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

Please
Recycle

Adobe PostScript™

Contents

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 documentation 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)" describes the functional limitations that you must observe in this version of the DR software.

Chapter 2 "Introduction to DR on the Sun Fire 15K Server" gives an overview of DR on the Sun Fire 15K

Chapter 3 "DR State and Condition Models" describes all the possible conditions and states in which system boards, board slots, and components can be before, during, and after DR operations.

Chapter 4 "DR Operations and Software Components on the Domain" provides an overview of DR operations performed on Sun Fire 15K domains.

Chapter 5 "DR User Interfaces on the Domain" describes commands and options that are available on Sun Fire 15K domains.

Chapter 6 "DR Domain Procedures" describes the steps you take to perform DR operations from Sun Fire 15K domains using the `cfgadm`(1M) command.

Using UNIX Commands

This document does not contain complete information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices. See the Solaris software documentation that you received with your system for this information.

Typographic Conventions

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

Shell Prompts

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

Related Documentation

Application	Title	Part Number
User information	*System Management Services (SMS) 1.2 Dynamic Reconfiguration User Guide*	816-4279
Reference	*System Management Services (SMS) 1.2 Reference Manual*	816-2528
SMS Administration	*System Management Services (SMS) 1.2 Administrator Guide*	816-2527

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 (816-4278-10) of your document in the subject line of your email.

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

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

DR Limitations

- 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 attempt to perform DR operations on a Max CPU system board that resides in a 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 system.

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 physically remove boards from, or add them to, the server.

Where You Execute DR Commands

You can execute DR operations either 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(1M) command, which runs the configuration administration program that you use to perform DR operations on a Solaris domain.

Graphical User Interface (GUI)

The optional Sun Management Center 3.0 Platform Update 4 software, which is designed for use on Sun Fire 15K systems, provides features such as domain management, as well as a graphical user interface (GUI) where you perform DR operations. If you prefer to use a graphical user interface instead of a command line interface, use the Sun Management Center 3.0.

To use the Sun Management Center 3.0 Platform Update 4 software, you must attach the system controller (SC) 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 Platform Update 4 software, refer to the *Sun Management Center 3.0 User's Guide*, shipped with the Sun Management Center 3.0 Platform Update 4 software. For instructions on how to connect the system controller to a network connection on the system controller (SC) board, see your system's 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, called *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; and clean-up after a DR operation. The sysevent facility enables applications to register for system events and receive notifications of those events. The automatic DR framework interfaces with the RCM and the sysevent 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 continues to run. DR then configures the resources on the board into the domain. If you use DR 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 boards 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) 1.2 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.

Note – If you are unsure whether a device is detachable, consult your Sun service representative.

Quiescence

During an unconfigure operation on a system board that has 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 domain must cease during this 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.
- A device exists that cannot be paused by the operating environment.

Note – Real-time processes do not prevent quiescence.

The conditions that cause processes to fail to suspend are generally temporary. Examine the reasons for the failure, and if the operating environment encountered a failure to suspend a process, simply try the operation again.

Suspend-Safe and Suspend-Unsafe Devices

When DR suspends the operating environment, all 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 remains in quiescence. A driver is suspend-safe if it supports operating environment quiescence (if it can be suspended and then resumed). 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 `dr.conf` file resides in the following directory: `/platform/SUNW,Sun-Fire-15000/kernel/drv/`. 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 that identifies the active, unsafe driver. You must manually stop usage of the device by performing one or more of the following tasks:

- Kill processes that are using the device.
- Unload the driver using the `modunload`(1M) command.
- Disconnect the cables (depending on the type of device).

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

> **Note –** If you are unsure whether a device is suspend-safe, contact your Sun service representative.

Attachment Points

An *attachment point* is a collective term that refers to a board slot, a system board installed in the slot, and any devices connected to the board. DR can display the status of the board, the board slot, and the attachment point. The term *occupant* refers to the combination of a board and its attached devices.

- A board slot (sometimes referred to as a *receptacle*) has the ability to electrically isolate the occupant from the host machine. The software can put a board slot into power-off mode.
- Board slots can be named according to slot numbers, or can be anonymous (for example, a SCSI chain).
- An occupant I/O board includes any external storage devices connected by interface cables.

