



---

Sun™ Enterprise™ 10000 Dynamic  
Reconfiguration User's Guide

**Solaris™ 2.6 Hardware: 5/98**

---

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

Part No. 805-3525-10  
May 1998, Revision A

Send comments about this document to: [smcc-docs@sun.com](mailto:smcc-docs@sun.com)

Copyright 1998 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

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

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, SunSoft, SunDocs, SunExpress, Solaris, Solstice, DiskSuite, SunFastEthernet, Ultra Enterprise, Sun Enterprise, AnswerBook, and OpenBoot are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

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

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

---

Copyright 1998 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

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

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, SunSoft, SunDocs, SunExpress, Solaris, Solstice, DiskSuite, SunFastEthernet, Ultra Enterprise, Sun Enterprise, AnswerBook, and OpenBoot sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

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



Please  
Recycle



Adobe PostScript

# Table of Contents

---

**Preface ix**

**1. Introduction to DR 1**

**2. DR Configuration Issues 3**

Memory: `dr-max-mem` 3

`dr-max-mem` With Solaris 2.6 3

`dr-max-mem` With Solaris 2.5.1 4

▼ To Set `dr-max-mem` With Solaris 2.5.1 4

Configuration for DR Detach 6

I/O Devices 6

Memory 7

Pageable and Nonpageable Memory 7

Target Memory Constraints 8

Correctable Memory Errors 8

Swap Space 9

Reconfiguration After a DR Operation 9

When to Reconfigure 10

Disk Devices 11

DR and AP Interaction 11

RPC Time-Out or Loss of Connection	12
System Quiescence Operation	13
Suspend-Safe/Suspend-Unsafe Devices	14
Special Handling for Tape Devices	14
Special Handling of Sun™ StorEdge™ A3000	15

## DR and DDI 15

DR and DDI_DETACH	15
DR and DDI_SUSPEND/DDI_RESUME	16
Adding Suspend-Safe Devices	17
Adding Suspend-Bypass Devices	18
DR Detach-Safe Devices	19

### **3. Using Dynamic Reconfiguration 21**

#### Attaching a System Board 21

Init Attach 21

Complete Attach 22

Attach Buttons 23

▼ To Attach a Board With Hostview 23

▼ To Attach a Board By Using `dr(1M)` 27

#### Detaching a System Board 30

Drain 30

Complete Detach 31

Network Devices 31

Non-Network Devices 32

Processors 33

Finishing the Complete Detach Operation 33

Hostview Detach Buttons 34

▼ To Detach a Board With Hostview 35

- ▼ To Detach a Board By Using `dr (1M)` 38

#### Viewing Domain Information 41

- ▼ To View Domain Information with Hostview 42
- ▼ To Specify How Windows Are Updated 42
- ▼ To View DR CPU Configuration Information 43
- ▼ To View DR Memory Configuration Information 44
- ▼ To View DR Device Configuration Information 47
- ▼ To View DR Device Detailed Information 48
- ▼ To View DR OBP Configuration Information 49
- ▼ To View the Suspend-Unsafe Devices Across the Entire Domain 50

#### Index 1



# Figures

---

- FIGURE 3-1 Attach—Board and Domain Selection Window 24
- FIGURE 3-2 Dynamic Reconfiguration Window With `init attach` Button 25
- FIGURE 3-3 Dynamic Reconfiguration Window 26
- FIGURE 3-4 Detach—Board and Domain Selection Window 36
- FIGURE 3-5 Dynamic Reconfiguration Window for drain Operation 37
- FIGURE 3-6 System Information Buttons 42
- FIGURE 3-7 DR Properties Window 42
- FIGURE 3-8 DR CPU Configuration Window 43
- FIGURE 3-9 DR Memory Configuration Window 45
- FIGURE 3-10 DR Device Configuration Window 47
- FIGURE 3-11 DR Detail Device Window 48
- FIGURE 3-12 DR OBP Configuration Window 50
- FIGURE 3-13 DR Unsafe Devices Window 51





# Preface

---

This book describes the Dynamic Reconfiguration (DR) feature, which enables you to logically attach and detach system boards from the Sun™ Enterprise™ 10000 server while other domains continue running.

---

## Before You Read This Book

This book is intended for the Sun Enterprise 10000 system administrator. Users of the Enterprise 10000 system should have 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* in AnswerBook™ format provided with this system and consider UNIX system administration training.

---

## How This Book Is Organized

This document contains the following chapters:

**Chapter 1 “Introduction to DR”** introduces basic concepts related to the Dynamic Reconfiguration feature.

**Chapter 2 “DR Configuration Issues”** describes how to configure the Dynamic Reconfiguration system before you begin using it.

**Chapter 3 “Using Dynamic Reconfiguration”** describes how to use DR to attach and detach system boards.

---

# Using UNIX Commands

This document does 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 sources for this information:

- AnswerBook online documentation for the Solaris 2.x software environment, particularly those dealing with Solaris system administration
- Other software documentation that you received with your system

---

# Typographic Conventions

The following table describes the typographic changes used in this book.

TABLE P-1 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.	% <b>su</b> Password:
AaBbCc123	Book titles, new words or terms, words to be emphasized. Command-line variable; replace with a real name or value.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be <i>root</i> to do this. To delete a file, type <code>rm filename</code> .

---

# Shell Prompts

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

**TABLE P-2** 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

DR is normally started from the Hostview GUI in the SSP environment. See the following documentation for more information about DR:

- *SMCC Release Notes Supplement*, a printed document in your media box, part number 805-3537-10. This document may contain Dynamic Reconfiguration Release Notes.
- *Sun Enterprise 10000 SSP User's Guide*, part number 805-2955-10
- *Sun Enterprise 10000 SSP Reference Manual*, part number 805-3362-10

---

# Ordering Sun Documents

SunDocs<sup>SM</sup> is a distribution program for Sun Microsystems technical documentation. Contact SunExpress for easy ordering and quick delivery. You can find a listing of available Sun documentation on the World Wide Web.

**TABLE P-3** SunExpress Contact Information

Country	Telephone	Fax
Belgium	02-720-09-09	02-725-88-50
Canada	1-800-873-7869	1-800-944-0661
France	0800-90-61-57	0800-90-61-58
Germany	01-30-81-61-91	01-30-81-61-92
Holland	06-022-34-45	06-022-34-46
Japan	0120-33-9096	0120-33-9097
Luxembourg	32-2-720-09-09	32-2-725-88-50
Sweden	020-79-57-26	020-79-57-27
Switzerland	0800-55-19-26	0800-55-19-27
United Kingdom	0800-89-88-88	0800-89-88-87
United States	1-800-873-7869	1-800-944-0661

**World Wide Web:** <http://www.sun.com/sunexpress/>

---

---

# Sun Documentation on the Web

The docs.sun.com web site enables you to access Sun technical documentation on the World Wide Web. You can browse the docs.sun.com archive or search for a specific book title or subject at <http://docs.sun.com>.

---

# Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at [smcc-docs@sun.com](mailto:smcc-docs@sun.com). Please include the part number of your document in the subject line of your email.



# Introduction to DR

---

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

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

You can execute DR operations through the Hostview GUI (see `hostview(1M)`) or through the `dr(1M)` shell application. DR supports the following operations:

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

While DR operations are being performed within a domain, `dr_daemon(1M)` (see the *Solaris Reference Manual for SMCC-Specific Software*) and the operating system write messages regarding the status or exceptions to the domains' syslog message buffer (`/var/adm/messages`) and the SSP message files (`$(SSPOPT)/adm/host/messages` and `$(SSPOPT)/adm/messages`). In addition to the status and exception information displayed by Hostview and the `dr(1M)` shell application, the `dr_daemon(1M)` and operating system messages are useful for determining the status of DR requests.

---

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

---



## DR Configuration Issues

---

This chapter describes how to configure a domain for all DR operations and capabilities. The DR features are enabled only when the OpenBoot™ PROM (OBP) environment variable `dr-max-mem` is set to a non-zero value. The sections in this chapter include more information about `dr-max-mem`.

---

**Note** – DR features are disabled on domains that have less than 512-Mbytes of memory.

---



---

**Caution** – Be careful when choosing the slot into which a board is inserted to prevent disk controller renumbering. For more information, see “Reconfiguration After a DR Operation” on page 9.

---

---

### Memory: `dr-max-mem`

The value for `dr-max-mem` depends on the version of the Solaris operating environment (2.5.1, 2.6, or higher) that is running in the domain. This section includes information on both versions.

#### `dr-max-mem` With Solaris 2.6

With Solaris 2.6 or higher, the memory-related data structures are dynamically allocated during the DR Attach operation. They are also dynamically removed during the DR Detach operation; therefore, `dr-max-mem` with Solaris 2.6 becomes an on/off switch. With `dr-max-mem` set to zero, DR operations are disabled. This is true no matter which version of Solaris is running in the domain.

## dr-max-mem With Solaris 2.5.1

The kernel has a number of memory-related data structures such as page structures, which are statically allocated at boot time and are based on the amount of physical memory in the domain at that time. Use DR Attach to dynamically add a board and its physical memory after the domain is booted. This extra memory can be supported by the kernel only if enough memory data structures are allocated at boot time to support it. These structures cannot be added dynamically after boot time.

