

Dynamic Reconfiguration User's Guide

Sun Microsystems Computer Company
A Sun Microsystems, Inc. Business
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part No: 805-0080-10
Revision A, March 1997



THE NETWORK IS THE COMPUTER™

Copyright 1997 Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, California 94043-1100 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.

Portions of this product may be derived from the UNIX[®] system, licensed from Novell, Inc., and from the Berkeley 4.3 BSD system, licensed from the University of California. UNIX is a registered trademark in the United States and other countries and is exclusively licensed by X/Open Company Ltd. Third-party software, including font technology in this product, is protected by copyright and licensed from Sun's suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

Sun, Sun Microsystems, the Sun logo, Solaris, SunSoft, SunPics, the SunSoft logo, SunOS, OpenWindows, DeskSet, ONC, ONC+, Power Management, TurboGX, TurboGX Plus, S24, SunFastEthernet, OpenBoot, IPX, Power Management, AnswerBook, Jump Start, XGL, SLC, ELC, IPC, Sun-3, SunCD, SunCD Plus, SunCD 2Plus, SunVideo, SunDials, SunButtons, NeWSprinter, NeWSprinter CL+, NeWSprint, SunDiag, Ultra, UltraServer, UltraSPARC, Ultra Enterprise, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK[®] and Sun[™] Graphical User Interfaces were 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.

X Window System is a trademark of X Consortium, Inc.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.



Copyright 1997 Sun Microsystems Inc., 2550 Garcia Avenue, Mountain View, Californie 94043-1100, U.S.A. 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 de sa documentation associée 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.

Des parties de ce produit pourront être dérivées du système UNIX[®] licencié par Novell, Inc. et du système Berkeley 4.3 BSD licencié par l'Université de Californie. UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd. 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.

Sun, Sun Microsystems, le logo Sun, Solaris, SunSoft, le logo SunSoft, SunOS, OpenWindows, DeskSet, ONC, ONC+, Power Management, TurboGX, TurboGX Plus, S24, SunFastEthernet, OpenBoot, IPX, Power Management, AnswerBook, Jump Start, XGL, SLC, ELC, IPC, Sun-3, SunCD, SunCD Plus, SunCD 2Plus, SunVideo, SunDials, SunButtons, NeWSprinter, NeWSprinter CL+, NeWSprint, SunDiag, Ultra, UltraServer, UltraSPARC, Ultra Enterprise, et NFS sont des marques déposées ou enregistrées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC, utilisées sous licence, sont des marques déposées ou enregistrées 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.

Les interfaces d'utilisation graphique OPEN LOOK[®] et Sun[™] ont été développées 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 aussi 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.

Le système X Window est un produit du X Consortium, Inc.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, NI EXPRESSE NI IMPLICITE, Y COMPRIS, ET SANS QUE CETTE LISTE NE SOIT LIMITATIVE, DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DES PRODUITS A RÉPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ILS NE SOIENT PAS CONTREFAISANTS DE PRODUITS DE TIERS.

Table of Contents

1. Introduction to DR	1-1
2. DR Configuration Issues	2-1
Configuring the System for DR	2-1
Configuring for DR Attach	2-1
I/O Devices	2-1
Memory: dr-max-mem	2-2
Processors	2-4
Configuring for DR Detach	2-4
Enabling DR Detach	2-4
I/O Devices	2-4
Memory	2-5
Target Memory Constraints	2-6
Swap Space	2-6
Reconfiguration After a DR Operation	2-7
I/O Devices	2-7

Disk Devices	2-7
When to Reconfigure	2-8
DR and AP Interaction	2-8
RPC Timeout or Loss of Connection	2-9
Operating System Quiesce	2-10
Suspend-Safe / Suspend-Unsafe Devices	2-11
Special Quiesce Handling for Tape Devices	2-12
DR and DDI	2-12
DR and DDI_DETACH	2-12
▼ To Detach a Detach-Unsafe Driver that is Loaded	2-13
DR and DDI_SUSPEND/DDI_RESUME	2-13
Adding Suspend-Safe Devices	2-14
Adding Suspend-Bypass Devices	2-15
DR Detach-Safe Devices	2-16
Adding Detach-Safe Devices	2-17
3. Using Dynamic Reconfiguration	3-1
Attaching a System Board	3-1
Init Attach	3-2
Complete Attach	3-2
Attach Buttons	3-3
▼ To Attach a Board with Hostview	3-4
▼ To Attach a Board with dr(1M)	3-8
Detaching a System Board	3-12
Drain	3-12

Complete Detach	3-13
Network Devices	3-13
Non-Network Devices	3-14
Processors	3-15
Finishing the Complete Detach Operation	3-15
Hostview Detach Buttons	3-17
▼ To Detach a Board with Hostview	3-18
▼ To Detach a Board with dr(1M)	3-21
Viewing System Information	3-23

Figures

Figure 3-1	Attach—Board and Domain Selection Window	3-4
Figure 3-2	Dynamic Reconfiguration Window — Init Attach	3-5
Figure 3-3	Dynamic Reconfiguration Window—Complete Attach	3-7
Figure 3-4	Detach—Board and Domain Selection Window	3-18
Figure 3-5	Dynamic Reconfiguration Window — Drain	3-19
Figure 3-6	Hostview — System Information Buttons	3-24
Figure 3-7	Hostview Properties Window—DR Properties	3-24
Figure 3-8	Hostview—CPU Configuration Window	3-25
Figure 3-9	Hostview—Memory Configuration Window	3-26
Figure 3-10	Hostview—Device Configuration Window	3-28
Figure 3-11	Hostview—Device Detail Window	3-29
Figure 3-12	Hostview—OBP Window	3-30
Figure 3-13	Hostview—Unsafe Devices Window	3-31

Preface

This document describes the Dynamic Reconfiguration (DR) feature, which enables you to logically attach and detach system boards from the Ultra™ Enterprise™ 10000 server while the corresponding domain continues running.

Who Should Use This Book

This manual is intended for the Enterprise 10000 system administrator who should have a working knowledge of UNIX® systems, particularly those based on the Solaris operating environment. If you do not have such knowledge, refer to the Solaris™ User and System Administrator AnswerBooks provided with this system.

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.

Related Documentation

DR is normally started from the Hostview GUI in the SSP environment. See also:

- *Dynamic Reconfiguration Reference Manual*, part number 805-0081-10
- *SMCC Open Issues Supplement, Solaris 2.5.1 Hardware:4/97*, a printed document in your media box, part number 802-7683-10. This document may contain Dynamic Reconfiguration Release Notes in the section, “Ultra Enterprise 10000 Servers”
- *Ultra Enterprise 10000 SSP 3.0 User’s Guide*, part number 805-0076-10
- *Ultra Enterprise 10000 SSP Reference*, part number 805-0077-10

Ordering Sun Documents

The SunDocsSM program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals using this program.

