

Veritas[™] Cluster Server Application Note: Dynamic Reconfiguration of Sun Servers

Solaris

5.0 Maintenance Pack 3



Veritas Cluster Server

Application Note: Dynamic Reconfiguration of Sun Servers

Copyright © 2008 Symantec Corporation. All rights reserved.

Symantec, the Symantec logo, and Veritas are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation/reverse engineering. No part of this document may be reproduced in any form by any means without prior written authorization of Symantec Corporation and its licensors, if any.

THIS 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, SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

The Licensed Software and Documentation are deemed to be “commercial computer software” and “commercial computer software documentation” as defined in FAR Sections 12.212 and DFARS Section 227.7202.

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014
www.symantec.com

Third-party legal notices

Third-party software may be recommended, distributed, embedded, or bundled with this Symantec product. Such third-party software is licensed separately by its copyright holder. All third-party copyrights associated with this product are listed in the accompanying release notes.

Solaris is a trademark of Sun Microsystems, Inc.

Technical support

For technical assistance, visit <http://support.veritas.com> and select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

VCS Application Note: Dynamic Reconfiguration of Sun servers

- [Dynamic reconfiguration in a VCS environment - An overview](#)
- [Supported software and hardware](#)
- [Preparing to perform dynamic reconfiguration](#)
- [Stopping and starting VCS](#)
- [Performing dynamic reconfiguration](#)

Dynamic reconfiguration in a VCS environment - An overview

This application note describes how to perform Dynamic Reconfiguration operations on VCS clustered system domains of the Sun™ servers.

The dynamic reconfiguration operations typically include configuring and unconfiguring CPU/memory boards to and from domains and configuring and unconfiguring I/O boards in a domain. These operations allow switching boards from one domain to another or permit removing a board or card to upgrade or replace it. Dynamic reconfiguration operations can be performed while the operating environment continues to run.

However, a dynamic reconfiguration operation performed on a CPU/memory board that has permanent memory requires that the system domain be temporarily suspended. In this case, VCS must be stopped. Do not use the following procedures to dynamically reconfigure a system board containing a VCS private heartbeat link. If you need to do so, you must stop VCS before proceeding.

For a dynamic reconfiguration operation performed on an I/O board, ensure that all devices that are in use and belong to the I/O board are released i.e. they are not in use by any application modules.

For users of *Veritas SF for Oracle RAC*, it is necessary to stop the Oracle RAC instance within the domain being reconfigured if VCS must be stopped. This permits communications among other RAC instances to occur while the instance in the one domain is temporarily stopped.

See [“Scenarios requiring a VCS shutdown”](#) on page 8.

See [“Stopping and starting VCS”](#) on page 9.

Boards with I/O controllers can be dynamically reconfigured as long as you use VxVM with the Dynamic Multipathing (DMP) feature to manage the shared storage.

The Solaris dynamic reconfiguration utility enables you to reconfigure the resources of system boards so that the boards can be replaced without system downtime.

In such cases, before you can physically remove a board, you must “detach” it, or reconfigure it such that its resources can be disabled and removed from the domain configuration. Likewise, after you have physically replaced a board in a domain, you must “attach” it, or reconfigure it into the domain.

The Sun documentation for dynamic reconfiguration contains comprehensive descriptions of procedures and commands. To avoid damaging system boards and components, you should be familiar with the procedures for their removal and replacement.

Note: Currently, VCS does not support using dynamic reconfiguration in clusters where I/O controllers and storage use Sun's Alternate Pathing (AP).

Supported software and hardware

Supported software

- Solaris 8, 9, and 10.
- Veritas Cluster Server version [2.0, 3.5, 4.0, 4.1, 5.0, 5.0 MP1, 5.0 MP3]
- Veritas Volume Manager (VxVM), as supported by the VCS version
- Veritas File System, as supported by the VCS version

Supported hardware

- SunFire/Enterprise servers (s6800, e12K/15K, e10K, e25K)

Preparing to perform dynamic reconfiguration

Make sure that you determine which devices on the system board will be impacted by the dynamic reconfiguration operations and determine how to mitigate the impact. To be dynamically reconfigured, the boards must satisfy the following conditions:

- Critical resources on the boards must be redundant. For example, boards for which CPUs and memory are redundant can be reconfigured after their function has been replaced and their activity stopped. A CPU board that contains the only CPU in a domain cannot be moved.
- A memory board containing permanent memory, such as the OpenBoot™ PROM or kernel memory, can be moved after the memory has been moved to another board. Dynamic reconfiguration on boards with permanent memory requires VCS to be shut down.
- Disk drives must be accessible via alternate pathways. The Dynamic Multipathing (DMP) feature can provide alternate paths. Before moving a host bus adapter (HBA), switch all the card's functions to an alternate card. An HBA that controls sole access to an active drive cannot be moved.
- Activity on a PCI card must be stopped before the card is removed.

Stopping and starting VCS

It is necessary to stop VCS and unconfigure GAB and LLT in certain circumstances.

Scenarios requiring a VCS shutdown

VCS must be shutdown under the following circumstances:

- When performing dynamic reconfiguration on a system board (CPU/Memory board) with permanent memory.
- When the I/O board requiring reconfiguration contains all the private network links used by the domain.
- When the I/O board contains the only public network links used by the domain.
- When the I/O board contains all of the paths to a storage device.

The necessity of performing a VCS shutdown can be reduced by some device layout planning before clustering the domains.

CPU/Memory boards with permanent memory

If the CPU/memory board to be removed contains permanent memory, the operating system's function must be suspended to permit dynamic reconfiguration to occur. In such a case, VCS must be stopped.

However, you do not need to stop VCS when you are performing dynamic reconfiguration on a board that does not contain permanent memory. Typically, in a domain with multiple CPU/memory boards, one board has permanent memory, while the others do not. When you are performing dynamic reconfiguration to add a new board to the domain, the existing functions in the domain are not affected by the dynamic addition of a new CPU/memory board.

Note: If you must reconfigure multiple boards and a board with permanent memory is among them, reconfigure the board with permanent memory last. This sequence ensures minimum VCS downtime.

You can perform the following steps to determine if the CPU/memory board has permanent memory.

To determine if the CPU/memory board has permanent memory

- 1 Log into the domain as domain administrator.

- 2 List the boards with permanent memory in the domain by entering the following command:

```
# cfgadm -av | grep permanent
SB2::memory connected configured ok base address
0x1e000000000, 16777216 KBytes total, 2001200 KBytes
permanent
```

The output in the example shows SB2 to contain permanent memory. Before this board can be dynamically reconfigured, VCS must be stopped.

See [“Stopping and starting VCS”](#) on page 9.

Other CPU/memory boards in the domain do not contain permanent memory and may be dynamically reconfigured without stopping VCS.

Stopping and starting VCS

This section contains the procedures for stopping VCS if it is required for dynamic reconfiguration and the procedures for starting VCS if it has been stopped for dynamic reconfiguration.

- See [“Stopping VCS in a standard environment”](#) on page 9.
- See [“Restarting VCS in a standard environment”](#) on page 11.
- See [“Stopping VCS in Veritas SF for Oracle RAC environment”](#) on page 11.
- See [“Restarting VCS in Veritas SF for Oracle RAC environment”](#) on page 13.

Stopping VCS in a standard environment

When you dynamically reconfigure CPU/Memory boards and I/O boards, it may be necessary, in some circumstances, to stop VCS in the domain.

Applications running on clusters of three or more domains remain highly available on two or more domains if VCS operation must be stopped on one domain. In a cluster of two domains, the applications running during reconfiguration are not highly available when VCS must be stopped on one of the domains.

If you are running *Veritas SF for Oracle RAC*, see [“Stopping VCS in Veritas SF for Oracle RAC environment”](#) on page 11.

To stop VCS in a standard environment

- 1 Log in as administrator to the domain (*wildcat*, for example) you are reconfiguring.
- 2 List the VCS service groups to determine which are online on the domain.

```
# hagrps -list
```

- 3 If you can switch the service groups running on the domain to another domain (*cheetah*, for example), do the following:
 - a Switch the service groups.
`# hagrps -switch service_grp_name -to cheetah`
 - b Verify the service groups are offline on *wildcat*.
`# hastatus`
 - c Stop VCS on *wildcat*.
`# hstop -local`
- 4 If you cannot switch the online service groups to another system, freeze each of them for the duration of dynamic reconfiguration as follows:
 - a Make the VCS configuration writable.
`# haconf -makerw`
 - b Freeze each of the service groups persistently.
`# hagrps -freeze service_grp_name -persistent`
 - c Verify the groups are frozen.
`# hagrps display | grep Frozen`
 - d Make the configuration read-only.
`# haconf -dump -makero`
 - e Stop VCS.
`# hstop -local -force`
- 5 Unconfigure GAB.
`# /sbin/gabconfig -U`
- 6 Unconfigure LLT.
`# /sbin/lltconfig -U`
 Answer “**y**” to confirm that you want to stop LLT.
- 7 Remove the GAB and LLT modules from the kernel.
 - a Determine the IDs of the GAB and LLT modules:
`# modinfo | egrep "gab|llt"`

```
305 78531900 30e 305 1 gab
292 78493850 30e 292 1 ll
```
 - b Unload the GAB and LLT modules based on their module IDs:
`# modunload -i 305`
`# modunload -i 292`
- 8 You can begin performing dynamic reconfiguration.