To reserve enough memory data structures to support DR Attach operations, each domain supports the OBP environment variable, `dr-max-mem`, which the kernel reads at boot time. `dr-max-mem` specifies the maximum number of megabytes to which the domain can grow without requiring a reboot. Each domain has its own unique copy of `dr-max-mem`.

### ▼ To Set `dr-max-mem` With Solaris 2.5.1

1. Calculate the optimum value for `dr-max-mem` by combining the amount of memory most likely to be added during all DR Attaches and the current amount of memory present in the domain and setting `dr-max-mem` to the total.

Note that if `dr-max-mem` is too large relative to the memory in the domain, its size can impact the performance of the operating system. Therefore, the operating system limits the maximum value of `dr-max-mem` at boot time, as follows:

TABLE 2-1 `dr-max-mem` Maximum Values

Current Physical Memory	<code>dr-max-mem</code> Maximum Value
256-Mbytes	0
512-Mbytes	16-Gbytes
1024-Mbytes	32-Gbytes
2048-Mbytes	64-Gbytes

If the value of `dr-max-mem` is smaller than the amount of physical memory present when the domain is booted, the operating system sets its working copy of `dr-max-mem` to the current memory size. You cannot attach more memory, but you can detach, then re-attach memory. The maximum amount of memory you can re-attach in this manner is the amount present when the domain was booted. Note that the OBP variable `dr-max-mem` is not modified in this situation.



**Caution** – Set `dr-max-mem` high enough so that all anticipated new memory can be dynamically attached, but no higher. If you set it too low and later attach a board whose memory combined with domain memory exceeds the value of `dr-max-mem`,

the memory on that board will not be attached. If you set the value of `dr-max-mem` too high, you over-allocate data structures, which can waste available memory and adversely affect system performance. If you set it to zero, the DR functions are disabled.

---

`dr-max-mem` must be set *before* the domain is booted.

2. Set the `dr-max-mem` environment variable by bringing up the OBP prompt for the domain and typing the following command:

```
ok# setenv-dr dr-max-mem NNN
```

where `NNN` is the number of megabytes of memory to be supported by the domain after the boards are attached. The value of `dr-max-mem` persists across domain reboots, and is only applicable to that particular domain. This value will apply to all boot environments for the domain.

---

**Note** – Once you have set the `dr-max-mem` value for a domain, that value remains the same no matter which Solaris boot disk you select.

---

If the `dr-max-mem` variable is non-zero, the following messages are displayed at boot time in the domain's syslog message buffer (`/var/adm/messages`):

```
DR: current memory size is XXX MBytes  
DR: capacity to allow an additional YYY MBytes of memory
```

In this message, `XXX` represents the amount of physical memory available to the operating system and is effectively the same as the operating system variable, `physinstalled`. `YYY` is the difference between `dr-max-mem` and `XXX`.

When a board with memory is successfully attached or detached, another message is displayed:

```
DR: capacity to allow an additional ZZZ MBytes of memory
```

In this message, `ZZZ` represents the updated amount of memory that can still be attached.

---

# Configuration for DR Detach

This section describes how to configure DR before you perform a detach operation.

---

**Note** – The DR Detach feature requires that the OBP variable `dr-max-mem` is set to a non-zero value. This setting is required at the time the domain is booted.

---

## I/O Devices

The DR Detach feature relies on the Alternate Pathing (AP) feature or Solstice™ DiskSuite™ mirroring when used to detach a board that hosts I/O controllers that are attached to vital system resources. Currently, AP and Solstice DiskSuite do not work together; however, AP does work with Veritas Volume Manager. If, for example, the root or `/usr` partition is on a disk attached to a controller on the board, the board cannot be detached unless there is a hardware alternate path to the disk—and AP has been configured to take advantage of it—or the disk is mirrored. The alternate path or the mirrors must be hosted by other boards in the system. The same applies to network controllers. The board that hosts the Ethernet controller that connects the SSP to the Enterprise 10000 platform cannot be detached unless an alternate path exists to an Ethernet controller on another board for this network connection.

The domain swap space should be configured as multiple partitions on disks attached to controllers hosted by different boards. With this kind of configuration, a particular swap partition is not a vital resource because swap partitions can be added and deleted dynamically (see `swap(1M)` for more information).

---

**Note** – When memory (`swapfs`) or disk swap space is detached, there must be enough memory or swap disk space remaining in the domain to accommodate currently running programs.

---

A board that is hosting non-vital system resources can be detached whether or not there are alternate paths to the resources. All of the board's devices must be closed before the board can be detached; all of its file systems must be unmounted; and, its swap partitions must be deleted. You may have to kill processes that have open files or devices, or place a hard lock on the file systems (using `lockfs(1M)`) before you unmount the boards. There is a domain disruption penalty associated with the detach operation.

All I/O device drivers involved with I/O devices on the board(s) must support the `DDI_DETACH` option in the driver detach entry point. This option releases all system resources associated with that device or adapter.

## Memory

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

```
mem_board_interleave_ok
```

If `mem_board_interleave_ok` is present, you may not be able to detach a board that contains memory.

## Pageable and Nonpageable Memory

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

You can use the `dr(1M)` command `drshow(1M)` to determine if a board's memory is pageable or nonpageable:

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

Similarly, you can determine if a board's memory is pageable by looking at the DR Memory Configuration window, which is available when you perform a detach operation within Hostview. The DR Memory Configuration window is described in "Viewing Domain Information" on page 41.

The kernel and OBP load into the lowest physical address space, which generally is on the lowest numbered system board in the domain. Exceptions to this rule do exist, so you should always use the `drshow(1M)` command to check the board's memory.

## Target Memory Constraints

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

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

If no target board is found, the detach operation is refused, and DR displays a warning message on the system console:

```
Jul 28 06:00:00 unix: WARNING:dr_build_adg_detach_list:no target
memory board found
```

## Correctable Memory Errors

Correctable memory errors (CEs) indicate that a system board's memory (that is, one or more of its Dual Inline Memory Modules, or DIMMs, or portions of the hardware interconnect) may be faulty and need replacement. When the SSP detects correctable memory errors, it initiates a record-stop dump to save the diagnostic data, which can interfere with a DR detach operation. 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 detach operation.

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

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

```
SSP% edd_cmd -x stop
```

This command suspends all event detection on all of the domains.

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

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

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

4. Perform the DR detach operation.

5. Enable event detection:

```
SSP% edd_cmd -x start
```

## Swap Space

The domain swap configuration consists of the swap devices and `swapfs` (memory). The domain must contain enough swap space so that it can flush pageable memory. For example, if you want to remove 1-Gbyte of memory from a 2-Gbyte domain, you will need 1-Gbyte of swap space, depending on the load. Insufficient swap space prevents DR from completing the detach of a board that contains memory. If this happens, the memory drain phase does not complete, so you must abort the detach operation.

---

# Reconfiguration After a DR Operation

This section describes how to reconfigure your domain after you have attached or detached a system board.

The DR user interface enables you reconfigure the domain after a DR Attach or DR Detach operation. The reconfiguration sequence is the same as the reconfiguration boot sequence (`boot -r`):

```
drvconfig; devlinks; disks; ports; tapes;
```

When you execute the reconfiguration sequence after you attach a board, device path names not previously seen by the domain are written to the `/etc/path_to_inst` file. The same path names are also added to the `/devices` hierarchy, and links to them are created in the `/dev` directory.

## When to Reconfigure

You should reconfigure the domain if any of the following conditions occur:

- **Board Addition** – When you add a board to a domain, you must execute the reconfiguration sequence to configure the I/O devices that are associated with the board.
- **Board Deletion** – If you remove a board that is not to be replaced, you may, but do not have to, execute the reconfiguration sequence to clean up the `/dev` links.
- **Board Replacement** – If you remove a board then reinsert it in a different slot or if you replace a board with another board that has different I/O devices, you must execute the reconfiguration sequence to configure the I/O devices that are associated with the board. However, if you replace a board with another board that hosts the *same* set of I/O devices, inserting the replacement into the *same* slot, you do not need to execute the reconfiguration sequence. But, be sure to insert a replacement board into the same slot that was vacated to retain the original `/dev` link names.



# Disk Devices

Disk controllers are numbered consecutively as the `disks(1M)` program encounters them. All disk partitions are assigned `/dev` names according to the disk controller number that `disks(1M)` assigns. For example, all disk partitions that are accessible using disk controller 1 are named `/dev/dsk/c1tLdMsN`

where:

`c1` is the disk controller number

`L`, in most cases, corresponds to the disk target number

`M` corresponds to the logical unit number, and

`N` corresponds to the partition number.

When the reconfiguration sequence is executed after a board is detached, the `/dev` links for all of the disk partitions on that board are deleted. The remaining boards retain their current numbering. Disk controllers on a newly inserted board are assigned the next available lowest number by `disks(1M)`.

For example, suppose the system has four system boards numbered 0 to 3, and you detach boards 1 and 2, which are then removed from the system. Your service provider repairs board 2 and reinserts it, and you attach it. If you now execute `disks(1M)`, controller numbers from board 1 are reassigned to controllers on board 2 if the old board 1 controller numbers are the next available lowest numbers.

---

**Note** – The disk controller number is part of the `/dev` link name used to access the disk. If that number changes during the reconfiguration sequence, the `/dev` link name also changes. This change may affect file system tables and software, such as Solstice DiskSuite, which use the `/dev` link names. Update `/etc/vfstab` files and execute other administrative actions to change the `/dev` link names.