There are two types of attachment points:

- A *physical* attachment point describes the software driver and location of the slot. Examples of physical attachment point names are:

```
/devices/pseudo/dr@0:SBx  (for a CPU/memory board in slot 0)
-OR-
/devices/pseudo/dr@0:IOx  (for an I/O board or Max CPU board in slot 1)
```

Where, *x* represents the number of a particular board (from 0 through 17).

> **Note –** CPU/memory boards are installed only in slot 0. I/O boards and Max CPU boards are installed only in slot 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 CPU/memory boards in slot 0)
-OR-
IOx  (for I/O boards or Max CPU boards in slot 1)
```

To obtain a list of all available logical attachment points, use the cfgadm(1M) command with its -l option.

Conditions and States

A *state* is the operational status of either a board slot or its occupant. 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: connect, configure, unconfigure, and disconnect. A board that is brought into a domain is first connected and then configured. A board that is removed from a domain is first unconfigured and then disconnected.

- During the connect operation, the system provides power to the slot, and the operating environment begins monitoring the board's temperature. For I/O boards, the connect operation is included in the configure operation. You connect a board before you configure it.
- During the configure operation, the operating environment assigns functional roles to the board, and loads device drivers for the board and for devices attached to the board. You always connect a board before you configure it.
- During the unconfigure operation, the system detaches the 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. You unconfigure a board before you disconnect it.
- During the disconnect operation, the system stops monitoring the board and power to the slot is turned off. You always unconfigure a board before you disconnect it.

To power-off a board that is in use (configured), first stop its use (unconfigure it), and then disconnect it from the domain. After a new or upgraded system board is inserted into the slot, connect the board and configure it.

The cfgadm(1M) command can connect and configure (or unconfigure and disconnect) in a single command. To connect and configure a board using a single command, see the section "Adding a Board" on page 35. To unconfigure and disconnect a board using a single command, see the section "Removing a Board" on page 33.

If necessary, each operation (connect, configure, unconfigure, or disconnect) can be performed separately using the cfgadm(1M) command.

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*, which are comprised of logical and physical groupings of system board slots. Each domain is electrically isolated into hardware partitions, which ensures that a problem in one domain does not affect other domains.

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 system board slots are logically partitioned into domains. The domain configuration represents the intended domain configuration. Thus, the configuration can include empty slots and occupied slots.

The number of slots available to a given domain is controlled by an available component list (ACL) that is maintained on the system controller. (Refer to the *System Management Services (SMS) 1.2 Administrator Guide* for more information about the ACL.) 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 assign and connect 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 and *not* be part of a physical domain.

After a domain is booted, the system boards and empty slots can be assigned to (or unassigned from) a logical domain; however, they cannot 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 in whose ACLs they are listed. These boards can be assigned to a domain by the platform administrator. Or, an ACL 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:

Component Type	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 Devices

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

If you need to remove an I/O board with attached devices from a domain temporarily and then re-add it before any other boards with I/O devices are added, reconfiguration is not necessary. In this case, device paths to the board devices remain unchanged.

Potential 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 9 Sun 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 control. 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.

Permanent and Non-permanent Memory

Before you can delete a board, the operating environment must vacate the memory on that board. Vacating a board entails flushing the contents of its non-permanent memory to swap space; and copying the contents of its permanent memory (that is, the kernel and OpenBoot™ PROM software) to another memory board.

To relocate permanent memory, the operating environment on a domain must be temporarily quiesced. The length of the quiescence 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 quiesced; therefore, you should know where permanent memory resides so that you can avoid impacting the operation of the domain significantly. To display the size of permanent memory, use the `cfgadm`(1M) command with its `–v`

option. To vacate a board that has permanent memory, the operating environment must find a sufficiently large block of available memory, called *target* memory, on which to copy the current contents of permanent memory, which is referred to as *source* memory.

Target Memory Constraints

A DR memory operation can be disallowed if the target domain does not have enough memory to hold the contents currently stored in 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.

Note – When a record-stop occurs from a correctable memory error, allow the record-stop dump to complete 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, 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 understand the dynamic 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.
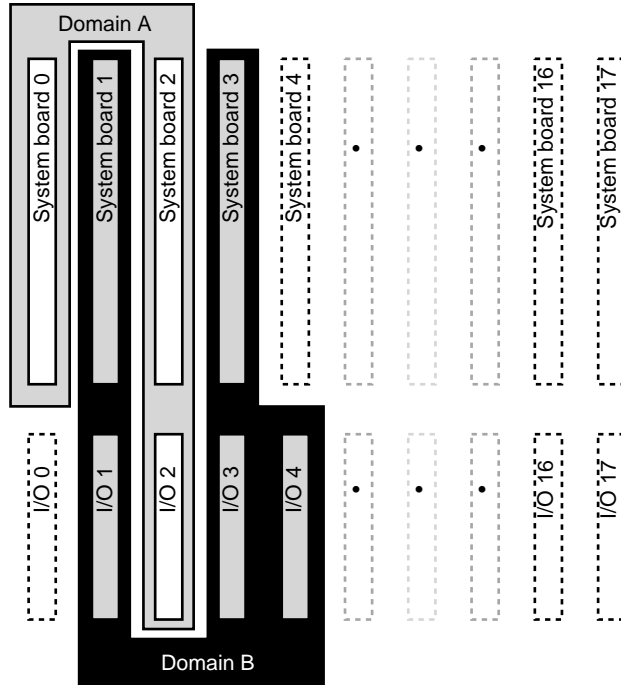
FIGURE 2-1   Domains A & B Before Reconfiguration

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 -c configure SB4 IO0 IO4
```