Restarting VCS in a standard environment

If you are ready to restart VCS in the domain where you are performing dynamic reconfiguration, use the following procedure. If you are running *Veritas SF for Oracle RAC*, and are ready to restart VCS, see [“Restarting VCS in Veritas SF for Oracle RAC environment”](#) on page 13.

To restart LLT, GAB, and VCS

- 1 Restart LLT.
`# /etc/rc2.d/S701lt start`
- 2 Restart GAB.
`# /etc/rc2.d/S92gab start`
- 3 Start VCS.
`# hstart`
- 4 Verify GAB and VCS are started.
`# /sbin/gabconfig -a`
GAB Port Memberships
=====

```
Port a gen 4a1c0001 membership 012
Port h gen g8ty0002 membership 012
```

To bring service groups online

- 1 Determine which service groups are frozen.
`# hagr -display | grep Frozen`
- 2 Make the configuration writable.
`# haconf -makerw`
- 3 Unfreeze the frozen service groups.
`# hagr -unfreeze service_grp_name -persistent`
- 4 Make the configuration read-only.
`# haconf -dump -makero`

Stopping VCS in Veritas SF for Oracle RAC environment

If you must stop VCS on a domain where Veritas SF for Oracle RAC is running, the Oracle RAC application on the domain being reconfigured must be brought offline. In addition, the GAB, LLT, LMX, and VXFEN modules must be unconfigured. Performing these steps ensures that other instances do not attempt communication with the stopped instance. This could cause the application to hang, when the instance does not respond.

To stop VCS in a Veritas SF for Oracle RAC environment

- 1 Log in as administrator to the domain being reconfigured (`wildcat`, for example).
- 2 List the configured VCS service groups and see which are online in the domain:

```
# hagrps -list
```
- 3 Based on the output of [step 2](#), bring each service group that is online to offline in the domain `wildcat`. Use the following command:

```
# hagrps -offline service_grp_name -sys wildcat
```
- 4 Stop VCS.

```
# hastop -local
```

In addition to port `h`, this command stops the CVM drivers using ports `v` and `w`.
- 5 If any CFS file systems outside of VCS control are mounted, unmount them.
- 6 Stop and unconfigure the drivers required by DBE/AC:

```
# cd /opt/VRTSvcs/rac
# ./unload_drv
Unloading qlong
Unloading odm
Unloading fdd
Unloading vxportal
Unloading vxfs
```
- 7 Unconfigure the VCSMM and I/O fencing drivers, which use ports `b` and `o`, respectively:

```
# /sbin/vxfenconfig -U
# /sbin/vcsmmconfig -U
```
- 8 Unconfigure the LMX driver:

```
# /sbin/lmxconfig -U
```
- 9 Verify that the drivers `h`, `v`, `w`, `f`, `q`, `d`, `b`, and `o` are stopped. They should not show memberships when you use the `gabconfig -a` command:

```
# gabconfig -a
GAB Port Memberships
=====
Port a gen 4a1c0001 membership 01
```
- 10 Unload the VCSMM, I/O fencing, and LMX modules.
 - a Determine the module IDs for VCSMM, I/O fencing, and LMX:

```
# modinfo | egrep "lmx|vxfen|vcsmm"
```

```
237 783e4000 25497 237 1 vcsmm (VERITAS Membership Manager)
238 78440000 263df 238 1 vxfen (VERITAS I/O Fencing)
239 7845a000 12b1e 239 1 lmx (LLT Mux 3.5B2)
```

- b** Unload the VCSMM, I/O fencing, and LMX modules based on their module IDs:


```
# modunload -i 237
# modunload -i 238
# modunload -i 239
```
- 11 Unconfigure GAB


```
# /sbin/gabconfig -U
```
- 12 Unconfigure LLT


```
# /sbin/lltconfig -U
```
- 13 Remove the GAB and LLT modules from the kernel.
 - a** Determine the IDs of the GAB and LLT modules:


```
# modinfo | egrep "gab|llt"
305 78531900 30e 305 1 gab
292 78493850 30e 292 1 llT
```
 - b** Unload the GAB and LLT modules based on their module IDs:


```
# modunload -i 305
# modunload -i 292
```
- 14 You can begin performing dynamic reconfiguration.

Restarting VCS in Veritas SF for Oracle RAC environment

If you used the procedure described in “[Stopping VCS in Veritas SF for Oracle RAC environment](#)” on page 11 before dynamically reconfiguring a CPU/memory board, use the following procedures to restart VCS and bring the service groups on the domain online.

To restart LLT, GAB, VCS, and DBE/AC processes

- 1 Restart LLT.


```
# /etc/rc2.d/S70llt start
```
- 2 Restart GAB.


```
# /etc/rc2.d/S92gab start
```
- 3 Restart the LMX driver.


```
# /etc/rc2.d/S71lmx start
```
- 4 Restart the VCSMM driver.


```
# /etc/rc2.d/S98vcsmm start
```
- 5 Restart the VXFEN driver.


```
# /etc/rc2.d/S97vxfen start
```
- 6 Restart the ODM driver.


```
# mount /dev/odm
```

- 7 Start VCS.
`# hastart`
 - 8 Verify that the CVM service group is online.
`# hagrps -state cvm`
 - 9 Verify the GAB memberships required for DBE/AC for Oracle9i RAC are configured.
`# /sbin/gabconfig -a`
GAB Port Memberships
=====
- ```
Port a gen 4a1c0001 membership 012
Port b gen g8ty0002 membership 012
Port d gen 40100001 membership 012
Port f gen f1990002 membership 012
Port h gen g8ty0002 membership 012
Port o gen f1100002 membership 012
Port q gen 28d10002 membership 012
Port v gen 1fc60002 membership 012
Port w gen 15ba0002 membership 012
```
- 10 Bring the service groups that had been take offline in [step 3](#) on page 11 online.  
`# hagrps -online service_grp_name -sys wildcat`

## Performing dynamic reconfiguration

Following are the procedures to perform dynamic reconfigurations on SunFire s6800; e12K/15K/25K and Sun Enterprise e10K.

- See [“Performing dynamic reconfiguration on SunFire \(s6800; e12K/15K/ e25K\)”](#) on page 14.
- See [“Performing dynamic reconfiguration on SunEnterprise 10K”](#) on page 36.

## Performing dynamic reconfiguration on SunFire (s6800; e12K/15K/ e25K)

You may dynamically reconfigure CPU/memory boards, I/O boards and PCI on I/O boards for SunFire s6800/e12K/e15K/e25K.

- See [“Performing dynamic reconfiguration on a CPU/memory board”](#) on page 15.
- See [“Performing dynamic reconfiguration on PCI cards on I/O boards”](#) on page 19.
- See [“Performing dynamic reconfiguration on I/O boards”](#) on page 23.

## Performing dynamic reconfiguration on a CPU/memory board

You may want to remove a CPU/memory board that is malfunctioning or you may want to reconfigure a board from one domain to another where it is needed more.

To reassign a board from one domain to another, you must unconfigure it from one domain and reassign it to another domain. This can be done without physically removing the board from its slot. To replace a board, however, you must unconfigure it from one domain, physically remove it, add its replacement board and reconfigure it to the domain.

Use the following procedures to dynamically reconfigure a CPU/memory board.

### To determine the status of the board you are reconfiguring

- 1 If necessary, log in as the administrator to the domain containing the CPU/memory board.
- 2 Determine the attachment point of the board you are removing:  

```
cfigadm
Ap_Id Type Receptable Occupant Cond
.
N0.SB2 CPU connected configured ok
.
```
- 3 Make sure you have checked whether the board has permanent memory. See [“To determine if the CPU/memory board has permanent memory”](#) on page 8.
- If the board in the domain you want to dynamically reconfigure contains permanent memory, be sure you have first stopped VCS using the procedures described in [“Stopping and starting VCS”](#) on page 9.
  - See [“Stopping VCS in a standard environment”](#) on page 9.
  - See [“Restarting VCS in a standard environment”](#) on page 11.
  - See [“Stopping VCS in Veritas SF for Oracle RAC environment”](#) on page 11.
  - See [“Restarting VCS in Veritas SF for Oracle RAC environment”](#) on page 13.
- If the board you want to reconfigure does not contain permanent memory, you can proceed to dynamically reconfigure it.