For a list of documents and how to order them, see the catalog section of the SunExpressTM Internet site at <http://www.sun.com/sunexpress>.

What Typographic Changes Mean

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

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
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. machine_name% You have mail.
AaBbCc123	What you type, contrasted with on-screen computer output	machine_name% su Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be root to do this.

Shell Prompts in Command Examples

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 prompt	machine_name%
C shell superuser prompt	machine_name#
Bourne shell and Korn shell prompt	\$
Bourne shell and Korn shell superuser prompt	#

Introduction to DR



Dynamic Reconfiguration (DR) enables you to logically attach and detach system boards to and from the operating system without incurring 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 system board upgrade
- Modify the domain configuration on the Ultra Enterprise 10000 system
- Remove a faulty system board

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 procedures through the Hostview GUI program (see `hostview(1M)`) and 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 the system and are available to the system. The system board must already be present in the system, powered on, and not 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 system 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 via Hot Swap by your service provider.

While DR operations are being performed within a domain, messages regarding status or exceptions are written to the system message buffer (`/var/adm/messages`) and the SSP message files (`$(SSPOPT)/adm/host/messages` and `$(SSPOPT)/adm/messages`) by `dr_daemon(1M)` and the operating system. These messages, in addition to those displayed by Hostview and the `dr(1M)` shell application, 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



Configuring the System for DR

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 256MB of memory.

Configuring for DR Attach

This section describes how to configure DR before performing an attach operation.

I/O Devices



Warning – Be careful when choosing the slot into which a board is inserted to prevent disk controller re-numbering. For more information see “Reconfiguration After a DR Operation” on page 2-7

Memory: `dr-max-mem`

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. However, 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 calculate the optimum value for `dr-max-mem`, combine the amount of memory most likely to be added during all DR Attaches and the current amount of memory present in the domain. Then, set `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 system limits the maximum value of `dr-max-mem` at boot time, as follows:

Table 2-1 Maximum Values of `dr-max-mem`

Current Physical Memory	<code>dr-max-mem</code> Maximum Value
256 MB	0
512 MB	16 GB
1 GB	32 GB
2 GB	32 GB

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 `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.

To set the `dr-max-mem` environment variable, bring up the OBP prompt for the domain and enter:

```
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.

If the `dr-max-mem` variable is non-zero, the following messages are displayed at boot time in the system `msgbuf` (`/var/adm/messages`):

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

where *XXX* represents the amount of physical memory available to the operating system and is effectively the same as the operating system variable, `physinstalled`; and where *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
```

where *ZZZ* represents the updated amount of memory that can still be attached.

Processors

Prior to attaching a board, diagnostics are run on the board. This requires the presence of at least one processor on the board. The processor must not be blacklisted.

Configuring for DR Detach

This section describes how to configure DR before performing an attach operation.

Enabling DR Detach

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 board feature relies on the Alternate Pathing feature or Solstice DiskSuite mirroring, or a mixture of the two, when used to detach a board that hosts I/O controllers attached to vital system resources. 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 or the disk is mirrored. The alternate path and/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 system 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 since 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 system to accommodate currently running programs.

A board hosting non-vital system resources can be detached whether or not there are alternate paths to the resources. There is a domain disruption penalty associated with the detach operation since 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 unmounting them.

All I/O device drivers 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

Note – The information in this section applies only to systems in which memory is *not* interleaved between boards.

If you use memory interleaving between system boards, those system boards cannot be detached. This is because DR does not yet support inter-board 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.

Before a board can be detached, the memory on that board must be vacated by the operating system. Vacating a board means flushing its pageable memory to swap space and copying its *permanent memory*—that is, non-pageable kernel and OBP memory—to another memory board. When permanent memory is on the detaching board, the operating system must find other memory to receive the copy.

You can use the `dr(1M)` command `drshow(1M)`, as follows, to determine whether a board's memory is pageable:

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

Alternatively, you can determine whether 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 System Information” on page 3-23.

Target Memory Constraints

When permanent memory is detached, DR chooses a target memory area to receive a copy of the memory. The target memory area must adhere to the following rules. The DR software automatically checks for total adherence, and disallows the DR memory operation if it cannot verify that. The rules are shown here to illustrate the possible reasons why a DR memory operation might be disallowed.

- It must be large enough to hold a copy of the non-pageable memory.
- It must not be interleaved with memory on other boards.

If no target board is found, the detach operation is refused and DR displays the following messages on the system console:

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

Swap Space

The system swap configuration consists of the swap devices and `swapfs` (memory). The system must contain enough swap space so it can flush pageable memory. For example, if you want to remove 1 GB of memory from a 2 GB system, you will need up to 16 GB 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 of the detach operation does not complete, and you must abort the operation.

Reconfiguration After a DR Operation

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

I/O Devices

The DR user interface enables you reconfigure the system 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 the reconfiguration sequence is executed after a board is attached, device path names not previously seen by the system are entered into 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.

Disk Devices

Disk controllers are numbered consecutively as they are encountered by the `disks(1M)` program. All disk partitions are assigned `/dev` names according to the disk controller number that `disks(1M)` assigns. For example, all disk partitions accessible via disk controller 1 are named `/dev/dsk/c1tLdMsN`, where `c1` is the disk controller number and, in most cases, `L` corresponds to the disk target number, `M` to the logical unit number, and `N` to the partition number. When the reconfiguration sequence is executed after a board is detached, the `/dev` links for all 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 there are 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; board 1 is still out.

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 uses the `/dev` link names. Update `/etc/vfstab` files and execute other administrative actions necessary due to the changes in the `/dev` link names.

When to Reconfigure

You should reconfigure the system under several conditions, as described below.

- **Board Addition** – When adding a board to a domain, you must execute the reconfiguration sequence to configure the I/O devices 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 for disk devices.
- **Board Replacement** – If you remove a board then reinsert it in a different slot, or replace a board with another board that has different I/O devices, you must execute the reconfiguration sequence to configure the I/O devices 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 into the same slot that was vacated to retain the original `/dev` link names.

DR and AP Interaction

DR notifies the Alternate Pathing (AP) subsystem when system boards are attached, detached, or placed in the detach 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 system messages buffer and DR operations continue without error. To disable this interaction, use the `-a` option when invoking `dr_daemon(1M)`. See the `dr_daemon(1M)` man page in the *Solaris Reference Manual for SMCC-Specific Software*.