The following system configuration is the result. Only the way in which the boards are *connected* has changed, not the physical layout of the boards within the cabinet.
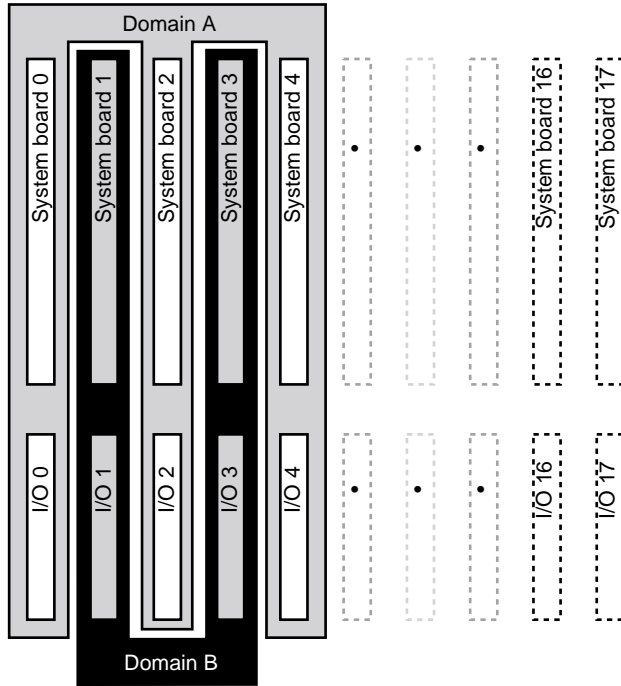
FIGURE 2-2   Domains A & B After Reconfiguration

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: board slots and occupants.

Before you attempt to perform any DR operation on a board or component from the domain, determine its state and condition. To display the type, state, and condition of each component, and the state and condition of each board slot in the domain, use the cfgadm(1M) command with its -la options. See the "Component Types" on page 10 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 SB*xx* or IO*xx*, where the two-digit board numbers include leading zeroes.

Board and Slot States and Conditions

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

Board Slot States

A slot can have one of three states: empty, disconnected, or connected.

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 both 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 when it is in the connected state.

Whenever you insert a board into a slot, the slot's state changes from empty to disconnected. Whenever you remove a board, the slot's 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.

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.

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.

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 Slot States

A component cannot be individually connected or disconnected. Thus, components can have only one slot 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 more software components to accomplish the DR operation than 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) 1.2 Dynamic Reconfiguration User Guide*.

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, and do not contain information that is specific to the SC.

Before You Perform DR Operations

Before you perform DR operations for the first time on a domain after it has been booted, make sure the board is available to the domain. To display a list of boards that are available to the domain, use the `cfgadm` command with its `-l` option.

An error may occur if you attempt to perform DR operations on a board that:

- Is not listed in the domain's ACL and is not assigned to the domain; or
- Is listed in the domain's ACL, but is assigned to another domain.

In either of these cases, the board is not available to the domain. For more information about the ACL refer to the *System Management Services (SMS) 1.2 Administrator Guide*.

Connect Operation

During the connect operation, DR 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, DR requests that the SC power on and test the board. After the board has been tested, DR requests the SC to connect the board electronically to the system bus, which makes the board part of the physical domain. The operating system then probes the components on the board.

