition of each board slot by using the `-a` option.

State Change Functions

State change functions that change the state of a board slot or a component on the board can be issued concurrently against any attachment point. Only one state change operation is permitted at a given time. A Υ in the Busy field indicates an operation is in progress.

The following list contains the functions that change the state:

- `configure`
- `unconfigure`
- `connect`
- `disconnect`

Availability Change Functions

Functions that change the availability of a board can be issued concurrently against any attachment point. Only one availability change operation is permitted at a given time. A Υ in the Busy field indicates an operation is in progress.

The following list contains the functions that change the availability:

- `assign`
- `unassign`

Condition Change Functions

Functions that change the condition of a board slot or a component on the board can be issued concurrently against any attachment point. Only one condition change function is permitted at a given time. These functions also change the information string in the `cfgadm -l` output. A Υ in the Busy field indicates an operation is in progress.

The following list contains the functions that change the condition:

- `power on`
- `power off`
- `test`

Options and Operands

The following options and operands are supported:

Options and Operands	Specifies
-c connect <i>ap_id</i>	A change in the receptacle state to connected
-c disconnect <i>ap_id</i>	A change in the receptacle state to disconnected
-c configure <i>ap_id</i>	A change in the occupancy state to configured
-c unconfigure <i>ap_id</i>	A change in the occupancy state to unconfigured
-x <i>function ap_id</i>	A platform specific function
-t <i>ap_id</i>	The system board to be tested
-l <i>ap_id</i>	The state, status, and condition of system boards and components to be displayed

The *ap_id* operand corresponds to the attachment point of the system board or component.

DR Domain Procedures

This chapter contains a description of how to use the `cfgadm(1M)` command on the Sun Fire™ 15K domain. It also contains a description of attachment points and procedures for displaying board status.

Attachment Points

Before you use the `cfgadm(1M)` command, you must understand the syntax for attachment points on the Sun Fire 15K platform. There are two types of attachment points: single attachment points for board slots and dynamic attachment points for components. Attachment points created by the DR driver have a physical and logical path.

Physical attachment points for system boards take the following form:

```
/devices/pseudo/dr@0:slotx.y
```

where *x* represents the expander position (0 to 17) for a particular board and *y* represents the slot number (0 or 1).

Logical attachment points for system boards take the following form:

```
SBx (for CPUs and memory) or IOx (for I/O boards)
```

where *x* represents the board number (0 to 17) for a particular board.

Dynamic attachment points refer to components on the system boards, such as CPUs, memory, I/O devices. The attachment points are created by the DR driver. Refer to the `dr(7D)` man page for more details.

Displaying Board Status

The `cfgadm(1M)` command displays information about boards and slots. Refer to the `cfgadm_sbd(1M)` man page for options to this command.

Basic Status Display

Many operations require that you specify the system board names. To obtain these system names, type:

```
% cfgadm -a -s "select=class(sbd)"
```

You must have domain privileges to get information about the specified domain.

The following display shows the typical output.

Ap_Id	Type	Receptacle	Occupant	Condition
SB0	CPU	connected	configured	ok
SB0::cpu0	cpu	connected	configured	ok
SB0::memory	memory	connected	configured	ok
I01	PCI	connected	configured	ok
I01::pci0	io	disconnected	unconfigured	failed

Detailed Status Display

For a more detailed status report, use the `cfgadm(1M)` command with the `-v` option. The `-v` option turns on expanded (verbose) descriptions. Besides the basic information, such as the attachment point ID, the receptacle state, and occupant state, and the board status, the expanded status report also includes the date when the board was configured into the domain, the type of board, the activity state, and the physical attachment point.

Removing a Board

This section contains procedures that describe how to remove a CPU/Memory and an I/O board.

▼ To Remove a Board

To perform the following steps, you must have domain administrator privileges.

1. **Log in to the domain.**
2. **Use the `cfgadm(1M)` command with the `-l` option to determine the attachment point for the board.**
3. **Stop all activity on the board.**

You must halt all accesses by other CPU/Memory boards and prevent any further use until the board is replaced by using the appropriate Solaris commands.

4. **Verify that the board does not have bound processes running.**

If a process is bound to a CPU, the board cannot be removed until the process is unbound. Refer to the `pbind(1M)` man page for more information.

5. **Disconnect the board:**

```
% cfgadm -c disconnect SBslot_number
```



Caution – Do not remove a board until it is disconnected, or the board will be damaged.

▼ To Remove an I/O Board

To remove an I/O board, you must first terminate the usage of the board. This section contains a procedure for both phases of the process. To complete the steps in this procedure, you must have domain administrator privileges.

1. **Log in to the domain.**

2. Check the status of the board.

```
% cfgadm -a -s "select=class(sbd)"
```

3. If the system is using multipathing software:

- a. Switch all board functions to the alternate board.
- b. Remove any multipathing databases and/or private regions.
- c. Wait until all of the alternate paths are functioning before proceeding.

4. Unmount file systems, including metadevices that have a board resident partition (for example: `umount /partition`).

5. If the board contains Sun RSM Array™ 2000 controllers, take the controllers off line, using the `rm6` or `rdacutil` commands.