For more information, also see the *Alternate Pathing 2.0 User's Guide*.

RPC Timeout or Loss of Connection

`dr_daemon(1M)` (which runs in each domain) communicates with Hostview and the `dr(1M)` shell application (both of which run on the SSP) via remote procedure calls (RPC). If an RPC timeout 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
```

where `dr_daemon_pid` is the process ID of the DR daemon, and `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 `/usr/platform/sun4ul/sbin/dr_daemon`.

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

Operating System Quiesce

During a DR detach operation on a system board with non-pageable OBP or kernel memory, the operating system is briefly quiesced; that is, all operating system and device activity on the domain centerplane must cease for a few seconds during a critical phase of the operation. The quiesce 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:

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

The conditions that cause processes to fail to suspend are generally temporary in nature. You can retry the operation until the quiesce succeeds.

A quiesce 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 quiesce even if forcible conditions are still present.

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

If a real-time process is running, determine whether suspending the process would have an adverse effect on the functions performed by the process. If not, you can force the operating system to quiesce. (To force a quiesce, you can either choose the `Force` button within `Hostview` as described in “To Detach a Board with `Hostview`” on page 3-18, 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 in the domain 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 2-11.

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) require your approval (e.g. a real-time process is running) or intervention (e.g. a suspend-unsafe device 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 (e.g., 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.

The drivers released by Sun Microsystems that are known to be suspend-safe are `sd`, `isp`, `esp`, `fas`, `hme` (Sun FastEthernet™), `nf` (NPI-FDDI), `qe` (Quad Ethernet), `le` (Lance Ethernet), and the SSA drivers (`soc`, `pln`, and `ssd`). See `sd(7D)`, `isp(7D)`, `esp(7D)`, and `ssd(7D)` in the *SunOS Reference Manual*. The known suspend-unsafe drivers are the tape-related drivers (`sga` and `st`). These lists will change over time. To add a driver to either list, see “Adding Suspend-Safe Devices” on page 2-14.

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 and reconnect them after the operating system resumes. Doing this prevents traffic from arriving at the device and, thus, the device has no reason to access the domain centerplane. 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.



Warning – If you attempt to do a forced quiesce operation when suspend-unsafe devices are present, the domain may hang if any activity occurs on any suspend-unsafe device during the period of system quiescence. This will not affect other running domains on the Ultra Enterprise 10000 system.

Special Quiesce Handling for Tape Devices

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 executing a DR operation that quiesces the operating system, make sure all tape devices are closed or not in use.

DR and DDI

Not all drivers support the Ultra 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 2-16

▼ To Detach a Detach-Unsafe Driver that is Loaded

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

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

- 2. Use standard Solaris interfaces to manually close and modunload all such drivers on the board.**

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

- 3. Detach the system board in the normal fashion.**

If you cannot execute the above steps, you may reboot your domain with the board blacklisted (see `blacklist(4)`), and the board can later be removed.

Note – Many third-party drivers (those purchased from vendors other than Sun Microsystems) do not properly 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 strongly 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 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 2-14. Note that a quiesce is required only when detaching a board that contains non-pageable 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)`.

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 to use DR, Sun Microsystems strongly 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 system to quiesce. Doing so forces the 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, that can be safely quiesced. The Note in “Suspend-Safe / Suspend-Unsafe Devices” on page 2-11 lists the known suspend-safe and suspend-unsafe devices at the time this document was printed.

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 as described below. This file enables you to append the list already maintained in the 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 quiesce is requested, the device is suspended and resumed prior to the quiesce. Tape devices are *not* suspend-safe; do *not* append such devices to the suspend-safe list via the `/etc/system` file.

Note – Previously, the suspend-safe list was called the `dr_safe` list. You 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)
```

To add new devices that support quiescing to the `/etc/system` file, use 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.

Adding Suspend-Bypass Devices

The Ultra Enterprise 10000 system has a preset list of devices that it ignores during the quiesce 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 Ultra 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 2-13. 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 enables you to append 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 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`.

Adding Detach-Safe Devices

Add new devices that support DR Detach to the `/etc/system` file in 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 3-4.

You can attach system boards that are present in the machine, powered on, and not part of an active domain (i.e., 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. See “Configuring the System for DR” on page 2-1 for a discussion of which board configurations can be attached.

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

Once you’ve selected an eligible board and a target domain, the DR Attach operation proceeds through two steps: 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:

- Adds this 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 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 this 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 board configuration is presented to the operating system. The devices are added to the operating system configuration but are not attached.

After the Init Attach phase is completed, the OBP board configuration may be displayed to confirm which devices are present on the board. You may then enter the Complete Detach phase to proceed or you may 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 via 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. 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 system configuration for the board.

After a board is successfully attached, you have the option of reconfiguring the I/O devices as described in “Reconfiguration After a DR Operation” on page 2-7. This operation may 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 3-2). Once the operation has completed successfully, the label on this button changes to `complete`.
- `complete`—Completes the attach operation (see “Complete Attach” on page 3-2).
- `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 2-7).
- `abort`—Cancels the attach operation. This button is enabled after the `Init Attach` operation has successfully completed (see “Init Attach” on page 3-2).
- `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. `dismiss` terminates any work being done on the SSP for the attach operation. For example, if `hpost(1M)` is running when you select `dismiss`, that `hpost(1M)` process is terminated. Note that `dismiss` does not terminate work being done on the host via RPC calls to the `dr_daemon(1M)`. Once an RPC call is initiated, the host completes the RPC call regardless of whether the calling program is waiting for the RPC call 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. So you can dismiss the window, then return to the DR operation later and complete or abort the attach.

- help—Access online information regarding DR Attach operations.

▼ To Attach a Board with Hostview

Note – Before performing the following steps, be sure to read “Attaching a System Board” on page 3-1”.

1. **From Hostview, choose Configuration ► Board ► Attach. The following window is displayed.**

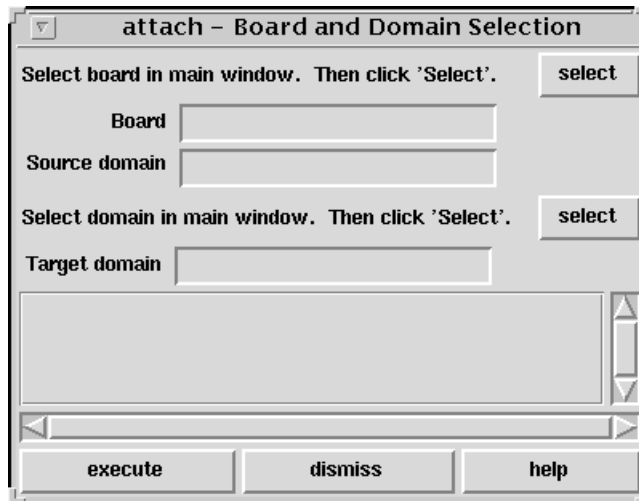


Figure 3-1 Attach—Board and Domain Selection Window

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 in the window shown above.**
The Board and Source Domain fields are automatically filled in for you. (You can also manually edit those fields.) The source domain is the domain that the board currently belongs to. If the board is not a member of any domain, the source domain entry is no_domain.