To connect a system board through the domain rather than the SC, use the cfgadm(1M) command as follows:

```
# cfgadm -c connect SBx
```

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

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

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

After the 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, DR attempts to connect the board slot if its state is disconnected. It then traverses the tree of devices that was created during the connect operation. (DR creates Solaris device tree nodes and attaches device drivers if necessary.)

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

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

When you configure a board into a domain using cfgadm, the board is automatically connect and configured

CPUs and Memory

To configure a CPU on a system board through the domain rather than the SC, use the cfgadm(1M) command as follows:

```
# cfgadm -c configure SBx::cpuy
```

where *x* represents the board's number (0 to 17) 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 the memory on the system board.

To configure all the CPUs and memory on a system board, use the following command:

```
# cfgadm -c configure SBx
```

I/O Boards

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

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

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

To configure all the busses on an I/O board, use the following command:

```
# cfgadm -c configure IOx
```

After the Configure Operation

The states and conditions for a configured attachment point are:

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

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

Disconnect Operation

During a disconnect operation, DR attempts to perform the tasks related to the unconfigure operation, and requests that the SC program the interconnect to remove the system board from the physical domain.

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).

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 can consist of a single operation or two separate operations, depending on the presence of permanent memory. If the system board hosts permanent memory, before the unconfigure operation DR moves the memory contents from the specified board to available memory on a target board in the domain. See the section "Permanent and Non-permanent Memory" on page 11 for more information about boards that host permanent memory.

Non-permanent Memory

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

During the unconfigure operation, DR unconfigures the board resources from the Solaris operating environment and leaves the board in the unconfigured state.

If the board hosts CPUs and/or memory, DR removes them from the Solaris operating environment, making them unusable to the operating system. If the board is an I/O board, DR 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, board 1. Board 0 is the source board, and board 1 is the target board.

For brevity, the CPU information has been removed from the code examples. On the domain, the unconfigure operation is started with the cfgadm(1M) command:

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

First, a block of memory on the target board that resides in the same address range as the permanent memory on the source board 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 the target board, it is marked as unconfigured. The memory on the source board 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 the source board is then copied to the target board. After it has been copied, the occupancy state for the memory is switched. The memory on the source board becomes unconfigured, and the memory on the target board becomes configured. At this point in the process, only the source board 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 the source board 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 the source board has been unconfigured. At this point, you can initiate a new status change operation on either board.

Software Components

This section describes the software components that reside on the domain and make DR operations possible. However, it does not contain descriptions of all of the DR components on the Sun Fire 15K platform. Refer to the *System Management Services (SMS) 1.2 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 remote DR request is received. A single instance of the DCS runs in each domain on the Sun Fire 15K. 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) 1.2 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 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 Sun 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)
- Applications that monitor DR operations (for example, Sun Management Center)
- 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 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 describes the user interfaces on the Sun Fire 15K domain, which include the commands and options that are available to the user, and important files.

DR Commands and Options on the Domain

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

The sbd.so.1 hardware-specific plug-in provides DR functionality: connecting, configuring, unconfiguring, and disconnecting system boards, which enables you to connect or disconnect a system board from a running system without having to reboot the system.

The cfgadm(1M) command resides in the /usr/sbin directory. (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 cfgadm(1M) command with its –a option.

State Change Functions

State change functions, which change the state of a board slot or a component on the board, can be issued against any attachment point.

The following are state change functions:

- configure
- unconfigure
- connect
- disconnect

Availability Change Functions

Availability change functions, which change the availability of a board, can be issued against any attachment point.

The following are availability change functions:
- assign
- unassign

Condition Change Functions

Condition change functions, which change the condition of a board slot or a component on the board, can be issued against any attachment point.

The following are condition change functions:
- poweron
- poweroff
- test

Options and Operands

The following options and operands are supported for the functions shown:

Options and Operands	Specifies
-c connect *ap_id*	A change in the receptacle state to connected
-c disconnect *ap_id*	A change in the receptacle state to disconnected
-c configure *ap_id*	A change in the occupancy state to configured
-c unconfigure *ap_id*	A change in the occupancy state to unconfigured
-x *function ap_id*	A platform-specific function
-t *ap_id*	The system board to be tested
-l *ap_id*	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 describes how you use the cfgadm(1M) command on the Sun Fire 15K domain to perform DR operations. It also describes attachment points and procedures for displaying the status of system boards.