### To unbind processes bound to CPU on the board

- 1 To determine if any processes are bound to a CPU, enter:  

```
pbind -q
```

- 2 If a processes is bound to the board, the output indicates the process ID and the ID number of the CPU.

```
process id 650: 0
```

- 1 If you see no output or see output showing no processes bound to a CPU on the board, you are reconfiguring, perform the steps in [“To unconfigure the board”](#) on page 16.

- 2 Unbind all processes bound to the CPU on the board. For example, enter:

```
pbind -u 650
```

- 3 Rebind the processes to a processor on another board, if necessary. For example, bind process 650 to processor with ID 9, which is on another board, using the command:

```
pbind -b 650 9
```

- 4 If you attempt to unconfigure a board with processes bound to it, you receive a message that resembles:

```
cfgadm: Hardware specific failure: unconfigure SB15: Failed to
off-line:dr@0:SB15::cpu3
```

### To unconfigure the board

- 1 Unconfigure and disconnect the board:

```
cfgadm -v -c disconnect SB2
```

- 2 If the board does not contain permanent memory, the command's output resembles the following with slight variations for each server:

```
request delete capacity (4 cpus)
request delete capacity (2097152 pages)
request delete capacity SB2 done
request offline SUNW_cpu/cpu448
request offline SUNW_cpu/cpu449
request offline SUNW_cpu/cpu450
request offline SUNW_cpu/cpu451
request offline SUNW_cpu/cpu448 done
request offline SUNW_cpu/cpu449 done
request offline SUNW_cpu/cpu450 done
request offline SUNW_cpu/cpu451 done
unconfigure SB2
unconfigure SB2 done
notify remove SUNW_cpu/cpu448
notify remove SUNW_cpu/cpu449
notify remove SUNW_cpu/cpu450
notify remove SUNW_cpu/cpu451
notify remove SUNW_cpu/cpu448 done
notify remove SUNW_cpu/cpu449 done
notify remove SUNW_cpu/cpu450 done
notify remove SUNW_cpu/cpu451 done
disconnect SB2
disconnect SB2 done
```



```
poweroff SB2
poweroff SB2 done
unassign SB2 skipped
```

Skip to [step 4](#).

**3** If the board has permanent memory, the system prompts you to proceed:

System may be temporarily suspended; proceed (yes/no)?

If the answer is “yes,” dynamic reconfiguration proceeds. The system is suspended during reconfiguration. When the system resumes operation on another board, the board you are reconfiguring is disconnected. If the disconnect operation succeeds, the output resembles the following with slight variations for different servers:

```
request suspend SUNW_OS
request suspend SUNW_OS done
request delete capacity (2097152 pages)
request delete capacity SB15 done
request offline SUNW_cpu/cpu480
request offline SUNW_cpu/cpu481
request offline SUNW_cpu/cpu482
request offline SUNW_cpu/cpu483
request offline SUNW_cpu/cpu480 done
request offline SUNW_cpu/cpu481 done
request offline SUNW_cpu/cpu482 done
request offline SUNW_cpu/cpu483 done
unconfigure SB15
unconfigure SB15 done
notify remove SUNW_cpu/cpu480
notify remove SUNW_cpu/cpu481
notify remove SUNW_cpu/cpu482
notify remove SUNW_cpu/cpu483
notify remove SUNW_cpu/cpu480 done
notify remove SUNW_cpu/cpu481 done
notify remove SUNW_cpu/cpu482 done
notify remove SUNW_cpu/cpu483 done
disconnect SB15
disconnect SB15 done
poweroff SB15
```

Skip to [step 4](#).

---

**Note:** If there are real-time processes running on the board you are unconfiguring, the disconnect operation may not succeed. You must stop these processes in the appropriate manner before continuing with dynamic reconfiguration.

---

**4** If the board has real-time processes that must be stopped, the dynamic reconfiguration operation fails, indicating the PID of those processes that are running. There may be slight variations in output for different Sun Enterprise servers.

For example:

```
.
.
notify remove SUNW_cpu/cpu481 done
notify remove SUNW_cpu/cpu482 done
notify remove SUNW_cpu/cpu483 done
cfgadm: Hardware specific failure: unconfigure SB15:
Cannot
quiesce realtime thread: 621
```

- 5 To determine the name of the processes, use the command:  

```
ps -ef | grep PID
```
- 6 Stop the process in the appropriate manner. For example, the processes in our example must be stopped using the `kill` command:  

```
kill -9 PID
```
- 7 Retry the command in [step 1](#).
- 8 To verify the board is disconnected and unconfigured, use the `cfgadm` command:  

```
cfgadm
```

| Ap_Id  | Type | Receptable   | Occupant     | Cond    |
|--------|------|--------------|--------------|---------|
| .      |      |              |              |         |
| N0.SB2 | CPU  | disconnected | unconfigured | unknown |
| .      |      |              |              |         |

Now you can remove the board from the slot, or reassign it to another domain.

---

**Note:** Do not remove the board until you have verified it is disconnected.

---

- 9 If you are replacing the board immediately, see [“To add a board to a domain”](#) on page 18. Otherwise, return the cluster to operation without replacing the disconnected CPU/memory board using the procedure in the following section.

#### **To add a board to a domain**

- 1 Log in as administrator to the domain where you plan to add or configure the boards.
- 2 If you are adding a new or a replacement board to a domain (for example, `wildcat`), verify the state of the slot to contain the board. To be configured with a new board, the slot must have the following states and condition:
  - Receptacle state: empty
  - Occupant state: unconfigured
  - Condition: unknown

Verify this by using the `cfgadm` command to list the slots, as in the following example. In the `wildcat` domain, slot SB2 is to contain the CPU board:

- 3 Use the `cfgadm` command to connect and configure a CPU or memory board:

```
cfgadm -v -c configure SBx
For example:
cfgadm -v -c configure SB2
assign SB2
assign SB2 done
poweron SB2
poweron SB2 done
test SB2
test SB2 done
connect SB2
connect SB2 done
configure SB2
configure SB2 done
notify online SUNW_cpu/cpu448
notify online SUNW_cpu/cpu449
notify online SUNW_cpu/cpu450
notify online SUNW_cpu/cpu451
notify add capacity (4 cpus)
notify add capacity (2097152 pages)
notify add capacity SB2 done
```

- 4 Verify the new board has been connected and configured using the command `cfgadm`. For example:

```
cfgadm
Ap_Id Type Receptable Occupant Cond
.
SB2 CPU connected configured ok
```

## Performing dynamic reconfiguration on PCI cards on I/O boards

A card containing an HBA can be removed and replaced on an I/O board. If a failed HBA has been used with other adapters on separate cards in a dynamic multipathing (DMP) configuration, I/O can proceed through the alternate path and VCS need not be stopped.

### To determine the status of the card you are unconfiguring

- 1 Log into the domain as the administrator. For the following example, the I/O board is in the `wildcat` domain.
- 2 Check the status of the boards. Use the `cfgadm` command.

```
cougax# cfgadm
The output resembles:
```

### For Solaris 10:

| Ap_Id    | Type       | Receptacle | Occupant     | Condition |
|----------|------------|------------|--------------|-----------|
| IO4      | HPCI       | connected  | configured   | ok        |
| IO4_C3V0 | fibre/hp   | connected  | configured   | ok        |
| IO4_C3V1 | pci-pci/hp | connected  | configured   | ok        |
| IO4_C5V0 | pci-pci/hp | connected  | configured   | ok        |
| IO4_C5V1 | fibre/hp   | connected  | configured   | ok        |
| SB7      | CPU        | connected  | configured   | ok        |
| SB8      | CPU        | connected  | configured   | ok        |
| c0       | scsi-bus   | connected  | configured   | unknown   |
| c1       | scsi-bus   | connected  | unconfigured | unknown   |
| c2       | fc         | connected  | unconfigured | unknown   |
| c3       | fc         | connected  | unconfigured | unknown   |
| c4       | fc-fabric  | connected  | configured   | unknown   |
| c5       | fc         | connected  | unconfigured | unknown   |

```
cougar# uname -a
SunOS cougar 5.10 Generic_118833-17 sun4u sparc
SUNW,Sun-Fire-15000
cougar#
```