---

---

## DR and AP Interaction

---

**Note** – Alternate Pathing (AP) is not supported in Solaris 2.6 Hardware: 3/98.

---

DR notifies the AP subsystem when system boards are attached, detached, or placed in the drain state. In addition, DR queries AP about which controllers are in the AP database and their status (active or inactive). This communication occurs between the `dr_daemon(1M)` and `ap_daemon(1M)`. If the `ap_daemon(1M)` is not present,

an error message is placed in the domains' syslog messages buffer and DR operations continue without error. To disable this interaction, use the `-a` option when you invoke `dr_daemon(1M)`. See the `dr_daemon(1M)` man page in the *Solaris Reference Manual for SMCC-Specific Software*.

If you are using AP version 2.1, the operating system automatically switches off the active disk controllers on outgoing boards during the complete-detach phase of DR. If you are using AP version 2.0, you need to manually switch off the active disk controllers before you start the complete-detach phase. For Solaris 2.6, you must upgrade to AP version 2.1. For more information about DR and AP interaction, see the *Solaris 2.6 Hardware 3 Beta on Sun Enterprise Servers AnswerBook*.

---

## RPC Time-Out or Loss of Connection

The `dr_daemon(1M)`, which runs in each domain, communicates with Hostview and the `dr(1M)` shell application (both of which run on the SSP) by way of Remote Procedure Calls (RPCs). If an RPC time-out or connection failure is reported during a DR operation, check the domain. The daemon must be configured in the domain's `/etc/inetd.conf` file. The following line (which appears on a single line) must be present in the file:

```
300326/4 tli rpc/tcp wait root /usr/platform/sun4ul/ sbin/  
dr_daemon dr_daemon
```

If the DR daemon is configured in `/etc/inetd.conf`, kill the `dr_daemon(1M)` if it is currently running. In addition, send a HUP signal to the `inetd(1M)` daemon to cause it to re-read the `inetd.conf(4)` configuration file:

```
# kill dr_daemon_pid  
# kill -HUP inetd_pid
```

In the first command, `dr_daemon_pid` is the process ID of the DR daemon. In the second command, `inetd_pid` is the process ID of the `inetd(1M)` daemon. You can check `/var/adm/messages` for possible error messages from `inetd(1M)` if it's having trouble starting the `dr_daemon(1M)`. The DR daemon executable file should exist in the `/usr/platform/sun4ul/sbin/dr_daemon` directory.

At this point you should try the DR operation again, starting from the beginning.

---

# System Quiescence Operation

During a DR Detach operation on a system board with nonpageable OBP or kernel memory, the operating system is briefly quiesced; that is, all operating system and device activity on the domain centerplane must cease during the critical phase of the operation. The quiescence only affects the target domain; other domains in the system are not affected.

Before it can quiesce, the operating system must temporarily suspend all processes, processors, and device activities. If the operating system cannot quiesce, it displays its reasons, which may include the following:

- Real-time processes are running in the domain.
- A device that cannot be quiesced by the operating system (that is, a suspend-unsafe device) is open.

The conditions that cause processes not to suspend are generally temporary in nature. You can retry the operation until the quiescence succeeds.

A quiescent failure due to real-time processes or open suspend-unsafe devices is known as a forcible condition. You have the option of performing either a retry or forced retry. When you force the quiesce, you give the operating system permission to continue with the quiescence even if forcible conditions are still present.



---

**Caution** – Exercise care when using the *force* option.

---

If a real-time process is running, determine if suspending the process would produce an adverse effect on the functions performed by the process. If not, you can force the operating system to quiesce. (To force a quiescence, you can either click the **Force** button within Hostview as described in “To Detach a Board With Hostview” on page 35, or enter the `complete_detach` command with the `force` option within the `dr(1M)` shell application. Otherwise, you can abort the operation and try again later.

If any suspend-unsafe device is open and cannot be closed, you can manually suspend the device, and then force the operating system to quiesce. After the operating system resumes, you can manually resume the device (see “Suspend-Safe/Suspend-Unsafe Devices” on page 14).

If the operating system fails to quiesce, pay close attention to the reasons for the failure. If the operating system encountered a transient condition—a failure to suspend a process—you can try the operation again. If, however, the condition(s) requires your approval (for example, a real-time process is running) or intervention (for example, a suspend-unsafe device is open), you can force the operating system to quiesce.

## Suspend-Safe/Suspend-Unsafe Devices

A suspend-safe device is one that does not access the domain centerplane (for example, it does not access memory or interrupt the system) while the operating system is quiesced. A driver is considered suspend-safe if it supports operating system quiescence (suspend/resume) and guarantees that when a suspend request is successfully completed, the device that the driver manages will not attempt to access the domain centerplane, even if the device is open when the suspend request is made. All other I/O devices are suspend-unsafe when open.

---

**Note** – At the time of this printing, the drivers released by Sun Microsystems that are known to be suspend-safe are `sd`, `isp`, `esp`, `fas`, `sbus`, `pci`, `pei-pci`, `qfe`, `hme` (SunFastEthernet™), `nf` (NPI-FDDI), `qe` (Quad Ethernet), `le` (Lance Ethernet), the SSA drivers (`soc`, `pln`, and `ssd`), and the Photon drivers (`sf`, `socal`, `ses`). The known suspend-unsafe driver is the tape-related driver (`st`). This list will change over time. To add a driver to this list, see “Adding Suspend-Safe Devices” on page 17.

---

The operating system refuses a quiesce request if a suspend-unsafe device is open. If you can manually suspend the device, you can force the operating system to quiesce. To manually suspend the device, you may have to close the device by killing the processes that have it open, ask users not to use the device, or disconnect the cables. For example, if a device that allows asynchronous unsolicited input is open, you can disconnect its cables prior to quiescing the operating system, preventing traffic from arriving at the device and the device from accessing the domain centerplane. You can reconnect the cables after the operating system resumes. If you cannot make a device suspend its accesses to the domain centerplane, you should not force the operating system to quiesce. Doing so could cause a domain to crash or hang. Instead, postpone the DR operation until the suspend-unsafe device is no longer open.



---

**Caution** – If you attempt a forced quiesce operation while activity is occurring on a suspend-unsafe device, the domain may hang. However, if the domain hangs, it will not affect other domains that are running on the Enterprise 10000 system.

---

## Special Handling for Tape Devices

Although all tape devices are suspend-unsafe, all SCSI tape devices that are natively supported by Sun Microsystems are detach-safe (see `st(7D)` for a list of natively-supported drives). If a system board that you are detaching contains a natively-supported tape device and if the tape device is not in use, then you can detach the board. If you want to use a tape device that is not natively supported by Sun, you

can use it, but you should make it detach-safe. To ensure correct input/output and DR operations, you need to make a suitable entry in `/kernel/drv/st.conf` with the `ST_UNLOADABLE (0x0400)` flag set in the entry (see `st(7D)` for more information). After you update `st.conf`, you must reboot the domain to process the new entry.

The sequential nature of tape devices prevents them from being reliably suspended in the middle of an operation, and then resumed. Therefore, all tape drivers are suspend-unsafe and cannot be quiesced. Before you execute a DR operation that quiesces the operating system, make sure that all of the tape devices in the domain are closed or are not in use. If a tape device is detected in the domain, you can either force the quiescent operation, or you can unload the `st(7D)` driver by using the `modunload(1M)` command prior to the quiescent operation.

## Special Handling of Sun™ StorEdge™ A3000

The Sun StorEdge A3000 (formerly known as the RSM 2000) has dual controller paths with automatic load balancing and automatic failover. To detach a system board that has one or both of the StorEdge A3000's controllers, the controllers on the board that being detached must be idle or offline. You can take these controllers offline manually by using the `rm6` or `rdacutil` programs before you attempt to detach the system board.

---

## DR and DDI

Not all drivers support the Enterprise 10000 system's Dynamic Reconfiguration (DR) features. To support DR, a driver must be able to perform two basic DDI/DKI (Device Driver Interface/Device Kernel Interface) functions, `DDI_DETACH` and `DDI_SUSPEND/DDI_RESUME`. These two functions impact DR in different ways.

### DR and `DDI_DETACH`

You can detach a system board that hosts a device only if the driver for that device supports the `DDI_DETACH` interface, or is not currently loaded. `DDI_DETACH` provides the ability to detach a particular instance of a driver without impacting other instances that are servicing other devices. A driver that supports `DDI_DETACH` is called *detach-safe*; a driver that does not support `DDI_DETACH` is called *detach-unsafe* (see “DR Detach-Safe Devices” on page 19).

Detaching a detach-unsafe driver that is loaded involves the following process.

- Stopping all usage of the controller for the detach-unsafe device and all other controllers of the same type on all of the boards in the domain.

Since the detach-unsafe driver must be unloaded, you must stop use of that controller type on *all* of the system boards in the domain. The remaining controllers can be used again after the DR Detach is complete.

- Using standard Solaris interfaces to manually close and to unload all such drivers on the board.

See `modload(1M)` in the *SunOS Reference Manual*.

- Detaching the system board in the normal fashion.

If you cannot accomplish the above process, you can reboot your domain with the board blacklisted (see `blacklist(4)`), so the board can be removed later.