- In the main Hostview window, select the domain to which you want to attach the board.**
To do this, select any board that is currently a member of that domain.
- Click the bottom Select button in the window shown above.**
The Target Domain field is automatically filled in for you. (You can also manually edit that field.)
- Click Execute in the window shown above.**
If any errors occur, the error messages appear in the window shown above. Otherwise, the following window is displayed.

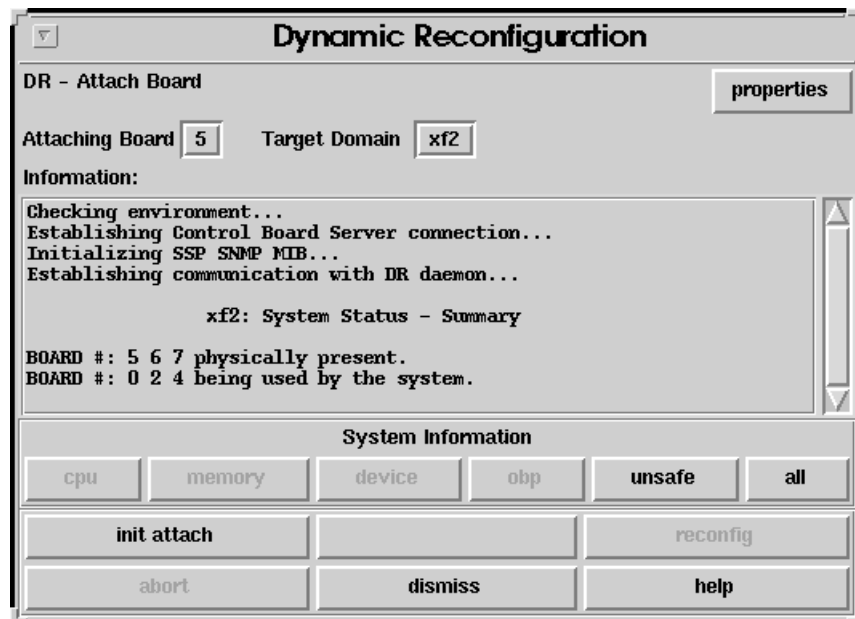


Figure 3-2 Dynamic Reconfiguration Window — Init Attach

7. Click the `Init Attach` button.

Clicking on the `Init Attach` button begins the first phase of the board attach process. 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 system information to verify that you want to proceed, as described in “Viewing System Information” on page 3-23.

The `Init Attach` operation may take a few minutes to complete. Output from the `hpost(1M)` commands is directed to the Information pane of the window, as shown above in Figure 3-2.

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 on page 3-7, with the `complete` button enabled.

8. **Select** complete.

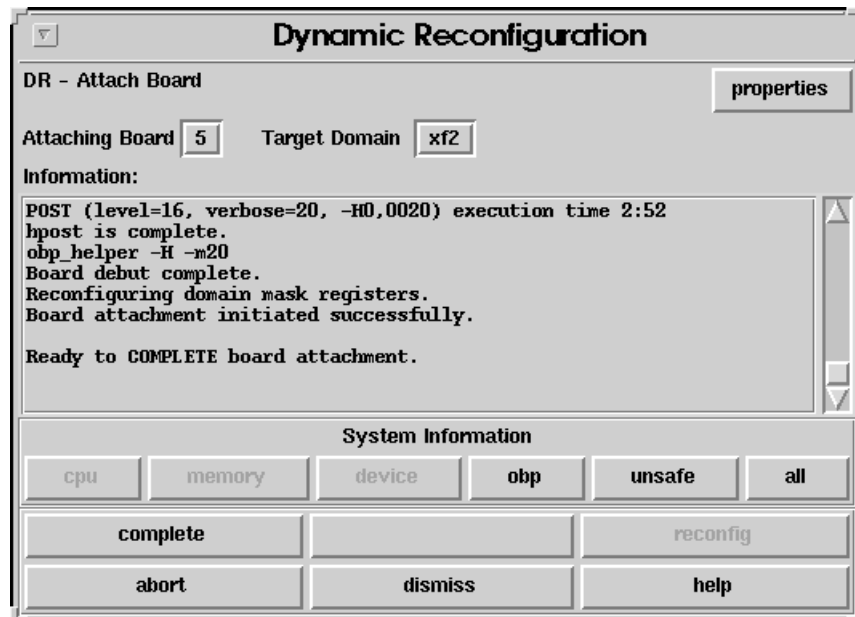


Figure 3-3 Dynamic Reconfiguration Window—Complete Attach

This operation may take a minute or so to complete. When it has successfully completed, DR displays the following message.

```
Board attachment completed successfully
```

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

You may wish to view system information about the newly attached board using the buttons (CPU, Memory, Device, and so forth), as described in “Viewing System Information” on page 3-23.

Caution – Before choosing the `reconfig` option, be sure to read “Reconfiguration After a DR Operation” on page 2-7.

9. **Select** dismiss.

The DR Attach operation is complete.

▼ To Attach a Board with `dr(1M)`

Before performing the following steps, see “Attaching a System Board” on page 3-1. 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, “Introduction to DR”. A quick reference guide is available in the `dr(1M)` application via the `help` command.