### For Solaris 9:

**jaguar# cfigadm**

| Ap_Id              | Type        | Receptacle | Occupant     | Condition |
|--------------------|-------------|------------|--------------|-----------|
| IO8                | HPCI        | connected  | configured   | ok        |
| IO17               | HPCI        | connected  | configured   | ok        |
| SB6                | V3CPU       | connected  | configured   | ok        |
| c0                 | scsi-bus    | connected  | configured   | unknown   |
| c1                 | scsi-bus    | connected  | unconfigured | unknown   |
| c2                 | fc-fabric   | connected  | configured   | unknown   |
| c3                 | fc-fabric   | connected  | configured   | unknown   |
| c4                 | scsi-bus    | connected  | configured   | unknown   |
| c5                 | scsi-bus    | connected  | configured   | unknown   |
| c6                 | fc          | connected  | unconfigured | unknown   |
| pcisch0:e08b1slot1 | ethernet/hp | connected  | configured   | ok        |
| pcisch1:e08b1slot0 | mult/hp     | connected  | configured   | ok        |
| pcisch2:e08b1slot3 | pci-pci/hp  | connected  | configured   | ok        |
| pcisch3:e08b1slot2 | pci-pci/hp  | connected  | configured   | ok        |
| pcisch4:e17b1slot1 | pci-pci/hp  | connected  | configured   | ok        |
| pcisch5:e17b1slot0 | pci-pci/hp  | connected  | configured   | ok        |
| pcisch6:e17b1slot3 | fibre/hp    | connected  | configured   | ok        |
| pcisch7:e17b1slot2 | pci-pci/hp  | connected  | configured   | ok        |

```
jaguar# uname -a
SunOS jaguar 5.9 Generic_118558-34 sun4u sparc
SUNW,Sun-Fire-15000
jaguar#
```

In case of Sol 10, the reporting of IO board slot names makes it somewhat easier to discover the relationship between physical and logical devices because slots on the IO boards are also numbered using the C[35]V[01] notation.

### To remove a PCI card

- 1 Disable the controllers on the I/O system card using the `vxdatapadm` command:  

```
vxdatapadm disable ctlr=ctlr
vxdatapadm disable ctlr=c3
```

 If the card has more than one controller, repeat this command for each controller on the card.
- 2 Disconnect the card:  

```
cfgadm -v -c disconnect pcisch1:sg8slot0
```
- 3 Check the states and the condition of the card using the `cfgadm` command:  

```
cfgadm
```

 The disconnected card must have the following states and condition:
  - Receptacle state: disconnected
  - Occupant state: unconfigured
  - Condition: unknown
- 4 Remove the disconnected card only if it is powered off.

### To add a card

- 1 Verify that the slot you selected can accept a device, such as a PCI card. To accept a device, the slot must have the following states and condition:
  - Receptacle state: empty or disconnected
  - Occupant state: unconfigured
  - Condition: unknown

Verify this by using the `cfgadm` command to list all of the system boards, as in the following example:

The output resembles:

For Solaris 10:

**cougar# cfgadm**

| Ap_Id        | Type       | Receptacle | Occupant     | Condition |
|--------------|------------|------------|--------------|-----------|
| IO4 HPCI     |            | connected  | configured   | ok        |
| IO4_C3V0     | fibre/hp   | connected  | configured   | ok        |
| IO4_C3V1     | pci-pci/hp | connected  | configured   | ok        |
| IO4_C5V0     | pci-pci/hp | connected  | configured   | ok        |
| IO4_C5V1     | fibre/hp   | connected  | configured   | ok        |
| SB7 CPU      |            | connected  | configured   | ok        |
| SB8 CPU      |            | connected  | configured   | ok        |
| c0 scsi-bus  |            | connected  | configured   | unknown   |
| c1 scsi-bus  |            | connected  | unconfigured | unknown   |
| c2 fc        |            | connected  | unconfigured | unknown   |
| c3 fc        |            | connected  | unconfigured | unknown   |
| c4 fc-fabric |            | connected  | configured   | unknown   |
| c5 fc        |            | connected  | unconfigured | unknown   |

```
cougar# uname -a
SunOS cougar 5.10 Generic_118833-17 sun4u sparc
SUNW,Sun-Fire-15000
cougar#
```

For Solaris 9:

```
jaguar# cfgadm
```

| Ap_Id              | Type        | Receptacle | Occupant     | Condition |
|--------------------|-------------|------------|--------------|-----------|
| IO8                | HPCI        | connected  | configured   | ok        |
| IO17               | HPCI        | connected  | configured   | ok        |
| SB6                | V3CPU       | connected  | configured   | ok        |
| c0                 | scsi-bus    | connected  | configured   | unknown   |
| c1                 | scsi-bus    | connected  | unconfigured | unknown   |
| c2                 | fc-fabric   | connected  | configured   | unknown   |
| c3                 | fc-fabric   | connected  | configured   | unknown   |
| c4                 | scsi-bus    | connected  | configured   | unknown   |
| c5                 | scsi-bus    | connected  | configured   | unknown   |
| c6                 | fc          | connected  | unconfigured | unknown   |
| pcisch0:e08b1slot1 | ethernet/hp | connected  | configured   | ok        |
| pcisch1:e08b1slot0 | mult/hp     | connected  | configured   | ok        |
| pcisch2:e08b1slot3 | pci-pci/hp  | connected  | configured   | ok        |
| pcisch3:e08b1slot2 | pci-pci/hp  | connected  | configured   | ok        |
| pcisch4:e17b1slot1 | pci-pci/hp  | connected  | configured   | ok        |
| pcisch5:e17b1slot0 | pci-pci/hp  | connected  | configured   | ok        |
| pcisch6:e17b1slot3 | fibre/hp    | connected  | configured   | ok        |
| pcisch7:e17b1slot2 | pci-pci/hp  | connected  | configured   | ok        |

```
jaguar# uname -a
SunOS jaguar 5.9 Generic_118558-34 sun4u sparc
SUNW,Sun-Fire-15000
jaguar#
```

In case of Sol 10, the reporting of IO board slot names makes it somewhat easier to discover the relationship between physical and logical devices because slots on the IO boards are also numbered using the C[35]V[01] notation.

- 2 Add the replacement PCI card to the empty card slot.
- 3 To configure the new card, use the `cfgadm` command. For example:  
 For s6800:  

```
cfgadm -c configure pcisch1:sg8slot0
```

 For e12K/15K:  

```
cfgadm -c configure pcisch1:e15b1slot0
```

 After the system configures and tests the board, it displays a message in the domain console log indicating the configuration of the components.
- 4 Check the states and the condition of the board using the `cfgadm` command; it must be “connected,” “configured,” and “ok.”
- 5 Enable the controller for the HBA:  

```
vxdatapadm enable ctrlr=ctrlr
```

```
vxdatapadm enable ctrlr=c3
```

Note that this command succeeds if the controller is accessible to the domain and I/O can be performed on it.

## Performing dynamic reconfiguration on I/O boards

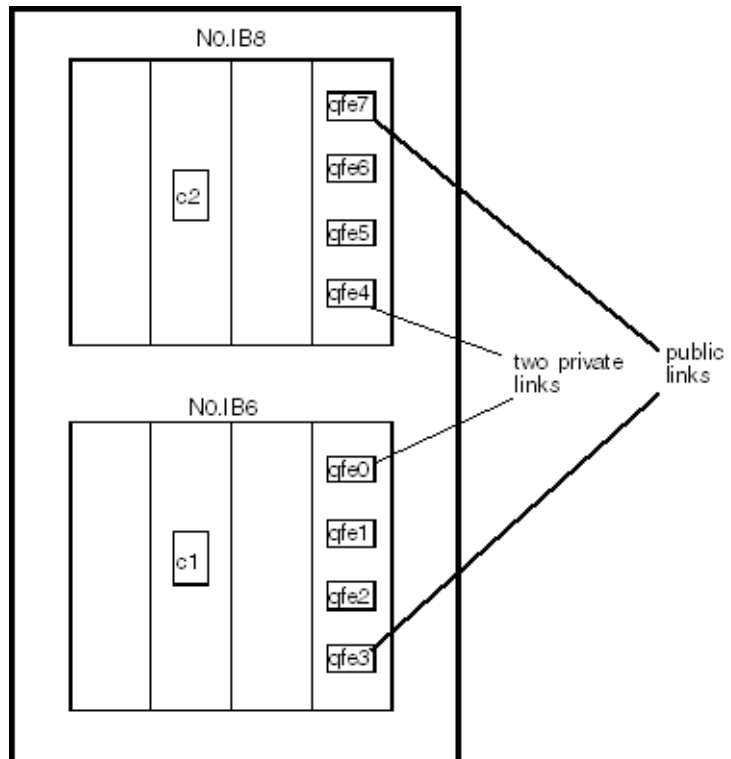
Under certain circumstances, you must stop VCS on the domain where you are reconfiguring a board.

See [“Scenarios requiring a VCS shutdown”](#) on page 8.