---

**Note** – Many third-party drivers (those purchased from vendors other than Sun Microsystems) do not support the standard Solaris `modunload` interface. Conditions that invoke the functions occur infrequently during normal operation and the functions are sometimes missing or work improperly. Sun Microsystems suggests that you test these driver functions during the qualification and installation phases of any third-party device.

---

## DR and DDI\_SUSPEND/DDI\_RESUME

To perform a DR Detach of a board that contains nonpageable OBP or kernel memory, the domain must be quiesced. Memory can be detached only when all of the drivers throughout the entire domain (not just on the board being detached) either support the `DDI_SUSPEND/DDI_RESUME` driver interface, or are closed. Drivers that support these DDI functions are called *suspend-safe*; drivers that do not are called *suspend-unsafe* (see “Adding Suspend-Safe Devices” on page 17). Note that a quiesce is required only when detaching a board that contains nonpageable memory.

The most straightforward way to quiesce a domain is to close any suspend-unsafe devices. For each network driver you must execute the `ifconfig(1M)` command with its `down` parameter, then again with its `unplumb` parameter (see `ifconfig(1M)` for more information).

---

**Note** – It should be possible to unplumb all network drivers. However, this action is rarely tested in normal environments and may result in driver error conditions. If you use DR, Sun Microsystems suggests that you test these driver functions during the qualification and installation phases of any *suspend-unsafe* device.

---

If the system refuses to quiesce because a *suspend-unsafe* driver is open, you can force the operating domain to quiesce. Doing so forces the operating system to permit the detach. Note that, although a detach can be forced to proceed when there are open *suspend-unsafe* devices in the system, it is not possible to force a detach when a *detach-unsafe* device resides on the board and its driver is loaded.

To successfully force the operating system to quiesce, you must manually quiesce the controller. Procedures to do that, if any, are device-specific. The device must not transfer any data, reference memory, or generate interrupts during the operation. Be sure to test any procedures used to quiesce the controller while it is open prior to executing them on a production system.



---

**Caution** – Using the *force* option to quiesce the operating system, without first successfully quiescing the controller, can result in a domain failure and subsequent reboot.

---

## Adding Suspend-Safe Devices

Before each Enterprise 10000 system is shipped, Sun Microsystems configures the DR driver (*dr*) to recognize those devices that support `DDI_SUSPEND/DDI_RESUME`; that is, the devices that can be safely quiesced. The Note in “Suspend-Safe/Suspend-Unsafe Devices” on page 14 lists the known suspend-safe and suspend-unsafe devices at the time this guide was produced.

If you want to add a device to your system and the device and its driver support `DDI_SUSPEND/DDI_RESUME`, configure the DR driver to recognize the device as suspend-safe by placing an entry in the `/etc/system` file. This file enables you to append the list already maintained in the operating system. No harm results from a device being listed multiple times. If you are not sure whether a device supports `DDI_SUSPEND/DDI_RESUME`, ask your service provider or the manufacturer of the device.

If a listed device is open when a quiescence is requested, the device is suspended and resumed prior to the quiescence. Tape devices are *not* suspend-safe; do *not* append such devices to the suspend-safe list in the `/etc/system` file.

---

**Note** – In an earlier release, the suspend-safe list was called the `dr_safe_list`. You can use the old name, but when the `dr` module is first loaded, the following messages are displayed:

```
NOTICE: dr: using old style safe/bypass list (dr_safe_listx)
NOTICE: dr: next time use new style (suspend_safe_listx)
```

---

You can add new devices that support quiescing to the `/etc/system` file with the following format, where device names represent device driver module names:

```
set hswp:suspend_safe_list1="device1 device2... devicen"  
set hswp:suspend_safe_list2="device1 device2... devicen"  
set hswp:suspend_safe_list3="device1 device2... devicen"  
set hswp:suspend_safe_list4="device1 device2... devicen"  
set hswp:suspend_safe_list5="device1 device2... devicen"
```

---

**Note** – The `/etc/system` file can contain up to five suspend-safe strings, each no more than 128 characters long.

---

You should add devices to the suspend-bypass list only if they do not perform I/O operations and do not support `DDI_SUSPEND/DDI_RESUME`.

## Adding Suspend-Bypass Devices

The Enterprise 10000 system has a preset list of devices that it ignores during the quiescent process, making no attempt to quiesce them. These devices, which include pseudo devices, do not perform I/O operations and do not need to be suspended during the quiesce.



---

**Caution** – Do not add suspend-unsafe devices to the suspend-bypass list.

---

You can add devices to the `/etc/system` file which do not support quiescing, but which can be safely ignored during the quiesce process. Do so in the following format, where device names represent device-driver module names:

```
set hswp:suspend_bypass_list1="device1 device2... devicen"  
set hswp:suspend_bypass_list2="device1 device2... devicen"  
set hswp:suspend_bypass_list3="device1 device2... devicen"  
set hswp:suspend_bypass_list4="device1 device2... devicen"  
set hswp:suspend_bypass_list5="device1 device2... devicen"
```

---

**Note** – The `/etc/system` file can contain up to five suspend-bypass strings, each no more than 128 characters long.

---



# DR Detach-Safe Devices

Before each Enterprise 10000 system is shipped, DR is configured to recognize those devices that can be safely detached. A driver is safe for detaching if it fully supports the `DDI/DKI DDI_DETACH` function in the driver's detach entry point. Normally, such DR-capable drivers must also support `DDI_SUSPEND` and `DDI_RESUME`, as described in "DR and `DDI_SUSPEND/DDI_RESUME`" on page 16. However, some exceptions do exist, such as tape devices that can be detach-safe while they are inherently suspend-unsafe.

If you want to add a device to your system and the device and its driver can be safely detached, be sure to add the device name to the detach-safe list in the `/etc/system` file. This file appends the list already maintained in the system. No harm results when a device is listed multiple times. If you are not sure whether a device can be safely detached, ask your service provider.

If a DR Detach operation fails because the board hosts a device that is not included in the detach-safe list and if the corresponding driver is loaded, the system displays a message similar to the following:

```
WARNING: DR: driver (xxx) not known to support DDI_DETACH
```

where `xxx` is the name of the driver module as it would reside under `/kernel/drv` and named in `/etc/name_to_major`.

You can add new devices that support DR Detach to the `/etc/system` file with the following format, where device names represent device driver module names:

```
set dr:detach_safe_list1="device1 device2 ... devicen"  
set dr:detach_safe_list2="device1 device2 ... devicen"  
set dr:detach_safe_list3="device1 device2 ... devicen"  
set dr:detach_safe_list4="device1 device2 ... devicen"  
set dr:detach_safe_list5="device1 device2 ... devicen"
```

---

**Note** – The `/etc/system` file can contain up to five detach-safe strings, each no more than 128 characters long.

---



# Using Dynamic Reconfiguration

---

---

## Attaching a System Board

---

**Note** – This section gives a broad overview of the actions that occur when you execute DR Attach. For step-by-step instructions, see “To Attach a Board With Hostview” on page 23.

---

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

---

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

---

Prior to attaching a board, diagnostics are run on the board, requiring that at least one processor be present on the board and not be blacklisted. After you’ve selected an eligible board and a target domain, the DR Attach operation proceeds through two operations: Init Attach and Complete Attach.

### Init Attach

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

- Adds the board to the target domain's board list in the `domain_config(4)` file on the SSP.
- Runs `hpost -H` on the board to configure it. `hpost` isolates the board on the Enterprise 10000 system centerplane by placing it into a single-board hardware domain (see `hpost(1M)`).
- Runs `obp_helper -H` which loads `download_helper` to the board, and takes the processors on the board out of reset mode, allowing them to spin in `download_helper`.
- Reconfigures the centerplane and board domain mask registers, placing the board in the target hardware domain.

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

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

After the Init Attach phase is completed, the OBP board configuration can be displayed to confirm which devices are present on the board. You can then enter the Complete Attach phase or you can abort the operation.

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

## Complete Attach

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

After a board is successfully attached, you have the option of reconfiguring the I/O devices. See “Reconfiguration After a DR Operation” on page 9 for more information. This operation can take several minutes to complete.

# Attach Buttons

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

- `init attach` – Begins the attach operation (see “Init Attach” on page 21). Once the operation has completed successfully, the label on this button changes to `complete`.
- `complete` – Completes the attach operation (see “Complete Attach” on page 22).
- `reconfig` – Automatically reconfigures the device directories in the domain. You may want to run the reconfiguration operation after attaching a board (see “Reconfiguration After a DR Operation” on page 9).
- `abort` – Cancels the attach operation. This button is enabled after the Init Attach operation has been successfully completed (see “Init Attach” on page 21).
- `dismiss` – Terminates the step that is currently in progress, but leaves the board in its current state (Present, Init Attach, In Use). You can remove the DR attach window by choosing `dismiss` at any point during the attach operation. The `dismiss` button terminates any work being done on the SSP for the attach operation. For example, if `hpost(1M)` is running when you click `dismiss`, that `hpost(1M)` process is terminated. Note that `dismiss` does not terminate work being done on the host by way of RPCs to the `dr_daemon(1M)`. Once an RPC is initiated, the host completes the RPC regardless of whether the calling program is waiting for the RPC to finish.