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

```
% dr
dr>
```

- 2. Use `drinit(1M)` to establish communication with the `dr_daemon(1M)` on the desired target domain.**

```
dr> drinit xf3
Checking environment...
Establishing Control Board Server connection...
Initializing SSP SNMP MIB...
Establishing communication with DR daemon...

          xf3: System Status - Summary

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

3. Begin the Init Attach operation for the designated board.

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

```
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 init_reset: Initial system resets...
phase jtag_integ: JTAG probe and integrity test...
phase mem_probe: Memory dimm probe...
phase iom_probe: I/O module type probe...
phase jtag_bbsram: JTAG basic test of bootbus sram...
phase procl: Initial processor module tests...
phase pc/cic_reg: PC and CIC register tests...
phase dtag: CIC DTAG tests...
phase mem: MC register and memory tests...
phase io: I/O controller tests...
phase procmem2: Processor vs. memory II tests...
phase lbexit: Centerplane connection tests...
phase npb_mem: Non-Proc Board MC and memory tests...
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 successfully completing the Init Attach operation, you can use `drshow(1M)` to see an inventory of the board resources:

```
dr> drshow board_number OBP
```

If you wish to abort the attach operation, execute `abort_attach(1M)` as follows:

```
dr> abort_attach board_number
```

- Alternatively, to complete the board attach operation, execute `complete_attach(1M)` as follows:

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

5. Execute `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 3-18.

System boards that are currently being used by the operating system can be detached if they meet the requirements covered in “Configuring for DR Detach” on page 2-4. 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 non-pageable 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 “Configuring for DR Detach” on page 2-4.

Once the Drain operation is started, the board’s pageable memory is flushed to disk, which removes it from use by the system. 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 resources. However, less memory is available to the system.

Note – When memory is drained, there must be enough memory and swap space remaining in the system 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 system 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.

The Drain operation is complete when all memory pages are drained. You can then Complete the operation.

Complete Detach

Before the detach operation can be completed, you must terminate all usage of board resources (processors, memory, and I/O devices). DR terminates the use of memory, processors, and network devices automatically, and you must terminate the use of all non-network I/O devices. This subject is discussed in “Configuring the System for DR” on page 2-1.

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

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 performing the detach operation, and 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 the above 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 operation fails if 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.
- 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. If the SSP is renamed, `/etc/ssphostname` must be manually updated.)
- The active alternate for an Alternate Pathing (AP) meta device when the AP meta device is plumbed. Interfaces used by the AP system should not be the active path when the board is being detached. Manually switch the active path to one 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 the following tasks for non-network devices.

- 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.
- Unmount file systems, including Solstice DiskSuite meta-devices that have a board resident partition. Example: `umount /partit`

- Remove Solstice DiskSuite or Alternate Pathing databases from board-resident partitions. The location of Solstice DiskSuite or Alternate Pathing databases is explicitly chosen by the user and can be changed.
- 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.
- Any RSM 2000 controllers on the board that is being detached should be taken offline, using the `rm6` or `rdacutil` commands.
- Remove disk partitions from the swap configuration.
- Either kill any process that directly opens a device or raw partition, or direct it to close the open device on the board.
- 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 effect NFS client systems.

Processors

The boot processor is responsible for servicing the tick-timer interrupts and for broadcasting the heartbeat request to the other processors in the domain. 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.

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 resolving the problem, you can perform the complete detach operation again.

The complete detach operation may also fail due to quiesce problems, which are described in “Operating System Quiesce” on page 2-10. After resolving the quiesce 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 (i.e., file systems were unmounted and networks were unplumbed), it is your responsibility to 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 updated in the `domain_config` file.

You can now attach the board to another domain, remove it via Hot Swap, leave it in the system unattached, or reattach it at a later time.

Hostview Detach Buttons

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

- `drain` Starts to drain memory (see “Drain” on page 3-12). Once the Drain operation is finished, the `drain` button becomes the `complete` button.
- `complete` Completes the detach operation (see “Complete Detach” on page 3-13).
- `force` If the `complete` detach operation fails due to a forcible quiesce condition, the `force` button is enabled, permitting you to complete the detach operation by forcibly quiescing the domain (see “Operating System Quiesce” on page 2-10).
- `reconfig` Automatically reconfigures device directories in a domain. You may want to run `reconfig` after permanently detaching a board. Use `reconfig` with extreme caution (see “Reconfiguration After a DR Operation” on page 2-7).
- `abort` Cancels the DR operation. This button is enabled after the drain operation starts and remains enabled until the complete detach operation starts. To stop the draining of memory and cancel the detach, choose `abort` (see “Detaching a System Board” on page 3-12).
- `dismiss` Cancels any step that is in progress, and leaves the board in its current state (In Use, drain, Present). At any point during the DR Detach operation you can remove the DR Detach window by choosing `dismiss` which terminates any work being done on the SSP for the detach operation. Note that `dismiss` does not terminate work being done on the host via RPC calls to the `dr_daemon(1M)`. Once an RPC call is initiated, the host completes the RPC call regardless of whether Hostview is waiting for the RPC call to finish.

The host `dr_daemon(1M)` keeps track of the progress of the detach operation. Once the `drain` is started, it remembers this state. So you can dismiss the window and then return to the DR operation later and either `complete` or `abort` the detach operation.

help Accesses online information regarding DR detach operations.

▼ To Detach a Board with Hostview

Note – Before executing the following steps, see “Detaching a System Board” on page 3-12.

1. **From the Hostview menu, choose** Configuration ► Board ► Detach. The following window is displayed.

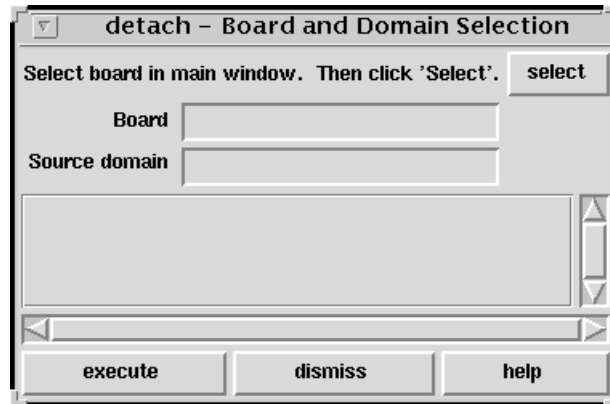


Figure 3-4 Detach—Board and Domain Selection Window

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 in the window shown above.**
The Board and Source domain fields are automatically filled in for you. (You can also manually edit those fields if you wish.)

4. Click **Execute** in the window shown above.

If the target domain is not currently booted, the attach operation simply manipulates the domain configuration file on the SSP. However, if the domain is currently running, the following window is displayed.