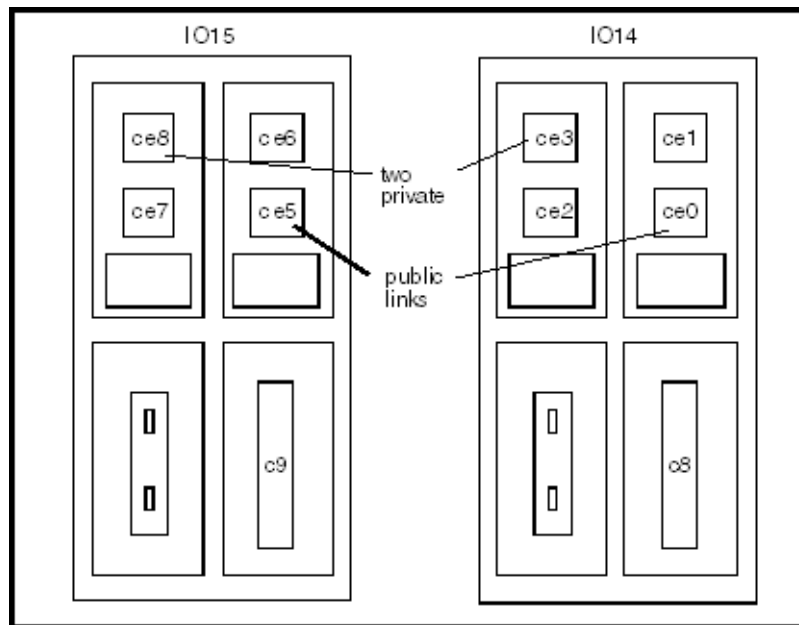
### For s6800:

In the following scenario, a cluster consists of the wildcat and the leopard domains. The cluster is running service groups on the wildcat domain, which includes I/O boards N0.IB8 and N0.IB6. N0.IB8 requires dynamic reconfiguration because of a malfunctioning component. The domain leopard includes I/O boards IO14 and IO15. The disk controllers and NICs are labeled in the following diagrams.

### Domain: wildcat



### Domain: Leopard

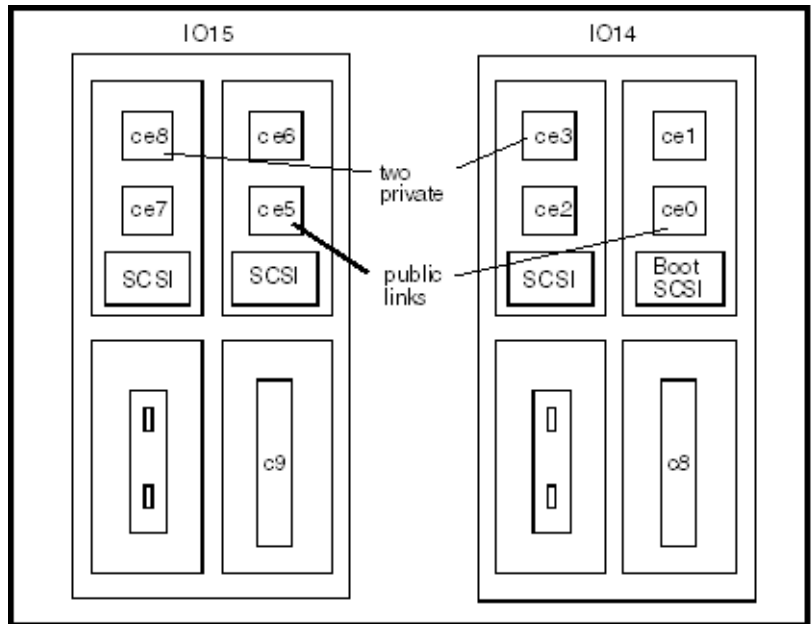


### For e12K/15K/25K:

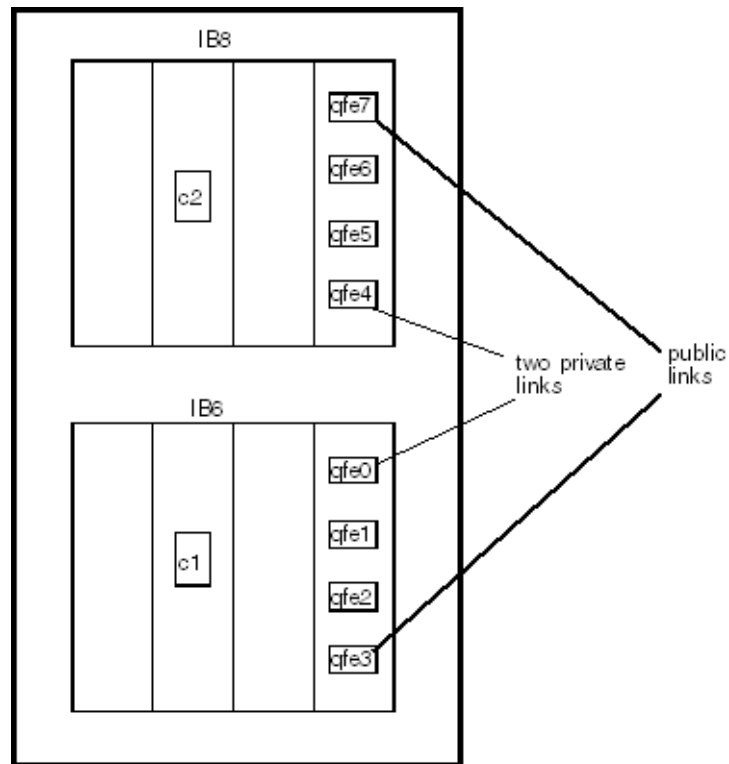
In the following scenario, a cluster consists of the leopard and the S6800f0 domains. The cluster is running service groups on the leopard domain, which includes I/O boards IO14 and IO15. IO15 requires dynamic reconfiguration because of a malfunctioning component. The domain S6800f0 includes I/O boards IB8 and IB6. The disk controllers and NICs are labeled in the following diagrams.



**Domain: Leopard**



### Domain: S6800f0



The highlights of the procedure to dynamically reconfigure the I/O boards (N0.IB8 board and IO15 board) in the wildcat and leopard domains for s6800 and e12K/15K/25K respectively include:

- ✓ Disabling all the active controllers on the board.
- ✓ Disabling all the NIC devices used for private communications on the board
- ✓ Disabling all the NIC devices used for public communications on the board
- ✓ Disabling the IO board and removing it
- ✓ Adding the replacement IO board
- ✓ Enabling the replacement board
- ✓ Enabling the public NIC devices
- ✓ Enabling the private NIC devices
- ✓ Enabling the active controllers

**To verify the status of the cluster before dynamic reconfiguration**

- 1 Use the VCS command `hastatus -sum` to verify the current state of the service groups in the cluster. Use the command before reconfiguring the I/O board and after reconfiguration to verify the cluster's state. The output is as follows with slight variations for the different Sun servers.

```
-- SYSTEM STATE
-- System State Frozen
A leopard RUNNING 0
A s6800f0 RUNNING 0

-- GROUP STATE
-- Group System Probed AutoDisabled State
B ServiceGroupA leopard Y N ONLINE
B ServiceGroupA s6800f0 Y N OFFLINE
B cvm leopard Y N ONLINE
B cvm s6800f0 Y N ONLINE
```

- 2 For s6800: By using the `cfgadm -lv` command, you can show the I/O boards and cards in the `wildcat` domain. For example:

```
cfgadm -lv
```

In the output (not shown), the board `N0.IB8` is reported to be `connected`, `configured`, and `ok`. In addition, the condition of each of the slots on `N0.IB8` are reported.

For e12K/15K: By using the `cfgadm -al` command, you can show the I/O boards and cards in the `leopard` domain. For example:

```
cfgadm -al
```

```
Ap_Id Type Receptacle Occupant
Condition
IO14 HPCI connected configured ok
IO14::pci0 io connected configured ok
IO14::pci1 io connected configured ok
IO14::pci2 io connected configured ok
IO14::pci3 io connected configured ok
IO15 HPCI connected configured ok
IO15::pci0 io connected configured ok
IO15::pci1 io connected configured ok
IO15::pci2 io connected configured ok
IO15::pci3 io connected configured ok
SB14 CPU connected configured ok
SB14::cpu0 cpu connected configured ok
.
.
.
pcisch1:e14b1slot0 fibre/hp connected configured ok
pcisch2:e14b1slot3 pci-pci/hp connected configured ok
pcisch3:e14b1slot2 ethernet/hp connected configured ok
pcisch4:e15b1slot1 pci-pci/hp connected configured ok
pcisch5:e15b1slot0 fibre/hp connected configured ok
pcisch6:e15b1slot3 pci-pci/hp connected configured ok
```

```
pcisch7:e15b1slot2 ethernet/hp connected configured ok
```

**To determine the controllers on a board**

- 1 Use the command `vxddmpadm listctlr all` to determine all controllers in the domain. For example, on the leopard domain:

```
vxddmpadm listctlr all
CTLR-NAME ENCLR-TYPE STATE ENCLR-NAME
=====
c0 Disk ENABLED Disk
c9 HDS9960 ENABLED HDS99600
c8 HDS9960 ENABLED HDS99600
```

- 2 To determine which controllers are on a specific board, for example IO15, use the following commands to display information about the disks in the domain, their controllers, and the location of the controllers on the IO boards.
- a Use the command `cfgadm -lv`, which provides a verbose listing of all boards in the domain. In the output, you can see the device slots listed for the board IO15.

```
cfgadm -lv
```

In the following example (not all output is shown) the listing might contain lines that resemble:

```
.
pcish4:e15b1slot1 . . .
/devices/pci@1fc,700000:e15b1slot1
pcish5:e15b1slot0 . . .
/devices/pci@1fc,600000:e15b1slot0
pcish6:e15b1slot3 . . .
/devices/pci@1fd,700000:e15b1slot3
pcish7:e15b1slot2 . . .
/devices/pci@1fd,600000:e15b1slot2
.
```

The listing indicates that the device labeled `pci@1fc` is used by slots 0 and 1 of board 15, the device labeled `pci@1fd` is used by slots 3 and 2.