The host `dr_daemon(1M)` keeps track of the progress of the attach operation. Once the Init Attach operation completes successfully, it remembers this state. Therefore, you can dismiss the window, then return to the DR operation later and complete or abort the attach.

- `help` – Accesses online information regarding DR Attach operations.

## ▼ To Attach a Board With Hostview

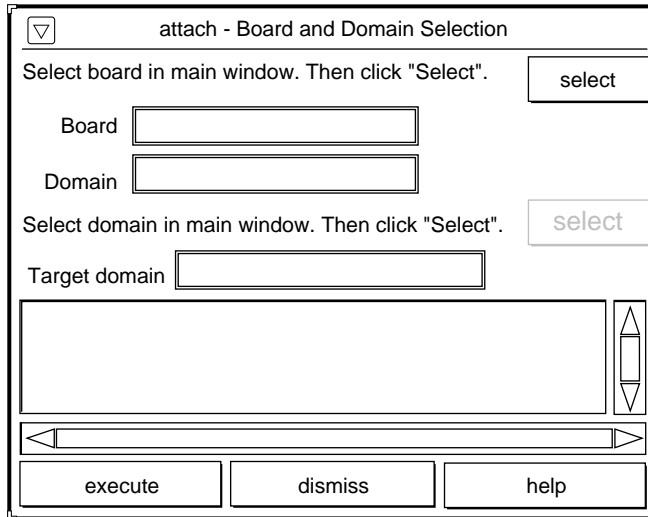
---

**Note** – Before you perform the following steps, you should read “Attaching a System Board” on page 21.

---

1. **From Hostview, choose** Configuration ► Board ► Attach.

The Attach—Board and Domain Selection window is displayed (FIGURE 3-1).



**FIGURE 3-1** Attach—Board and Domain Selection Window

**2. Select the board that you want to attach in the main Hostview window (if that board is not already selected).**

**3. Click the top `Select` button.**

The `Board` and `Source Domain` fields are automatically filled in for you. (You can also manually edit those fields.)

**4. In the main Hostview window, select the domain to which you want to attach the board.**

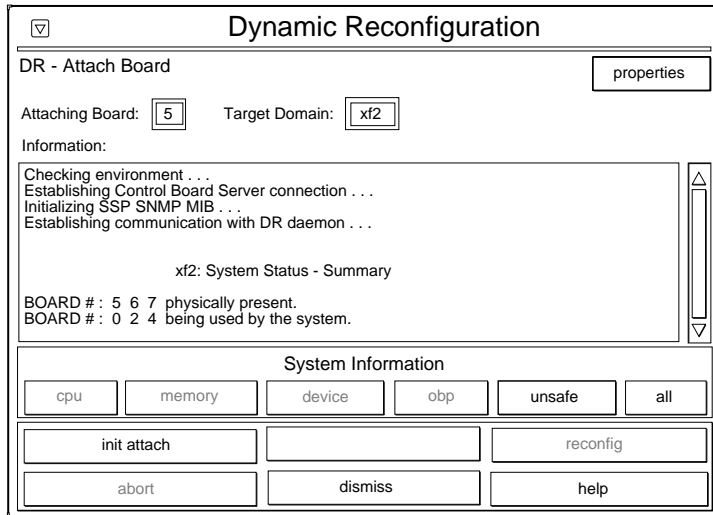
You can select any board that is currently a member of that domain.

**5. Click the bottom `Select` button.**

The `Target Domain` field is automatically filled in for you. (You can also manually edit that field.)

**6. Click the `Execute` button.**

If any errors occur, the error messages appear in the main Hostview window. Otherwise, the Dynamic Reconfiguration window is displayed with the `init attach` button visible (FIGURE 3-2).



**FIGURE 3-2** Dynamic Reconfiguration Window With `init attach` Button

**7. Click the `Init Attach` button.**

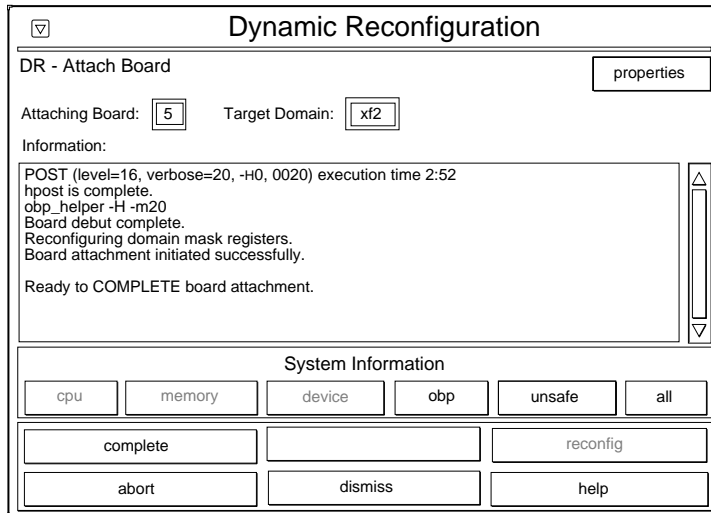
Clicking on the `Init Attach` button begins the first phase of the board attach process. First, the system updates the `SSP domain.config(4)` file by adding the system board to the target domain’s board list. Next, the system uses `hpost(1M)` to self-test the system board. After the self-test is complete, the board is made visible to the running target domain by merging it into the hardware domain via the centerplane and the system board hardware register modification. Finally, during the conclusion of the `init attach`, `OBP` probes the new board to discover what CPU, I/O, and memory resources are present on the board. When this phase is finished, the caption on the button changes to `complete`. Before you click the `complete` button, however, you may want to view the domain information to verify that you want to proceed, as described in “Viewing Domain Information” on page 41.

The `Init Attach` operation can take a few minutes to complete. Output from the `hpost(1M)` commands is directed to the Information pane of the Dynamic Reconfiguration window.

If the `Init Attach` fails, look for the cause in the output in the Information pane. Once you have determined the cause, you may want to choose `Init Attach` again.

The window should now appear similar to that shown in FIGURE 3-3, with the `complete` button enabled.

**8. Click the `complete` button.**



**FIGURE 3-3** Dynamic Reconfiguration Window

The `complete` operation normally takes less than one minute to finish. When it has successfully completed, DR displays the following message:

```
Board attachment completed successfully
```

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

You can view the domain information about the newly attached board by using the buttons (CPU, Memory, Device, and so forth), as described in “Viewing Domain Information” on page 41.




---

**Caution** – Before you choose the `reconfig` option, be sure to read “Reconfiguration After a DR Operation” on page 9.

---

**10. Click the `dismiss` button.**

The DR Attach operation is complete.



## ▼ To Attach a Board By Using `dr(1M)`

---

**Note** – The following procedure explains how to attach a board by using `dr(1M)` with SSP version 3.1, or higher. If you are using SSP version 3.0, refer to a previous version of the *Dynamic Reconfiguration User's Guide*.

---

Before you perform the following steps, read “Attaching a System Board” on page 21. The process of attaching a board is very similar whether you use Hostview or `dr(1M)`. The basic concepts are not repeated in this section.

The `dr(1M)` shell was introduced in Chapter 1. A quick reference guide is available in the `dr(1M)` application by using the `help` command.

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

```
% domain_switch xf3
```

2. **Execute the `dr(1M)` command in an SSP Window to bring up the `dr(1M)` prompt.**

In the following example, the target domain is called `xf3`.

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

      xf3: Domain Status - Summary

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

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

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

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

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

Ready to COMPLETE board attachment.
```

### 4. Abort or complete the attach operation.

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

```
dr> drshow board_number OBP
```

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

```
dr> abort_attach board_number
```

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

```
dr> complete_attach 6
Completing attach for board 6.
Board attachment completed successfully.
dr>
```

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

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

```
dr> drshow 6 IO

          SBus Controllers and Devices for Board 6

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

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

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

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

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

```
dr> exit
%
```

The SSP login shell is again displayed.

---

# Detaching a System Board

---

**Note** – This section gives a broad overview of the actions that occur when you execute DR Detach. For step-by-step instructions, see “To Detach a Board With Hostview” on page 35.

---

System boards that are currently being used by the operating system can be detached if they meet the requirements covered in “Configuration for DR Detach” on page 6. Once you select an eligible board, you can detach that board by performing two operations: drain and complete detach.

## Drain

The primary function of the drain operation is to determine how the board’s memory is to be vacated by the operating system and, if required, to select a target memory area for copying the board’s nonpageable memory. If a suitable target memory area is not available when the drain operation is requested, the request is denied. If the drain is rejected for this reason, you can continue to retry until target memory is available. See “Configuration for DR Detach” on page 6.

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

---

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

---

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

If you decide not to proceed with the detach operation, you can abort the operation, and the board’s memory is returned to regular usage. You can also abort the operation during the drain process or after the drain has been completed. If extreme memory pressure exists during the drain, you will see little, or no, progression in the

percentage of drained pages, and you may want to abort the drain and wait until the workload on the domain has decreased, enabling it to accommodate the reduction in memory.

The drain operation is complete when all of the memory pages are drained. You can then complete the detach operation.