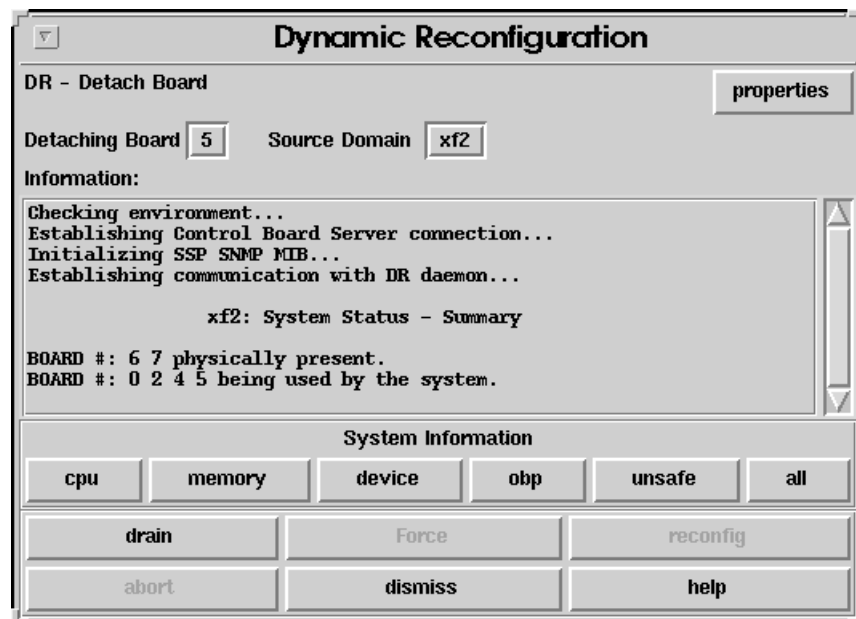


Figure 3-5 Dynamic Reconfiguration Window — Drain

5. Click the **drain** button.

Hostview begins draining memory. The memory display automatically appears and allows 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 System Information Displays` option in the DR Properties window, as described in “Viewing System Information” on page 3-23.

If the Drain operation fails, an explanatory message is displayed in the Information pane of the window, above. Once you’ve determined the cause, and corrected it, you may wish to 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.

This brings up the DR Device Configuration window which 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 3-13.

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

8. Choose `complete`.

This operation may take several minutes to complete. 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 system are on the board being detached.
- Critical network interfaces are on the board being detached. You must stop all usage of these networks manually (see "Complete Detach" on page 3-13).
- All usage of the I/O devices has not been stopped. The Information pane identifies the device on which the error was encountered (see "Complete Detach" on page 3-13).
- The Quiesce failed. You must determine and resolve the cause of the error (see "Operating System Quiesce" on page 2-10).

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 choosing the `reconfig` option, be sure to read "Reconfiguration After a DR Operation" on page 2-7.

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

▼ To Detach a Board with `dr(1M)`

Before executing the following steps, see “Detaching a System Board” on page 3-12. 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, “Introduction to DR”.

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

```
% dr
dr>
```

2. **Execute `drinit(1M)` to establish communication with the `dr_daemon(1M)` on the desired domain.**

```
dr> drinit xf3
Checking environment...
Establishing Control Board Server connection...
Initializing SSP SNMP MIB...
Establishing communication with DR daemon...

          xf3: System Status - Summary

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

3. Drain the board.

```

dr> drain 6
Removing board 6 from domain_config file.
Start draining board 6
Board drain started. Retrieving System 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 System= 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>

```

This initiates the drain operation, and returns immediately. You can monitor the progress of the drain operation with the following command.