Attachment Points

Before you use the cfgadm(1M) command, make sure you understand the syntax for attachment points on the Sun Fire 15K platform. There are physical and logical attachment points. In addition, single attachment points are used for board slots, and dynamic attachment points are used 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:SBx  (for CPU/memory boards)
-OR-
/devices/pseudo/dr@0:IOx  (for I/O boards)
```

where *x* represents the board number (0 through 17).

Logical attachment points for system boards take the following form:

```
SBx  (for CPU/memory boards)
-OR-
IOx  (for I/O boards)
```

where *x* represents the board number (0 through 17).

Dynamic attachment points refer to components (CPUs and memory) on system boards and I/O devices on I/O boards. 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)"
```

Note – The cfgadm command displays information only about those boards that are assigned to the domain; or those boards that appear in the available component list (ACL) for the domain and are not assigned to any other domain.

The following output is typical:

```
Ap_Id              Type       Receptacle      Occupant       Condition
SB0                CPU        connected       configured     ok
SB0::cpu0          cpu        connected       configured     ok
SB0::memory        memory     connected       configured     ok
IO1                HPCI       connected       configured     ok
IO1::pci0          io         disconnected    unconfigured   failed
```

Detailed Status Display

For a more detailed status report, use the cfgadm(1M) command with its -v option, which turns on expanded (verbose) descriptions. In addition to basic information such as the attachment point ID, receptacle and occupant states, and 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 describes how to remove a CPU/Memory and an I/O board.

▼ To Remove a CPU/Memory 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 its** -l **option to determine the attachment point for the board.**

3. **Stop all activity on the board.**

   You must halt all access by other CPU and 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. **Unconfigure and disconnect the board using the following single command:**

   ```
   # cfgadm -v -c disconnect SBx
   ```

   where x represents the board number (0 through 17).

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

▼ To Remove an I/O Board

To remove an I/O board, you must first stop all usage of the board. 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. **If any process directly opens a device or raw partition, either kill the process or direct it 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 IOx
   ```

where *x* represents the board number (0 through 17).

Adding a Board

When installing a board, consider the following points:

- Never use a board that is defective or even 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 perform a board installation from the domain, the board must already be assigned to the domain, or must be in the available component list (ACL). Refer to the *System Management Services (SMS) 1.2 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 either:

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

   -**OR**-

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

2. **Connect and configure the board using a single command.**

   ```
   # cfgadm -v -c configure SBx (CPU/memory board)
   -OR-
   # cfgadm -v -c configure IOx (I/O board)
   ```

   where *x* represents the board number (0 through 17).

   After a short delay during which the system tests the board, a message appears in the domain console log indicating that the components have been configured. 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

configured state, 16, 17
connect operation (DR), 8, 20
connected state, 16
consumers, 26
CPUs
  as component types, 10
  detachability, 5
  suspending, 5

D

DCA, 25
DCS, 25
DDI_DETACH, 5
deleteboard(1M), 3
detachability, 5
device drivers
  unloading with the modunload(1M)
    command, 6
disconnect operation (DR), 8, 22
disconnected state, 16
disk
  mirroring, 10
  partitions, 11
displaying board status, 32
domain configuration agent, 25
domain configuration server, 25
domains
  description of, 9
  logical, 9
  physical, 9
  platform configuration database, 9
DR
  an illustration of concepts, 12
  clients, 26
  command line interface, 3
  commands on the domain, 27
  concepts, 5
    illustrated, 12
  configure operation, 8
  connect operation, 8
  disconnect operation, 8
  driver, 26
  GUI, 4
  hardware-specific plugin and, 27
  limitations, 1

  on I/O devices, 10
  operations, 8
  options on the domain, 27
  unconfigure operation, 8
dr(7D), 32
dr.conf file, 6
drivers
  unsafe, 6
drmach, 26
DR-unsafe device, 6
dual-inline-memory-modules, 12
dynamic reconfiguration (DR), 3
dynamic system domains, 9

E

empty
  slots, 9
  state, 16

F

failed condition, 17, 18
functions
  availability change, 28
  condition change, 28
  state change, 27
fuser(1M), 10

H

hot swap, 4
hot-plug boards, 9

I

I/O devices
  and DR, 10
  as component types, 10
  detachability, 5
  potential problems with, 10
  suspending, 5
  suspend-safe, 6