## Complete Detach

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

---

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

---

## Network Devices

DR automatically terminates usage of all network interfaces on the board that is being detached. When you complete the detach operation, the `dr_daemon(1M)` identifies all configured interfaces on the board being detached and issues the following `ifconfig(1M)` commands on each such interface.

```
ifconfig interface down
ifconfig interface unplumb
```

Additionally, if FDDI interfaces are detached, DR kills the FDDI network monitoring daemon before you perform the detach operation. DR then restarts it after the detach is complete. Note that the `/usr/sbin/nf_snmd` daemon for `nf` devices is neither started nor stopped when a board that contains a FDDI interface is attached.

DR does not execute these commands on a board that contains a network interface that fits any of the following conditions. In these cases, the detach operation fails and DR displays an error message.

- The interface is the primary network interface for the domain; that is, the interface whose IP address corresponds to the network interface name contained in the file `/etc/nodename`. Note that bringing down the primary network

interface for the domain prevents network information name services from operating, which results in the inability to make network connections to remote hosts using applications such as `ftp(1)`, `rsh(1)`, `rcp(1)`, `rlogin(1)`. NFS client and server operations are also affected.

- The interface is on the same subnet as the SSP host for the system; that is, the subnet of the IP address that corresponds to the SSP host name found in `/etc/ssphostname`. Bringing down this interface interrupts communication between the host and SSP. Since DR operations are initiated on the SSP, control of the detach process would be lost. (Note that the `/etc/ssphostname` file contains the name of the SSP that controls the host; therefore, if you rename the SSP, the `/etc/ssphostname` must be manually updated.)
- The interface is the active alternate for an Alternate Pathing (AP) meta device when the AP meta device is plumbed. Interfaces used by AP should not be the active path when the board is being detached. Manually switch the active path to an interface that is not on the board being detached. If no such path exists, manually execute the `ifconfig down` and `ifconfig unplumb` commands on the AP interface. (To manually switch an active path, use the `apconfig(1M)` command.)

## Non-Network Devices

All non-network devices must be closed before they are detached. In the Hostview device display and in the `drshow(1M)` I/O listing, there is an open count field that indicates how many processes have opened particular devices. To see which processes have these devices open, use the `fuser(1M)` command on the domain.

You must perform certain tasks for non-network devices. Although the following list of tasks implies a sequence of order, strict adherence to the order is not necessary.

1. If the redundancy features of Alternate Pathing or Solstice DiskSuite mirroring are used to access a device connected to the board, reconfigure these subsystems so that the device or network is accessible via controllers on other system boards. Note that for Alternate Pathing 2.1, the system automatically switches the disk devices to an alternate interface if one is available.
2. Unmount file systems, including Solstice DiskSuite meta-devices that have a board-resident partition (for example, `umount /partit`).
3. Remove Alternate Pathing or Solstice DiskSuite databases from board-resident partitions. The location of Alternate Pathing or Solstice DiskSuite databases is explicitly chosen by the user and can be changed.
4. Remove any private regions used by Sun Volume Manager or Veritas Volume Manager. Volume manager by default uses a private region on each device that it controls, so such devices must be removed from volume manager control before they can be detached.

5. Remove disk partitions from the swap configuration by using `swap(1M)`.
6. Either kill any process that directly opens a device or raw partition, or direct it to close the open device on the board.
7. 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.

---

## Processors

The boot processor is responsible for servicing the tick-timer interrupts and for maintaining the netcon BBSRAM buffer. Before detaching a board on which the boot processor resides, the `dr_daemon(1M)` must assign the boot processor role to another active (online) processor.

---

**Note** – When a board is detached, all processes bound to its processors are automatically unbound. You can use `pbind(1M)` to rebind them to other processors.

---

## Finishing the Complete Detach Operation

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

If the board that you want to detach contains nonpageable memory, the complete detach operation may also fail due to quiescence problems, which are described in “System Quiescence Operation” on page 13. After you resolve the quiescent problem, you can again execute the complete detach operation.

If you decide that you do not want to proceed with the detach operation at this time, you can abort the detach. The board's memory is returned to normal usage and detached board devices are reattached. If the system configuration was modified to remove board usage (that is, file systems were unmounted and networks were unplumbed), you must undo these modifications and return the devices to normal operation.

After the board is successfully detached from the operating system, it is isolated from the centerplane by moving it out of the host's hardware domain. In addition, the board list is automatically updated in the SSP `domain_config(4)` file.

You can now attach the board to another domain, power it off, and remove it by way of hot swapping, leaving it in the system unattached, or reattaching it at a later time.

## Hostview Detach Buttons

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

**TABLE 3-1** Hostview Buttons

Button	Description
<code>drain</code>	Drains the memory (see “Drain” on page 30). After the drain operation is finished, the <code>drain</code> button becomes the <code>complete</code> button.
<code>complete</code>	Completes the detach operation after the board has been fully drained (see “Complete Detach” on page 31).
<code>force</code>	Permits you to complete the detach operation by forcibly quiescing the domain (see “System Quiescence Operation” on page 13). If the complete detach operation fails due to a forcible quiesce condition, the <code>force</code> button is enabled.
<code>reconfig</code>	Reconfigures device directories in a domain automatically. You may want to run <code>reconfig</code> after permanently detaching a board. Use <code>reconfig</code> with extreme caution (see “Reconfiguration After a DR Operation” on page 9 for more information).



**TABLE 3-1** Hostview Buttons

Button	Description
abort	Cancels the DR operation, and returns the board to normal operation. This button is enabled after the drain operation starts and remains enabled until the complete detach operation starts. To stop the draining of memory and cancel the detach, choose <code>abort</code> (see “Detaching a System Board” on page 30).
dismiss	<p>Cancels any step that is in progress, and leaves the board in its current state (<code>In Use</code>, <code>drain</code>, <code>Present</code>). At any point during the DR Detach operation you can remove the DR Detach window by choosing <code>dismiss</code> which terminates any work being done on the SSP for the detach operation. Note that <code>dismiss</code> does not terminate work being done on the host through RPC calls to the <code>dr_daemon(1M)</code>. After an RPC call is initiated, the host completes the RPC call regardless of whether Hostview is waiting for the RPC call to finish.</p> <p>The host <code>dr_daemon(1M)</code> keeps track of the progress of the detach operation. Once the <code>drain</code> is started, it remembers this state. Therefore, you can dismiss the window and then return later to either <code>complete</code> or <code>abort</code> the detach operation.</p>
help	Accesses online information regarding DR detach operations.

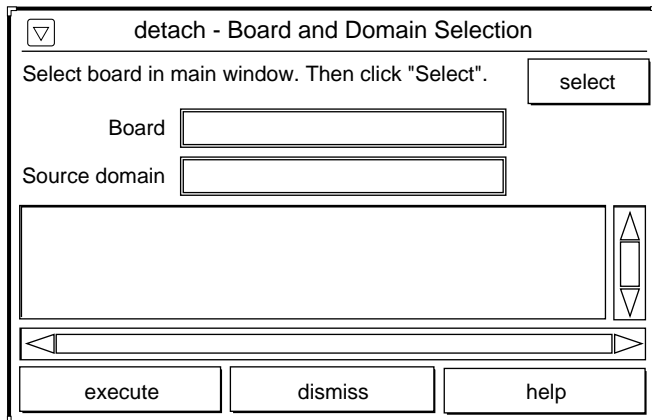
## ▼ To Detach a Board With Hostview

---

**Note** – Before you execute the following steps, read “Detaching a System Board” on page 30.

---

1. **From the Hostview menu, choose** `Configuration` ► `Board` ► `Detach`.  
The Detach—Board and Domain Selection window is displayed (FIGURE 3-4).



**FIGURE 3-4** Detach—Board and Domain Selection Window

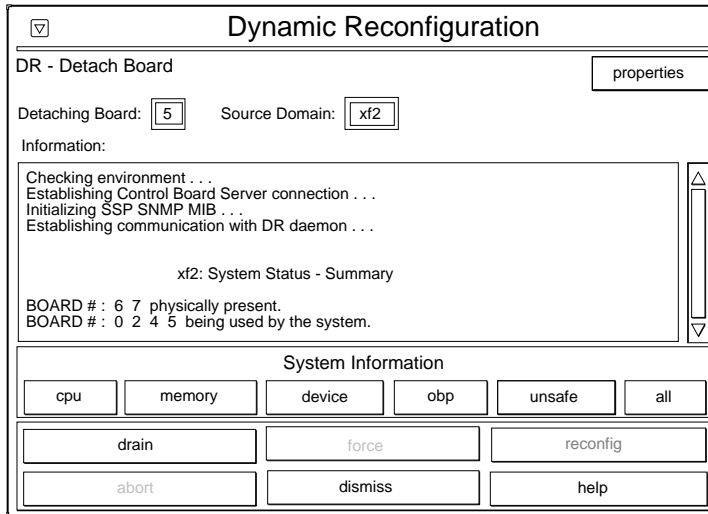
**2. Select the board that you wish to detach in the main Hostview window (if that board is not already selected).**

**3. Click the `select` button.**

The `Board` and `Source domain` fields are automatically filled in for you. (You can also manually edit these fields if you wish.)

**4. Click the `execute` button.**

If the target domain is not currently booted, the detach operation simply manipulates the domain configuration file on the SSP. However, if the domain is currently running, the following window is displayed (FIGURE 3-5).