```
dr> drshow board_number drain
```

In addition, you can initiate the drain with the `wait` option of the

`drain(1M)` command, which does not return 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, execute `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 quiesce due to forcible conditions.”, and you have determined the root cause of the quiesce 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 quiesce 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 System Information

Both `dr(1M)` and Hostview allow 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 via 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, and it is not guaranteed to be as detailed as the other three displays. This section shows how to use the displays.

When you are using Hostview to do a DR operation, you can view system information by using the following buttons:



Figure 3-6 Hostview — System Information Buttons

When you click any of the buttons shown above, a window is displayed. The window remains on the screen until you click the `dismiss` button within that window. If you click the `All` button, all of the currently enabled windows are displayed.

By default, the `cpu`, `memory`, `device`, and `unsafe` windows are updated periodically as long as they are displayed. You can set two properties that determine how the displays are updated. To do this, click the `Properties` button in the Dynamic Reconfiguration window (for attach or detach). The properties window is shown in Figure 3-7.

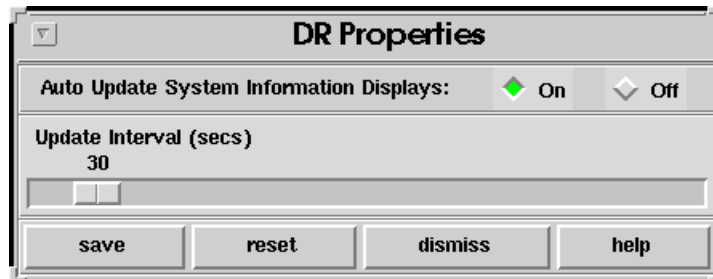
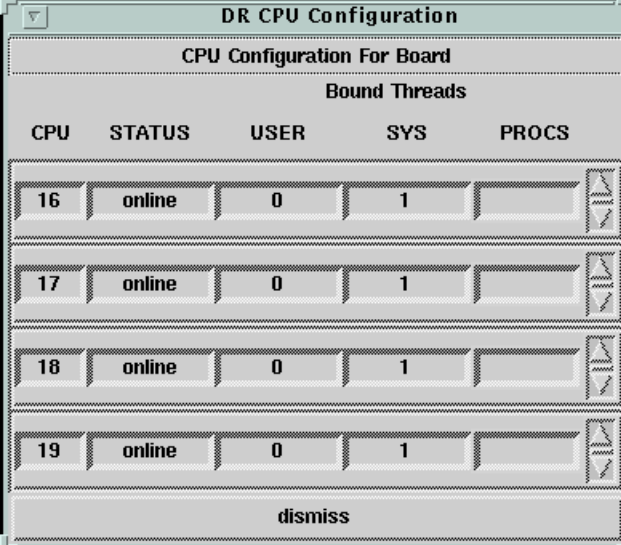


Figure 3-7 Hostview Properties Window—DR Properties

If you set `Auto Update System Information Displays` to `On` (which is the default), the displays are periodically updated. You can set the `Update Interval` to a value, in seconds, to determine how often updates occur. If you set `Auto Update System Information Displays` to `Off`, none of the displays are updated; each display is a snapshot taken at the time the button was pressed. 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 informational displays are active, responsiveness of the DR windows may be degraded. This is especially true when device detail displays are active. Each time an informational display is updated, an RPC call is issued to the `dr_daemon` running on the domain. `dr_daemon` is an iterative RPC server, so each RPC request is run sequentially.

To view processor configuration information, click on the `cpu` button, and the following window is displayed.



The screenshot shows a window titled "DR CPU Configuration" with a sub-header "CPU Configuration For Board". Below this is a section titled "Bound Threads" containing a table with the following data:

CPU	STATUS	USER	SYS	PROCS
16	online	0	1	
17	online	0	1	
18	online	0	1	
19	online	0	1	

At the bottom of the window is a "dismiss" button.

Figure 3-8 Hostview—CPU Configuration Window

For each processor on the selected board, the window shows the numeric ID, status (Online/Offline), and bound threads information. Threads may be bound to a processor via the `pbind(1M)` command. Some operating system device drivers may bind threads to processes to provide better servicing of a device. The number of user and system threads, and the process IDs of the bound threads are displayed.

To view memory configuration information, click the `memory` button, and the following window appears.

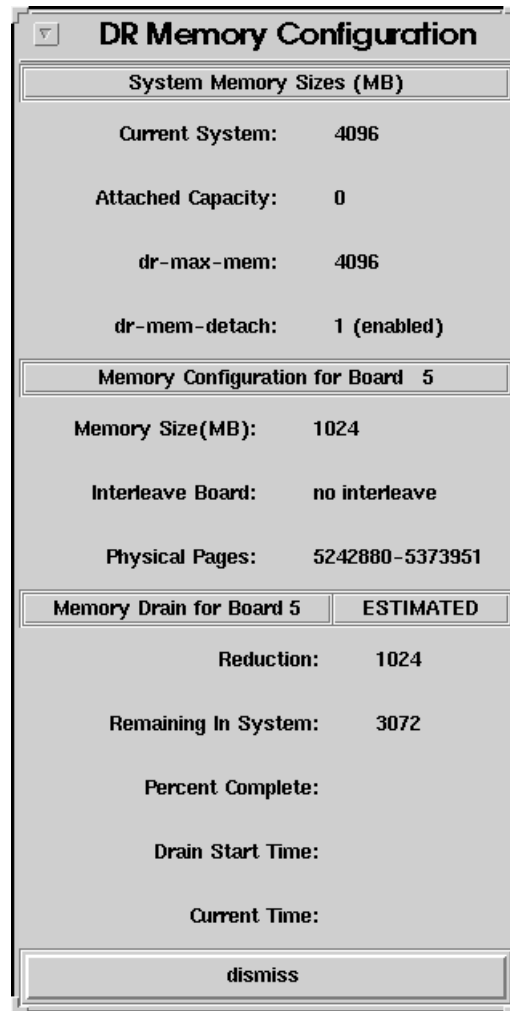


Figure 3-9 Hostview—Memory Configuration Window

The memory configuration window is divided into system memory information, information that pertains to a particular board, and if a Drain operation is in progress, information showing the status of the memory Drain.

The system memory display includes:

- `Current System`—The total size of memory in the system from all boards
- `Attach Capacity`—The amount of memory that can currently be added via a DR Attach operation
- `dr-max-mem`—The value of the OBP variable that determines the maximum amount of memory the system can support; you cannot perform any DR operation unless this variable has a non-zero value

The board memory window shows the amount of memory on the selected board, the board it is interleaved with, and the 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; but, for the sake of simplicity, such areas are not indicated.)

Note – Currently, DR is not able to detach boards that have interleaved memory.

The memory drain display may be in one of the following states:

`Unavailable`, `Estimated`, `In Progress`, or `complete`. `Unavailable` means that a suitable target memory area is not currently available. 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 in time. Note that the estimated values may differ from the in-progress values depending on the system memory usage at the time Drain was started. Once the Drain operation has been started, the state is either `In Progress`, or `complete`. The displayed values are defined as follows.

- `Reduction`—The amount of memory to be removed from system usage when the board is detached.
- `Remaining in System`—The system memory size once the board is detached.
- `Percent Complete`—How far along 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.

As an aid in tracking Drain progress, the Drain operation start time and current time are also displayed.

To view the device configuration information, click on the `device` button. The following window is displayed.

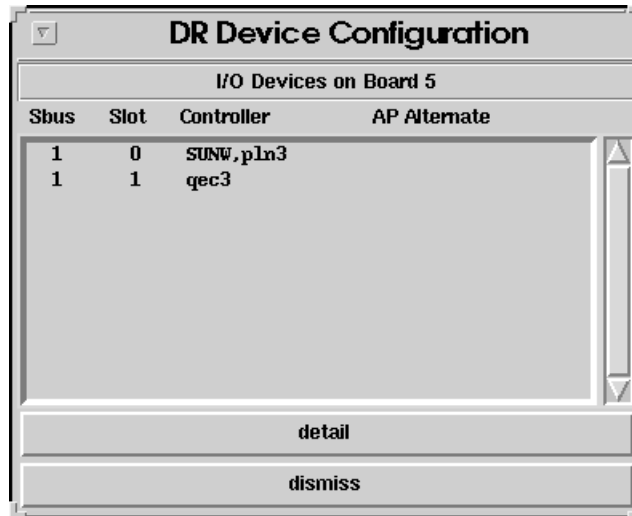


Figure 3-10 Hostview—Device Configuration Window