- b Using the `format` command in the domain, you can list the disk devices. The listing may be lengthy, but in the output, the controller, indicated by “c#” in the first two characters of the device name, corresponds to a device that is listed in the previous command (step a). For example:

```
format
c0t0d0 <SUN18G /pci@1dc,700000/pci@1...
.
```

```

c8t0d0 <HITACHI-OPEN
/pci@1dc,600000/fibre-channel ...
.
c9t0d0 <HITACHI-OPEN
/pci@1fc,600000/fibre-channel ...

```

- c A comparison of the output of the previous two commands shows that board 15 slot 0 contains the controller c9.

- 3 As an alternative to using the `format` command, you can also use the following alternative procedure to determine which storage controllers are impacted by dynamic reconfiguration on a given slot or I/O board for e25K on Solaris 10.

- a Verify which IO controllers are impacted by dynamic reconfiguration on the board IO4 on sol10 (cougar) by using the following command:

```

cougar# cfgadm -s "cols=ap_id:physid" | grep IO4
IO4 /devices/pseudo/dr@0:IO4
IO4_C3V0 /devices/pci@9c,600000:IO4_C3V0
IO4_C3V1 /devices/pci@9d,600000:IO4_C3V1
IO4_C5V0 /devices/pci@9c,700000:IO4_C5V0
IO4_C5V1 /devices/pci@9d,700000:IO4_C5V1

```

The `-s` parameter is used to limit output to the `ap_id` and physical id columns.

- b Notice the `pci@...` In the phys id, use `grep` again using `pci@9[cd],[67]00000`:

```

cougar# cfgadm -s "cols=ap_id:physid" | grep pci@9[cd],[67]
IO4_C3V0 /devices/pci@9c,600000:IO4_C3V0
IO4_C3V1 /devices/pci@9d,600000:IO4_C3V1
IO4_C5V0 /devices/pci@9c,700000:IO4_C5V0
IO4_C5V1 /devices/pci@9d,700000:IO4_C5V1
c0 /devices/pci@9c,700000/pci@1/scsi@2:scsi
c1 /devices/pci@9c,700000/pci@1/scsi@2,1:scsi
c2 /devices/pci@9c,600000/SUNW,qlc@1,1/fp@0,0:fc
c3 /devices/pci@9c,600000/SUNW,qlc@1/fp@0,0:fc
c4 /devices/pci@9d,700000/SUNW,qlc@1/fp@0,0:fc
c5 /devices/pci@9d,700000/SUNW,qlc@1,1/fp@0,0:fc

```

c0 and c1 are located on IO4\_C5V0, c2 and c3 are on IO4\_C3V0, and c4 and c5 are on IO4\_C5V1

On sol 9, the procedure is almost the same:

```

jaguar# cfgadm -s "cols=ap_id:physid" | grep e17
e17 corresponds to the IO board #17
pcisch4:e17b1slot1 /devices/pci@23c,700000:e17b1slot1
pcisch5:e17b1slot0 /devices/pci@23c,600000:e17b1slot0
pcisch6:e17b1slot3 /devices/pci@23d,700000:e17b1slot3
pcisch7:e17b1slot2 /devices/pci@23d,600000:e17b1slot2
jaguar# cfgadm -s "cols=ap_id:physid" | grep pci@23[cd],[67]
c4 /devices/pci@23c,700000/pci@1/scsi@2:scsi

```

```
c5 /devices/pci@23c,700000/pci@1/scsi@2,1:scsi
c6 /devices/pci@23d,700000/SUNW,qlc@1/fp@0,0:fc
pcisch4:e17b1slot1 /devices/pci@23c,700000:e17b1slot1
pcisch5:e17b1slot0 /devices/pci@23c,600000:e17b1slot0
pcisch6:e17b1slot3 /devices/pci@23d,700000:e17b1slot3
pcisch7:e17b1slot2 /devices/pci@23d,600000:e17b1slot2
c4 and c5 are on e17b1slot1 and c6 is on slot3
```

### To determine the network interfaces on the board

- 1 Verify which network interfaces correspond to which slot on the I/O board (since each I/O board can carry upto four PCI cards) by using the `grep` command to match the `/etc/path_to_inst` for pci identifiers.

For e25K on Solaris 10

```
IO4_C3V0 /devices/pci@9c,600000:IO4_C3V0
IO4_C3V1 /devices/pci@9d,600000:IO4_C3V1
IO4_C5V0 /devices/pci@9c,700000:IO4_C5V0
IO4_C5V1 /devices/pci@9d,700000:IO4_C5V1
```

```
cougar# grep pci@9[cd],[67] /etc/path_to_inst | grep network
"/pci@9c,700000/network@3,1" 0 "eri"
"/pci@9c,700000/pci@1/network@0" 0 "ce"
"/pci@9c,700000/pci@1/network@1" 1 "ce"
"/pci@9d,600000/pci@1/network@0" 2 "ce"
```

IO4\_C5V0 contains eri0, c0, and c1. IO4\_C3V1 contains ce2.

cougar#

For e25K on Solaris 9:

```
pcisch4:e17b1slot1 /devices/pci@23c,700000:e17b1slot1
pcisch5:e17b1slot0 /devices/pci@23c,600000:e17b1slot0
pcisch6:e17b1slot3 /devices/pci@23d,700000:e17b1slot3
pcisch7:e17b1slot2 /devices/pci@23d,600000:e17b1slot2
jaguar#
```

```
jaguar# grep pci@23[cd],[67] /etc/path_to_inst | grep network
"/pci@23c,700000/pci@1/network@0" 2 "ce"
"/pci@23c,700000/pci@1/network@1" 3 "ce"
"/pci@23c,700000/network@3,1" 1 "eri"
"/pci@23d,600000/pci@1/network@0" 4 "ce"
```

e17b1slot1 contains ce2, ce3, and eri1. e17b1slot2 contains ce4.

### To disable the controllers on the board

- 1 Disable the active controllers on the I/O system card using the `vxdmpadm` command.

```
vxdmpadm disable ctlr=ctlr
```

For s6800:

```
vxdmpadm disable ctlr=c2
```

For e12K/15K:

```
vxddmpadm disable ctlr=c9
```

- 2 Using the vxddmpadm command, verify that controller is disabled. The output for all Sun servers (s6800 and e12K/15K/25K) will be similar except minor differences.

```
vxddmpadm listctlr all
```

For s6800:

| CTLR-NAME | ENCLR-TYPE | STATE    | ENCLR-NAME |
|-----------|------------|----------|------------|
| =====     |            |          |            |
| c0        | Disk       | ENABLED  | Disk       |
| c2        | HDS9960    | DISABLED | HDS99600   |
| c1        | HDS9960    | ENABLED  | HDS99600   |

In this example, the only controller on board is c2.

For e12K/15K:

| CTLR-NAME | ENCLR-TYPE | STATE    | ENCLR-NAME |
|-----------|------------|----------|------------|
| =====     |            |          |            |
| c0        | Disk       | ENABLED  | Disk       |
| c9        | HDS9960    | DISABLED | HDS99600   |
| c8        | HDS9960    | ENABLED  | HDS99600   |

In this example, the only controller on board IO15 is c9.

- 3 If a card has more than one controller, repeat this command for each controller on the card to be reconfigured.

#### To list the status of the private network links and to disable them

- 1 Enter the command lltstat -nv:

```
lltstat -nv
```

The output resembles:

For s6800:

LLT node information:

| Node        | State    | Links |
|-------------|----------|-------|
| * 0 wildcat | OPEN     | 2     |
| 1 leopard   | OPEN     | 2     |
| 2           | CONNWAIT | 0     |
| .           |          |       |
| .           |          |       |
| 31          | CONNWAIT | 0     |

The output shows that both domains have two links for private communication. Both links are "OPEN," that is, operational.