**FIGURE 3-5** Dynamic Reconfiguration Window for drain Operation

**5. Click the drain button.**

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

The memory drain statistics are automatically updated at periodic intervals if you enable the `Auto Update Domain Information Displays` option in the DR Properties window, as described in “Viewing Domain Information” on page 41.

If the drain operation fails, an explanatory message appears in the Information pane. Once you’ve determined the cause, and corrected it, you can choose `drain` again.

You may proceed to the next step without waiting; it does not depend on completion of the drain.

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

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

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

For more information, see “Complete Detach” on page 31.

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

## 8. Select the `complete` button.

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

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

- All online processors in the domain are on the board being detached.
- The board you want to detach contains the last processor in the “default” processor set. You must add an additional processor from another system board before you retry the Detach operation.
- Primary network interfaces are on the board being detached. You must stop all usage of these networks manually (see “Complete Detach” on page 31).
- All usage of the I/O devices on the board you want to detach has not been stopped. The Information pane identifies the device on which the error was encountered (see “Complete Detach” on page 31).
- The OS quiesce failed. You must determine and resolve the cause of the error (see “System Quiescence Operation” on page 13).

Once you’ve resolved the reason for the failure, you can select either `complete` or `force` to complete the detach. If there are no further problems, the board is detached and reset. When the board is successfully detached, the following message is displayed:

```
Board detachment completed successfully.
```



---

**Caution** – Before you choose the `reconfig` button, you should read “Reconfiguration After a DR Operation” on page 9.

---

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

## ▼ To Detach a Board By Using `dr(1M)`

Before you execute the following steps, read “Detaching a System Board” on page 30. The process of detaching a board is very similar with either Hostview or `dr(1M)`. The basic concepts are not repeated in this section. The `dr(1M)` program was introduced in Chapter 1.

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

**2. Type the `dr(1M)` command in an SSP Window to bring up the `dr(1M)` prompt.**

In the following example, the target domain is called `xf3`.

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

      xf3: Domain Status - Summary

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

### 3. Drain the board.

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

    Bound Processes for Board 6

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

    Active Devices for Board 6

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

    Memory Drain for Board 6 - IN PROGRESS

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

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

dr>
```

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

```
dr> drshow board_number drain
```

---

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

---

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