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 Device Configuration display does not necessarily show all 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 via the `obp` button lists the cards on the board that are configured in the system.

For more detailed information, highlight one or more controller(s) and choose Detail. The following window is displayed for each selected controller.

DR Detail Device			
Device Detail for Shus 1 Slot 1			
Device	Opens	Name	Usage
qe12	-1	qe12	
qe13	-1	qe13	
qe14	-1	qe14	
qe15	-1	qe15	

dismiss

Figure 3-11 Hostview—Device Detail Window

The current usage information for each device is shown. The window includes an open count (if available) and the common name by which the device is known; a disk partition, meta device, or an interface name. 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 and 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 whether it is an AP alternate and whether it is active. For active AP alternates, the usage of the AP meta-device is displayed.

The OBP window contains board configuration information derived from the OBP device tree. This information is less detailed than that available from the other windows described here. For example, in the `Init Attach` state, only the I/O adapters—not the devices attached to those controllers nor the

memory interleave configuration—are known. The OBP window is usually used when a board is in the `Init Attach` state. The OBP window is shown in Figure 3-12.

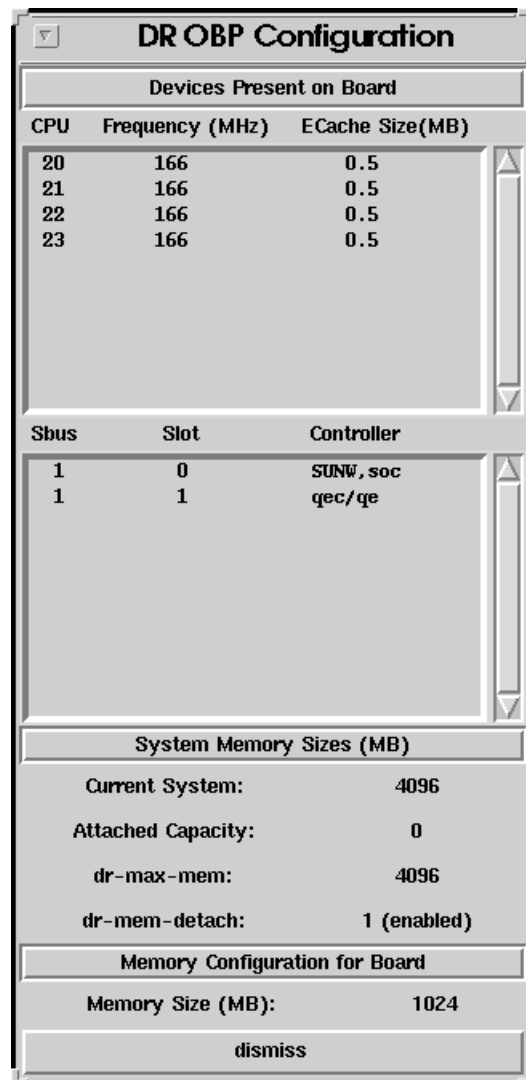


Figure 3-12 Hostview—OBP Window

To see a list of the suspend-unsafe devices across the entire domain—not just those that are resident on the selected system board—click on the `unsafe` button and the following window is displayed.

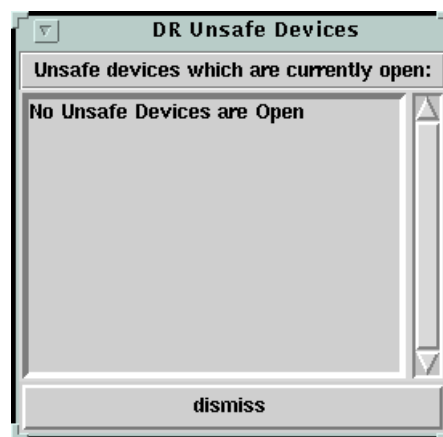


Figure 3-13 Hostview—Unsafe Devices Window

This display shows the suspend-unsafe devices that are currently open. This information is useful for determining the cause of operating system quiesce errors due to unsafe devices being open. In this example, no unsafe devices are open.

Index

Symbols

/etc/system file, 2-15

A

abort button, 3-3, 3-17
active DR operations, only one, 1-2
adding detach-safe devices, 2-17
amount of memory attachable, 3-27
attach, 1-1, 3-1
attach buttons, 3-3
attach, complete, 3-7
attach, dynamic reconfiguration window, 3-5
attach, parameter selection, 3-4
attachable memory, 3-27
attaching a system board, 3-1
attaching with dr(1M), 3-8
auto update screens, 3-24

B

board attach, 1-1, 3-1
board detach, 1-2, 3-12
board, attach, 3-4
buttons
 abort, 3-3, 3-17

complete, 3-3, 3-7, 3-17, 3-20
CPU, 3-25
device, 3-28
dismiss, 3-3, 3-17
drain, 3-17, 3-19
force, 3-17
help, 3-4, 3-18
init attach, 3-3, 3-6
reconfig, 3-3, 3-17
select, 3-4
system information buttons, 3-24

C

complete attach, 3-2, 3-7
complete attach vi dr(1M), 3-10
complete button, 3-3, 3-7, 3-17, 3-20
complete detach, 3-13, 3-20
complete detach via dr(1M), 3-23
CPU button, 3-25
CPU configuration window, 3-25

D

DDI/DKI, 2-16
detach, 1-2, 3-12
detach and network devices, 3-13
detach and non-network devices, 3-14

detach and processors, 3-15
detach buttons, 3-17
detach, parameter selection window, 3-18
detaching with dr(1M), 3-21
detaching with Hostview, 3-18
detach-safe devices, adding, 2-17
detach-safe list, 2-16
device button, 3-28
device configuration window, 3-28
device detail window, 3-29
dismiss button, 3-3, 3-17
DR attach, 1-1, 3-1
DR detach, 1-2, 3-12
DR overview, 1-1
DR parameter selection, 3-4
DR properties window, 3-24
DR unsafe devices, 3-31
dr(1M), attaching via, 3-8
dr_safe list, 2-15
drain, 3-12
drain button, 3-19
drain via dr(1M), 3-22
drain, percent complete, 3-27
draining button, 3-17
drinit, dr(!M) command, 3-8
drinit, dr(1M) command, 3-21
dr-max-mem environment variable, 2-2,
3-27
drshow, dr(1M) command, 3-11
DR-unsafe device, 2-11
dynamic reconfiguraiton window,
detach, 3-19
dynamic reconfiguration window,
attach, 3-5

E

enabling detach, 2-4
environment variables
dr-max-mem, 2-2, 3-27

F

force button, 3-17

H

help button, 3-4, 3-18
Hostview, detaching via, 3-18

I

I/O devices, configuring for detach, 2-4
init attach, 3-2, 3-6
init attach button, 3-3, 3-6
init attach, with dr(1M), 3-9
interval for screen updates, 3-24

M

memory attach capacity, 3-27
memory button
buttons
memory, 3-26
memory configuration window, 3-26
memory draining, detach, 3-12
memory reduction, detach, 3-27
memory remaining in system, 3-27
memory usage, 2-2
memory, configuring for detach, 2-5
memory, total size (all boards), 3-27

N

network devices and detach, 3-13
non-network devices and detach, 3-14

O

OBP window, 3-30
overview of DR, 1-1

P

parameter selection window, detach, 3-18
parameter selection, attach, 3-4

percent complete, drain, 3-27
processors and detach, 3-15
properties window, 3-24

memory configuration, 3-26
OBP, 3-30
properties, 3-24
unsafe devices, 3-31

R

reconfig button, 3-3, 3-17
reduction of memorym detacg, 3-27
refresh interval, 3-24
remaining memory, detach, 3-27

S

select button, 3-4
source domain, attach, 3-4
SSP messages file, 1-2
suspend-bypass devices, 2-15
swap space, configuring for detach, 2-6
system information, viewing, 3-23
system message buffer, 1-2

T

target domain, attach, 3-5

U

unsafe device, 2-11
unsafe devices, 3-31
unsafe devices window, 3-31
update interval, 3-24

V

viewing system information, 3-23

W

windows
CPU configuration, 3-25
detach parameter selection, 3-18
device configuration, 3-28
device detail, 3-29
DR parameter selection, 3-4
dynamic reconfiguration, 3-5