For e12K/15K:

LLT node information:

| Node        | State    | Links |
|-------------|----------|-------|
| 0 s6800f0   | OPEN     | 2     |
| * 1 leopard | OPEN     | 2     |
| 2           | CONNWAIT | 0     |
| .           |          |       |
| .           |          |       |
| 31          | CONNWAIT | 0     |

The output shows that both domains have two links for private communication. Both links are “OPEN,” that is, operational.

- 2 Display the `/etc/llttab` file on the wildcat domain.

```
cat /etc/llttab
```

For s6800:

```
set-node wildcat
set-cluster 13
link qfe4 /dev/qfe:4 - ether - -
link qfe0 /dev/qfe:0 - ether - -
```

The devices `qfe0` and `qfe4` are shown as the private network links.

For e12K/15K:

```
set-node leopard
set-cluster 13
link cd3 /dev/ce:3 - ether - -
link cd8 /dev/ce:8 - ether - -
```

The devices `ce3` and `ce8` are shown as the private network links.

- 3 Disable the private network link device.

For example for s6800, the private network link device is: `qfe4`, on I/O board N0.IB8.

```
/sbin/lltconfig -u qfe4
```

For example for e12K/15K, the private network link device is: `ce8`, on I/O board 15.

```
/sbin/lltconfig -u ce8
```

- 4 Check the status of the private network links:

```
llstat -nv
```

For s6800:

LLT node information:

| Node        | State    | Links |
|-------------|----------|-------|
| * 0 wildcat | OPEN     | 2     |
| leopard     | OPEN     | 1     |
| 2           | CONNWAIT | 0     |
| .           |          |       |
| .           |          |       |
| .           |          |       |
| 31          | CONNWAIT | 0     |

For e12K/15K:

LLT node information:

| Node        | State    | Links |
|-------------|----------|-------|
| 0 s6800f0   | OPEN     | 1     |
| * 1 leopard | OPEN     | 2     |
| 2           | CONNWAIT | 0     |
| .           |          |       |
| .           |          |       |
| .           |          |       |
| 31          | CONNWAIT | 0     |



**To list the status of the public NICs and to disable them**

- 1 Use the command `ifconfig -a`.

For s6800:

For example, `qfe3` (on board N0.IB6) and `qfe7` (on board N0.IB8), the NICs used for the public network connections, are operational.

**# ifconfig -a**

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232
index
 1 inet 127.0.0.1 netmask ff000000
ge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
index 2 inet 10.182.65.99 netmask fffff000 broadcast
 10.182.79.255 ether 0:3:ba:8:ec:40
qfe3:
flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,
 NOFAILOVER> mtu 1500 index 3 inet 10.182.66.143 netmask
 fffffff0 broadcast 10.255.255.255 groupname mn1 ether
 0:3:ba:8:ec:40
qfe7:
flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,
 NOFAILOVER> mtu 1500 index 4 inet 10.182.66.144 netmask
 fffffff0 broadcast 10.255.255.255 groupname mn1 ether
 0:3:ba:8:ec:40
```

For e12K/15K:

```
lo0: flags=1000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4> mtu 8232
index 1
 inet 127.0.0.1 netmask ff000000
ce4: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500
index 2
 inet 10.182.66.112 netmask fffff000 broadcast 10.182.79.255
 ether 0:0:be:a8:2c:5e
dman0: flags=1008843<UP,BROADCAST,RUNNING,MULTICAST,PRIVATE,IPv4
> mtu
 1500 index 3
 inet 10.2.1.6 netmask fffffffe0 broadcast 10.2.1.31
 ether 0:0:be:a8:2c:5e
ce0:
flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,
 NOFAILOVER> mtu 1500 index 5
 groupname mn1
 ether 0:0:be:a8:2c:5e
ce0:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500
 index 5
 inet 10.182.66.140 netmask fffff000 broadcast 10.182.79.255
ce5: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv
4,
 NOFAILOVER> mtu1500 index 6
 inet 0.0.0.0 netmask ff000000
 groupname mn1
 ether 0:0:be:a8:2c:5e
```

- 2
- For s6800: To disable the device `qfe7` on board `N0.IB8`, use the commands:  
`# ifconfig qfe7 down`  
`# ifconfig qfe7 unplumb`  
For e12K/15K: To disable the device `ce5` on board `IO15`, use the command:  
`# ifconfig ce5 down`
- 3
- For s6800: Use the `ifconfig -a` command to verify that `qfe7` is down:  
`# ifconfig -a`  
No information about `qfe7` should appear in the output.  
For e12K/15K/25K: Use the `ifconfig -a` command to verify that `ce5` is down:  
`# ifconfig -a`  
No information about `ce5` should appear in the output.

To disable and remove the IO board

- 1
- When the controllers and network interface cards are disabled, disconnect the board:  
For s6800:  
`# cfgadm -c disconnect N0.IB8`  
For e12K/15K:  
`# cfgadm -c disconnect IO15`  
The `-f` option is recommended only when normal disconnect attempt fails and there is no clear way to make the command succeed gracefully.
- 2
- Use the `cfgadm` command to check the status of the I/O board:  
`# cfgadm -al`  
For s6800: In the output, the fields `Receptacle`, `Occupant`, and `Condition` for `N0.IB8` show `disconnected`, `unconfigured`, and `unknown` respectively.  
The I/O board may be physically removed at this time. Before adding the new board to the wildcat domain, you must test it in another spare domain.  
For e12K/15K:
- | Ap_Id      | Type | Receptacle   | Occupant     |    |
|------------|------|--------------|--------------|----|
| Condition  |      |              |              |    |
| IO14       | HPCI | connected    | configured   | ok |
| IO14::pci0 | io   | connected    | configured   | ok |
| IO14::pci1 | io   | connected    | configured   | ok |
| IO14::pci2 | io   | connected    | configured   | ok |
| IO14::pci3 | io   | connected    | configured   | ok |
| IO15       | HPCI | disconnected | unconfigured |    |
| unknown    |      |              |              |    |
| SB14       | CPU  | connected    | configured   | ok |
| SB14::cpu0 | cpu  | connected    | configured   | ok |
| .          |      |              |              |    |
| .          |      |              |              |    |
- The I/O board, `IO15`, may be physically removed at this time.

### To add the new IO board

Use the following procedure to add the new I/O board. Make sure that the output of the `cfgadm` command shows the slot where the new board is to be added has the status `disconnected`, `unconfigured`, and `unknown`.

- 1 Physically add the board, connecting all necessary cables, and configure it:  
For s6800:  

```
cfgadm -c configure N0.IB8
```

  
For e12K/15K:  

```
cfgadm -c configure IO15
```
- 2 Run the `cfgadm -al` to verify the board has been configured; the board should be `connected`, `configured`, and `ok`.  
If you have stopped VCS, you may skip [step 3](#) through [step 6](#).
- 3 Reconfigure the network interface cards on the new board:  
For s6800:  

```
ifconfig qfe7 plumb
ifconfig qfe7 up
```

  
For e12K/15K:  

```
ifconfig ce5 plumb
```
- 4 Run the command `ifconfig -a` to verify that the NICs are up and running.
- 5 Reconfigure LLT to reestablish the private network links:  
For s6800:  

```
/sbin/lltconfig -t qfe4 -d /dev/qfe:4
```

  
For e12K/15K:  

```
/sbin/lltconfig -t ce8 -d /dev/ce:8
```
- 6 Verify the private network links are restored using the command `lltstat -nv`:  

```
/sbin/lltstat -nv
```
- 7 For s6800: Enable the controller `c2` on the `N0.IB8` using `vxddmpadm` command:  

```
vxddmpadm enable ctlr=c2
```

  
For e12K/15K: Enable the controller `c9` on the `IO15` using `vxddmpadm` command:  

```
vxddmpadm enable ctlr=c9
```
- 8 Verify the controller is up and running:  