```
dr> complete_detach 6
Completing detach of board 6
Operating System has detached the board.
Processors on board 6 reset.
Reconfiguring domain mask registers.
Board 6 placed into loopback.
Board detachment completed successfully.
dr>
```

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

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

---

## Viewing Domain Information

Both `dr(1M)` and `Hostview` enable you to display information about the suspend-unsafe devices as well as information about the board selected during DR operations. For `dr(1M)`, this information is accessible by using the `drshow(1M)` command. From `Hostview`, this information is available by clicking the `cpu`, `memory`, `device`, `obp`, and `unsafe` buttons in the attach or detach windows. The informational content is the same for both `dr(1M)` and `Hostview`. Note that the `cpu`, `memory`, and `device` displays are only enabled when the board is attached to the operating system. When the `cpu`, `memory`, and `device` displays are available, they always contain accurate information. The `obp` display shows the information known to OBP, but it is not as detailed as the other three displays. This section shows how to use the displays.

## ▼ To View Domain Information with Hostview

- Click on any of the System Info buttons during the DR operation (FIGURE 3-6).

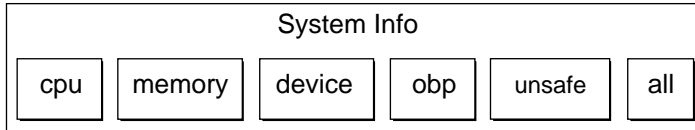


FIGURE 3-6 System Information Buttons

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

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

## ▼ To Specify How Windows Are Updated

1. Click the `Properties` button in the Dynamic Reconfiguration window for `attach` or `detach` (FIGURE 3-7).

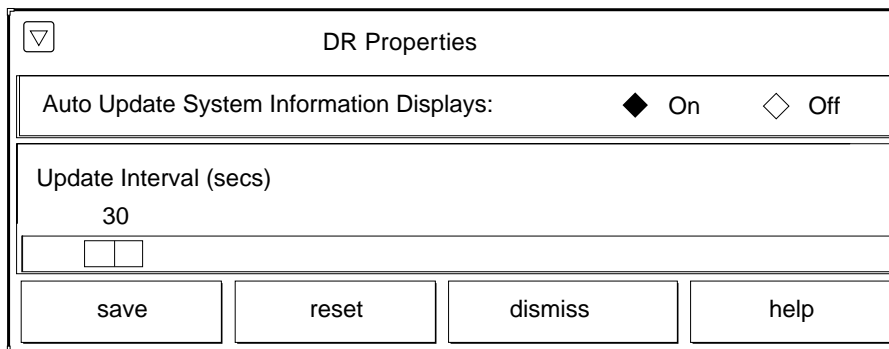


FIGURE 3-7 DR Properties Window

2. To cause displays to be updated, set `Auto Update Domain Information Displays` to `On` (the default).
3. Set the `Update Interval` to a value (in seconds) to determine how often updates occur.

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



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

---

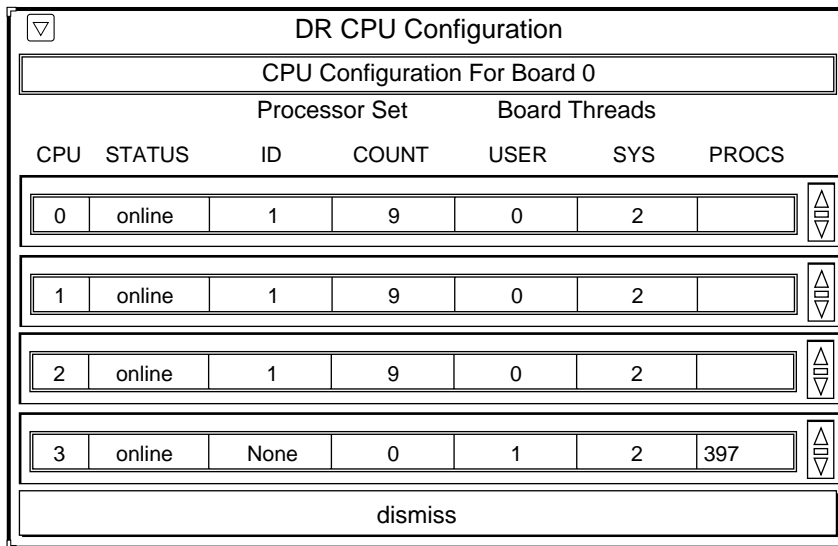
**Note** – When the update interval is set to a low value, such as 10 seconds, and several information windows are displayed, responsiveness of the DR windows may be degraded. This is especially true when device detail windows are displayed. Each time an information window is updated, an RPC is issued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

---

## ▼ To View DR CPU Configuration Information

- Click the `cpu` button.

The DR CPU Configuration window is displayed (FIGURE 3-8).



The screenshot shows a window titled "DR CPU Configuration" with a dropdown arrow in the top-left corner. Below the title is a header "CPU Configuration For Board 0". Underneath, there are two sub-headers: "Processor Set" and "Board Threads". A table follows with columns: CPU, STATUS, ID, COUNT, USER, SYS, and PROCS. The table contains four rows of data. Each row has a vertical scroll bar on its right side. At the bottom of the window is a "dismiss" button.

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

FIGURE 3-8 DR CPU Configuration Window

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

**TABLE 3-2** DR CPU Configuration Information

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

## ▼ To View DR Memory Configuration Information

- **Click the `memory` button.**

The DR Memory Configuration window is displayed (FIGURE 3-9).

▽
DR Memory Configuration

System Memory Sizes (MB)

Current System:	2048
Attached Capacity:	18432
dr-max-mem:	20480
Memory Detach:	enabled

Memory Configuration for Board 0

Memory Size(MB):	1024
Interleave Board:	no interleave
Physical Pages:	9437184-9568255

Board contains all pageable memory.

Memory Drain for Board 0	ESTIMATED
--------------------------	-----------

Reduction:	1024
Remaining in System:	1024
Percent Complete:	
Drain Start Time:	
Current Time:	

dismiss

**FIGURE 3-9** DR Memory Configuration Window

The DR Memory Configuration window is divided into three panels:

**TABLE 3-3** DR Memory Configuration Information

---

**System Memory Sizes (Domain Memory Information)**

Current Domain	Total size of memory in the domain from all boards
Attach Capacity	Amount of memory that can currently be added by using the DR Attach operation
dr-max-mem	Current value of the OBP variable dr-max-mem. For more information, see “Memory: dr-max-mem” on page 3.

**Memory Configuration for Board 0 (Board-Level Information)**

Memory Size (MB)	Amount of memory on the selected board
Interleave Board	Board that the selected board is interleaved with
Physical Pages	Highest and lowest physical pages that are occupied by this board's memory (Small memory areas in the middle of this range may not be used by this board. Note that DR is not able to detach boards that have interleaved memory.)

**Status/State**

Unavailable	A suitable target memory area is not currently available.
Estimated	The estimated values are displayed prior to starting the drain operation. The values displayed reflect the memory configuration that would result if the drain operation were started at this point. Note that the estimated values may differ from the in-progress values depending on the domain memory usage at the time drain was started.
In Progress	The drain operation is in progress.
Complete	The drain operation is in finished.

**Memory Drain Information**

Reduction	The amount of memory to be removed from domain usage when the board is detached.
Remaining in Domain	The domain memory size after the board is detached.

---

**TABLE 3-3** DR Memory Configuration Information

---

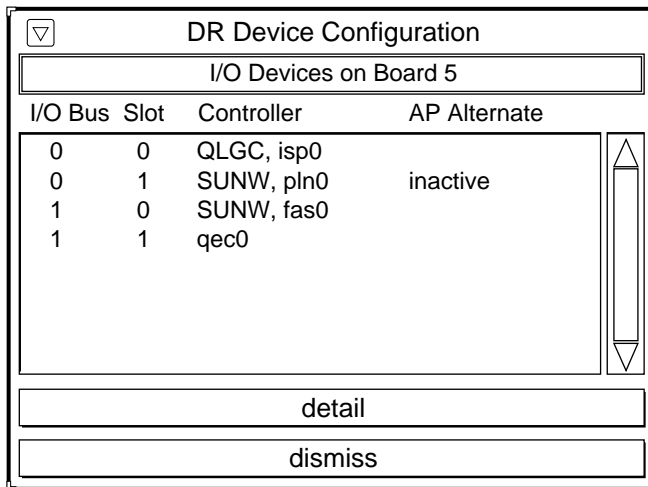
Percent Complete	How far the drain operation has progressed. Note that the time required to drain each memory page is not constant. Some memory pages take longer to drain than others.
Drain Start Time	The time the drain operation was started.
Current Time	The current time, which can be compared to the drain start time to see how long the drain operation has been in progress.

---

## ▼ To View DR Device Configuration Information

- **Click the device button.**

The DR Device Configuration window is displayed (FIGURE 3-10).



**FIGURE 3-10** DR Device Configuration Window

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

---

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

---

## ▼ To View DR Device Detailed Information

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

The following window is displayed for each selected controller (FIGURE 3-11).

Device	Opens	Name	Usage
ssd80	0	/dev/dsk/c0t5d0s0	
ssd81	0	/dev/dsk/c0t5d1s0	
ssd82	0	/dev/dsk/c0t5d2s0	
ssd83	31	/dev/dsk/c0t5d3s0	/
ssd83	0	/dev/dsk/c0t5d3s1	swap, /tmp
ssd83	10	/dev/dsk/c0t5d3s3	/var
ssd83	1	/dev/dsk/c0t5d3s5	/opt
ssd83	25	/dev/dsk/c0t5d3s6	/usr
ssd83	0	/dev/dsk/c0t5d3s7	/export
ssd84	0	/dev/dsk/c0t5d4s0	

FIGURE 3-11 DR Detail Device Window

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

---

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

---

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

## ▼ To View DR OBP Configuration Information

---

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

---

- **Click the `obp` button.**

The DR OBP Configuration window is displayed (FIGURE 3-12).

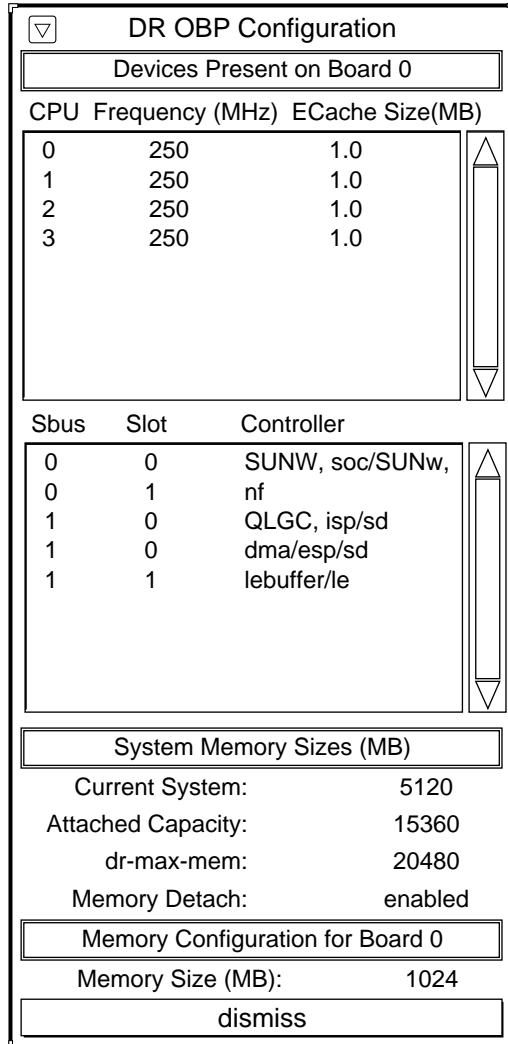


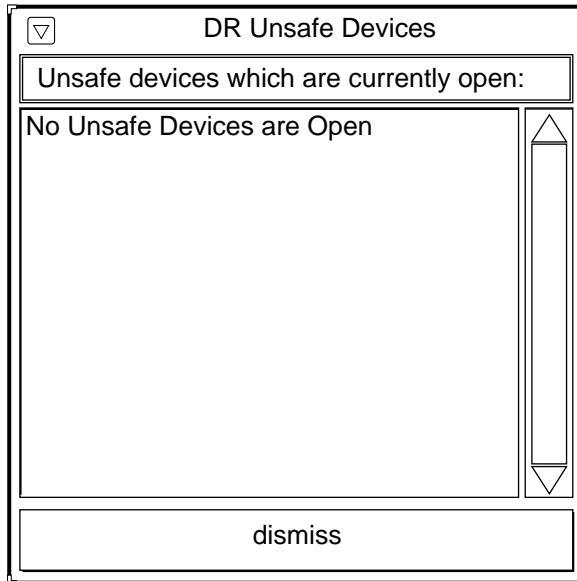
FIGURE 3-12 DR OBP Configuration Window

## ▼ To View the Suspend-Unsafe Devices Across the Entire Domain

- **Click the unsafe button.**

The DR Unsafe Devices window is displayed (FIGURE 3-13).





**FIGURE 3-13** DR Unsafe Devices Window

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



# Index

---

## SYMBOLS

/etc/system file, 18

## A

abort button, 23, 35  
active DR operations, only one, 2  
amount of memory attachable, 46  
attach, 1  
attach buttons, 23  
attach, complete, 25  
attach, dynamic reconfiguration window, 25  
attach, parameter selection, 24  
attachable memory, 46  
attaching with dr(1M), 27

## B

board attach, 1  
board detach, 1, 30  
board, attach, 24  
buttons  
  abort, 23, 35  
  complete, 23, 25, 34, 38  
  CPU, 43  
  device, 47  
  dismiss, 23, 35  
  force, 34  
  help, 23, 35  
  init attach, 23, 25

reconfig, 23, 34

select, 24

## C

complete attach, 22, 25  
complete attach vi dr(1M), 29  
complete button, 23, 25, 34, 38  
complete detach, 31, 38  
complete detach via dr(1M), 41  
CPU button, 43  
CPU configuration window, 43

## D

DDI/DKI, 19  
detach, 1, 30  
detach and network devices, 31  
detach and non-network devices, 32  
detach and processors, 33  
detach buttons, 34  
detach, parameter selection window, 36  
detaching with dr(1M), 38  
detaching with Hostview, 35  
detach-safe list, 19  
device button, 47  
device configuration window, 47  
device detail window, 48  
dismiss button, 23, 35

- DR attach, 1
- DR detach, 1, 30
- DR overview, 1
- DR parameter selection, 24
- DR unsafe devices, 51
- dr(1M), attaching via, 27
- dr\_safe list, 17
- drain, 30, 34
- drain via dr(1M), 40
- dr-max-mem environment variable, 46
- drshow, dr(1M) command, 29
- DR-unsafe device, 14
- Dynamic Reconfiguration Window for drain Operation, 37
- dynamic reconfiguration window, attach, 25

## **E**

- enabling detach, 6
- environment variables
  - dr-max-mem, 46

## **F**

- force button, 34

## **H**

- help button, 23, 35
- Hostview, detaching via, 35

## **I**

- I/O devices, configuring for detach, 6
- init attach, 21, 25
- init attach button, 23, 25
- init attach, with dr(1M), 28

## **M**

- memory attach capacity, 46

- memory button
  - buttons
    - memory, 44
- memory configuration window, 44
- memory draining, detach, 30
- memory remaining in system, 46
- memory, configuring for detach, 7
- memory, total size (all boards), 46

## **N**

- network devices and detach, 31
- non-network devices and detach, 32

## **O**

- OBP window, 50
- overview of DR, 1

## **P**

- parameter selection window, detach, 36
- parameter selection, attach, 24
- processors and detach, 33

## **R**

- reconfig button, 23, 34
- remaining memory, detach, 46

## **S**

- select button, 24
- source domain, attach, 24
- SSP messages file, 1
- suspend-bypass devices, 18
- swap space, configuring for detach, 9
- system information, viewing, 41

## **T**

target domain, attach, 24

## **U**

unsafe device, 14

unsafe devices, 51

unsafe devices window, 51

## **V**

viewing system information, 41

## **W**

windows

- CPU configuration, 43

- detach parameter selection, 36

- device configuration, 47

- device detail, 48

- DR parameter selection, 24

- dynamic reconfiguration, 25

- memory configuration, 44

- OBP, 50

- unsafe devices, 51