6. Remove disk partitions from the swap configuration.

7. Either kill any process that directly opens a device or raw partition, or direct such a process to close the open device on the board.

8. If a detach-unsafe device is present on the board, close all instances of the device and use `modunload(1M)` to unload the driver.



Caution – Unmounting file systems may affect NFS client systems.

9. Disconnect the board.

```
% cfgadm -v -c disconnect IOSlot_number
```

Adding a Board

When installing a board, you should consider the following points:

- Do not use a board that is bad or suspected to be unreliable. It can crash the system.
- The board type and option cards must be supported by DR.

▼ To Install a Board

To install a board from the domain, it must already be assigned to the domain, or it must be in the available component list (ACL). Refer to the *System Management Services (SMS) 1.0 Administrator Guide* for information on how to assign boards or to update the ACL.

1. Verify that the selected board slot can accept a board.

```
% cfgadm -a -s "select=class(sbd)"
```

The states and conditions should be:

- Receptacle state—Empty
- Occupant state—Unconfigured
- Condition—Unknown

or

- Receptacle state—Disconnected
- Occupant state—Unconfigured
- Condition—Unknown

2. Configure the board.

```
% cfgadm -v -c configure SBslot_number
```

There is a delay before the message appears. The system is testing the board during the delay. The states and conditions for a connected and configured attachment point should be:

- Receptacle state—Connected
- Occupant state—Configured
- Condition—OK

Now the system is aware of the usable devices on the board and the devices can be used.

Index

A

- adding a board, 34
- ADR on I/O boards, 10
- attachment point
 - as an operand, 29
- attachment points
 - description of, 6
 - syntax, 31
- Automatic DR, 4
- availability changes, 28
- available component list, 9

B

- boards
 - conditions, 7, 15
 - hot plug, 8
 - receptacle states, 15
 - slot, 27
 - states, 15
 - testing, 29

C

- cfgadm(1M)
 - adding a board, 34
 - attachment points, 7
 - displaying board status, 32
 - functions, 8
 - removing a board, 33

- commands, 27
- component
 - conditions, 18
 - states, 17
 - types, 9
- condition changes, 28
- condition models, 15
- configure operation, 8, 21
- configured state, 16, 17
- connect operation, 8, 20
- connected state, 16
- CPUs
 - detachability, 5
 - suspending, 5
 - types, 9

D

- DCA, 25
- DCS, 25
- DDI_DETACH, 5
- detachability, 5
- disconnect operation, 8, 22
- disconnected state, 16
- disk
 - mirroring, 10
 - partitions, 10
- displaying board status, 32
- domain configuration agent, 25
- domain configuration server, 25

- domains
 - description of, 8
 - logical, 9
 - physical, 9
 - platform configuration database, 9
- DR
 - clients, 26
 - concepts, 5
 - driver, 25
 - operations, 7
- dr.conf file, 6
- drivers
 - unsafe, 6
- drmach, 25
- DR-unsafe device, 6
- dual-inline-memory-modules, 11
- dynamic reconfiguration (DR), 3
 - an illustration of concepts, 12
 - command line interface, 3
 - GUI, 4
 - illustration of concepts, 8
 - limitations, 1
- dynamic system domains, 8

E

- empty
 - slots, 9
 - state, 16

F

- failed condition, 17, 18
- functions
 - availability, 28
 - conditions, 28
 - state changes, 28
- fuser(1M), 10

H

- hardware-specific plugin, 27
- hot swap, 4
- hot-plug boards, 8

I

- I/O devices
 - detachability, 5
 - suspending, 5
 - suspend-safe, 6
 - types, 9
 - with ADR, 10

L

- limitations
 - dynamic reconfiguration (DR), 1
- logical attachment point, 7
- logical domain, 9

M

- memory
 - correctable errors, 11
 - permanent, 11
 - target constraints, 11
- memory types, 9
- multipathing databases, 10

N

- ndd(1M), 11
- Nonpermanent memory, 11
- nonpermanent memory, 23

O

- occupant, 6
- occupant states, 16
- ok condition, 17, 18
- options, 27
- options and operands, 29

P

- permanent memory, 11, 23
- physical attachment point, 7

- physical domain, 9
- platform configuration database, 9
- platform specific functions, 29
- populated slots, 9
- prtdiag command, 15

Q

- Quiescence, 5

R

- raw partitions, 10
- RCM consumers, 25
- receptacle, 6
- receptacle state, 15
- reconfiguration coordination manager, 4, 25
- record-stop dumps, 12
- removing a board, 33
- RSM 2000 controllers, 10

S

- showdevices(1M)
 - with I/O devices, 10
- slot numbers, 7
- slots, 9
- State change functions, 28
- state models, 15
- status display
 - basic, 32
 - detailed, 32
- suspend-safe devices, 6
- sysevent, 4
- syseventd, 26
- system events framework, 26

U

- unconfigure operation, 8, 22
- unconfigured state, 16, 17
- unknown condition, 17, 18

- unsafe devices, 6
- unusable condition, 17
- user interfaces, 27

V

- volume managers, 10