```
vxddmpadm listctlr all
```

  
If you have stopped VCS before reconfiguring the I/O board, restart it. Refer to the section, [“Stopping and starting VCS”](#) on page 9.

## Performing dynamic reconfiguration on SunEnterprise 10K

The system board in a domain may contain I/O controllers, CPUs, or memory. Boards with I/O controllers can be dynamically reconfigured as long as you use VxVM with the Dynamic Multipathing (DMP) feature to manage the shared storage.

- See [“Detaching and attaching I/O system boards”](#) on page 36.
- See [“Detaching I/O system boards with DMP enabled”](#) on page 37.
- See [“Attaching I/O system boards with DMP enabled”](#) on page 38.
- See [“Detaching CPU/memory boards”](#) on page 39.
- See [“Attaching CPU/Memory boards”](#) on page 40.
- See [“Using VM without DMP enabled”](#) on page 41.

### Preparing environment for dynamic reconfiguration

Before performing dynamic reconfiguration operations on a domain, you must first set the appropriate environment variable.

#### To enable the kernel cage variable for dynamic reconfiguration

- 1 Using the Solaris 8 operating environment, you must set the `system(4)` variable, `kernel_cage_enable`, to 1 (enabled). By default, this variable is set to zero (kernel cage disabled), preventing dynamic reconfiguration Detach operations.
- 2 Edit the file `/etc/system` so that `kernel_cage_enable` equals 1.
 

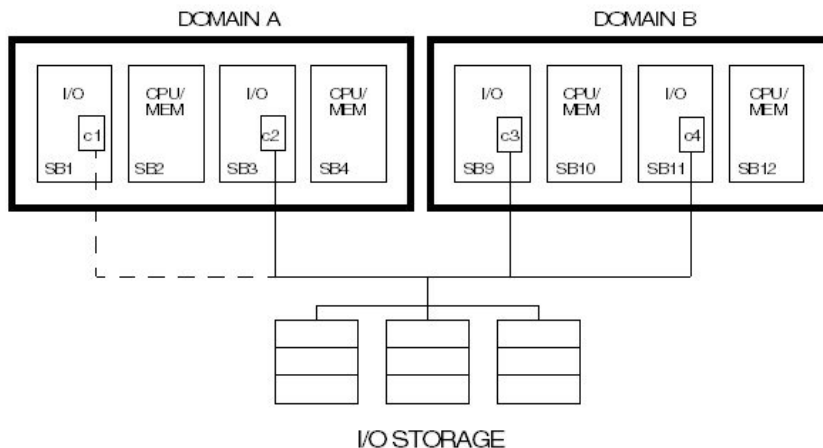
```
.
set kernel_cage_enable=1
.
```
- 3 Reboot the domain. To verify the kernel cage is enabled, check the file `/var/adm/messages`.
- 4 Look for the message:
 

```
NOTICE: DR Kernel Cage is ENABLED
```

### Detaching and attaching I/O system boards

In the configuration shown below, VCS runs on Domains A and B with service groups online on Domain A. Shared storage consists of a VxVM disk group with

Dynamic Multipathing (DMP) enabled. Dynamic Reconfiguration of I/O boards depends on DMP being configured for the storage.



In the example, the system board SB3, which has a disk controller, is to be removed, repaired, and replaced. The administrator disables the controller, and the disk controller on SB1 automatically takes over because of the DMP functionality. Using dynamic reconfiguration commands, the administrator can detach, or remove the board from the Domain A's configuration. When this is complete, the board can be physically removed.

Replacing the board—a controller board in this case—involves physically installing it and reconnecting it to the shared storage. Reconfiguring the board requires using dynamic reconfiguration commands to “attach” it to the domain, after which the controller can be re-enabled.

## Detaching I/O system boards with DMP enabled

Make sure the `kernel_cage_enable` variable is set.

See [“Preparing environment for dynamic reconfiguration”](#) on page 36.

You can detach system I/O boards using the following procedure:

### To attach an I/O board with DMP enabled

- 1 Freeze the VCS service groups running on the domain where you intend to perform dynamic reconfiguration operations. Freezing the service groups prevents them from being taken offline or failed over. Repeat the following command for each service group:  

```
hagrps -freeze ser_grp_name
```
- 2 Connect to the SSP server and log in to the domain whose system board requires Dynamic Reconfiguration.

```
ssp:D1% echo $SUNW_HOSTNAME
```

- 3 Enter the dr(1M) shell:

```
ssp:D1% dr
```

- 4 To verify the board is an I/O board, enter:

```
dr> drshow sb# IO
```

If the display lists the disks connected to the controller, the system board is an I/O board.

- 5 If the system board is an I/O board, open another window and log in as root to the domain you are currently reconfiguring.
- 6 Disable the controller on the I/O system board:

```
vxddmpadm disable ctrl=ctrl#
```

- 7 In the window where you are running dynamic reconfiguration, start detaching the I/O board by entering:

```
dr> drain sb#
```

- 8 Monitor the progress of the drain operation by entering:

```
dr> drshow sb# drain
```

- 9 When you see the message:

```
Percent Complete= 100% (0 KBytes remaining)
```

complete the detach operation:

```
dr> complete_detach sb#
```

- 10 To verify that the board is no longer configured, type the following command:

```
dr> drshow sb#
```

The detached board should not appear in the detailed listing.

- 11 Exit the dr shell:

```
dr> exit
```

- 12 If the board is not to be immediately replaced, unfreeze any frozen service groups:

```
hagrps -unfreeze ser_grp_name
```

Repeat for each service group.

## Attaching I/O system boards with DMP enabled

You can attach a system I/O board using the following procedure:

### To attach I/O system boards with DMP enabled

- 1 Freeze the VCS service groups running on the domain where you intend to attach a system board. Repeat the following command for each service group:

- # **hagrp -freeze ser\_grp\_name**
- 2 After physically replacing a previously removed I/O board, make sure it is connected to the shared storage.
- 3 From the SSP server, enter the `dr(1M)` shell:  
  
ssp:D1% **dr**
- 4 Follow the Sun procedure to attach the system board, described here briefly:  
dr> **init\_attach sb#**  
Complete the attach operation:  
dr> **complete\_attach sb#**
- 5 Verify that the dynamic reconfiguration attach operation has succeeded.  
Type:  
dr> **drshow #sb**  
The new system board should show in the list of configured boards.
- 6 Exit the `dr` shell.  
dr> **exit**
- 7 Log in as root to the domain where you are adding the system board. Enable the controller by entering:  
# **vxmpadm enable ctrlr=ctrlr#**
- 8 When you have successfully attached and enabled the system I/O board, unfreeze any frozen service groups:  
# **hagrp -unfreeze ser\_grp\_name**  
Repeat for each service group.
- 9 Verify that VCS is still up and running.

## Detaching CPU/memory boards

Use the following procedure if no I/O devices on the system board are used.

Make sure the `kernel_cage_enable` variable is set.

See [“Preparing environment for dynamic reconfiguration”](#) on page 36.

### To detach CPU/memory boards

- 1 Freeze the VCS service groups running on the domain where you intend to detach a CPU/Memory board. Freezing the service groups prevents them from being taken offline or failed over. Repeat the following command for each service group:  
# **hagrp -freeze ser\_grp\_name**
- 2 Connect to the SSP server and log in to the domain whose system board requires Dynamic Reconfiguration.  
ssp:D1% **echo \$SUNW\_HOSTNAME**

- 3 Enter the `dr(1M)` shell:  
`ssp:D1% dr`
- 4 In the window where you are running dynamic reconfiguration, start detaching the I/O board by entering:  
`dr> drain sb#`
- 5 Monitor the progress of the drain operation by entering:  
`dr> drshow sb# drain`
- 6 When you see the message  
 Percent Complete= 100% (0 KBytes remaining)  
 complete the detach operation:  
`dr> complete_detach sb#`
- 7 To verify that the board is no longer configured, type the following command:  
`dr> drshow sb#`  
 The detached board should not appear in the detailed listing.
- 8 Exit the `dr` shell:  
`dr > exit`
- 9 If the board is not to be immediately replaced, unfreeze any frozen service groups:  
`# hagrps -unfreeze ser_grp_name`
- 10 Repeat for each service group.

## Attaching CPU/Memory boards

Use the following procedure if none of the I/O devices on the system board are used.

### To attach a CPU/Memory board

- 1 Freeze the VCS service groups running on the domain where you intend to attach a system board. Repeat the following command for each service group:  
`# hagrps -freeze ser_grp_name`
- 2 Physically replace the CPU/Memory board.
- 3 From the SSP server, enter the `dr(1M)` shell:  
`ssp:D1% dr`
- 4 Follow the Sun procedure to attach the system board, described here briefly:  
`dr> init_attach sb#`  
 Complete the attach operation:  
`dr> complete_attach sb#`



- 5 Verify that the dynamic reconfiguration attach operation has succeeded.  
Type:  
`dr> drshow #sb`  
The new system board should show in the list of configured boards.
- 6 Exit the dr shell.  
`dr> exit`
- 7 When you have successfully attached the CPU/Memory board, unfreeze any frozen service groups:  
`# hagrps -unfreeze ser_grp_name`  
Repeat for each service group.
- 8 Verify that VCS is still up and running.

## Using VM without DMP enabled

If you have the Volume Manager DMP feature disabled for some or all of the disks in the shared storage, and you must perform dynamic reconfiguration operations within the cluster, we recommend using the VCS DiskReservation agent to guard against data corruption. In the event of a “split-brain” condition, that is, when two processors in a cluster can simultaneously write to the shared storage, the DiskReservation agent ensures that only one processor has access to the storage at one time. See the *VCS Bundled Agents Reference Guide* for information on configuring the DiskReservation agent.

