Sun™ Enterprise 10000 Dynamic Reconfiguration User Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

Part No. 816-3627-10
May 2002, Revision A

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

Sun Enterprise 10000 SSP Attributions:

Please
Recycle

Contents

Tables

Preface

This book describes the dynamic reconfiguration (DR) feature. DR enables you to logically attach boards to, and detach them from, dynamic system domains while the operating system continues to run.

Before You Read This Book

This book is intended for the systems administrator who has a working knowledge of UNIX® systems, particularly those based on the Solaris™ operating environment. If you do not have such knowledge, first read the Solaris user and system administrator books that were provided with your system, and consider UNIX system administration training.

How This Book Is Organized

This book contains the following chapters:

Chapter 1 introduces the DR feature.

Chapter 2 contains information and procedures for DR 3.0.

Appendix A lists error messages that are generated either by the SSP or on the domain.

Using UNIX Commands

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

Typographic Conventions

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

Shell Prompts

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

Related Documentation

Application	Title	Part Number
User information	*Sun Enterprise 10000 DR Configuration Guide*	816-3630
Command reference	*Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual*	806-7617
Installation instructions	*Sun Enterprise 10000 SSP 3.5 Installation Guide and Release Notes*	806-7615
Physically adding and removing system boards	*Sun Enterprise 10000 Systems Service Manual*	805-2817

Accessing Sun Documentation Online

A broad selection of Sun system documentation is located at:

`http://www.sun.com/products-n-solutions/hardware/docs`

A complete set of Solaris documentation and many other titles are located at:

`http://docs.sun.com`

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at:

`docfeedback@sun.com`

Please include the part number (816-3627-10) of your document in the subject line of your email.

CHAPTER 1

DR on the Sun Enterprise 10000 System

This chapter describes what dynamic reconfiguration (DR) is and what it can do for you. Then it describes the two models of DR that are available on the Sun Enterprise 10000 system.

Overview of Dynamic Reconfiguration

DR software is part of the Solaris operating environment. With the DR software you can dynamically reconfigure system boards to safely remove them or install them into a system. You perform DR operations while the Solaris operating environment is running, and with minimum disruption to user processes that are running in the dynamic system domain (referred to simply as a *domain* in this document).

With DR you can:

- Minimize the interruption of system applications while installing or removing a board
- Disable a failing device by removing it from the domain, before the failure can crash the operating system
- Display the operational status of boards in a domain
- Reconfigure a domain while the Solaris operating environment continues to run in the domain
- Hot-swap system boards without bringing the system down

If a system board is being used by a domain, 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 can attach it to the domain.

You can perform DR operations from the system service processor (SSP) by using the Automated DR (ADR) commands: `addboard`(1M), `moveboard`(1M), `deleteboard`(1M), and `showusage`(1M).

DR Concepts

This section contains descriptions of general DR concepts that pertain to the Sun Enterprise 10000.

Detachability

For a device to be detachable, it must conform to the following items:

- The device driver must support `DDI_DETACH`.
- Critical resources must be redundant or accessible through an alternate pathway. CPUs and memory banks can be redundant critical resources. Disk drives are examples of critical resources that can be accessible through an alternate pathway.

Some boards cannot be detached because their resources cannot be moved. For example, if a domain has only one board, that board cannot be detached. A board is not detachable if it controls the boot drive.

If there is no alternate pathway for a board, you can:

- Put the disk chain on a separate board. The secondary board can then be detached.
- Add a second path to the device through a second board so that the board can be detached without losing access to the secondary disk chain.

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

Quiescence

During the unconfigure operation on a system board with permanent memory (OpenBoot™ PROM or kernel memory), the operating environment is briefly paused, which is known as operating environment *quiescence*. All operating environment and device activity on the domain must cease during this critical phase of the unconfigure operation.

Before it can achieve quiescence, the operating environment must temporarily suspend all processes, CPUs, and device activities. If the operating environment cannot achieve quiescence, it displays the reasons, which can include devices that cannot be paused by the operating environment. The conditions that cause processes to fail to suspend are generally temporary.

Note – Execution threads and real-time processes do not affect quiescence.

Suspend-Safe and Suspend-Unsafe Devices

When DR suspends the operating environment, all of the device drivers that are attached to the operating environment must also be suspended. If a driver cannot be suspended (or subsequently resumed), the DR operation fails.

A *suspend-safe* device does not access memory or interrupt the system while the operating environment is in quiescence. A driver is suspend-safe if it supports operating environment quiescence (suspend / resume). A suspend-safe driver also guarantees that when a suspend request is successfully completed, the device that the driver manages will not attempt to access memory, even if the device is open when the suspend request is made. A *suspend-unsafe* device allows a memory access or a system interruption to occur while the operating environment is in quiescence.

DR 3.0 uses an *unsafe driver list* in the `ngdr.conf` file to prevent unsafe devices from accessing memory or interrupting the operating environment during a DR operation. The unsafe driver list is a property in the `ngdr.conf` file, with the following format:

```
unsupported-io-drivers="driver1","driver2","driver3";
```

DR reads this list when it prepares to suspend the operating environment so that it can unconfigure a memory component. If DR finds an active driver in the unsafe driver list, it aborts the DR operation and returns an error message. The message includes the identity of the active, unsafe driver. You must manually remove the usage of the device by performing one, or more, of the following tasks.

- Killing the processes using the device
- Unloading the driver by using the `modunload`(1M) command
- Depending on the device, disconnecting the cables.

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

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

DR Models

There are two models of DR available for the Sun Enterprise 10000 system. DR model 2.0 is sometimes referred to as “legacy DR,” and DR model 3.0 is referred to as “next generation DR.” Only DR 3.0 runs in a domain running version 9 of the Solaris operating environment. The following table shows the different versions of the Solaris operating environment and the SSP software that are used with DR models 2.0 and 3.0:

DR Model	Solaris Software Versions	SSP Software Versions
2.0	Solaris 5.1, 6, 7, and 8	3.3, 3.4, or 3.5
3.0	Solaris 8 10/01 and 02/02, Solaris 9	3.5 only

Only one model of DR can run within a domain at a time. To check the version of DR that is running, use the following command (available only with version 3.5 of the SSP software): `domain_status -m`. Make sure to verify the DR model before you execute any DR commands. The following is an example of the `domain_status`(1M) output. The `DR-MODEL` column indicates which model is enabled

```
# domain_status -m

DOMAIN     TYPE                    PLATFORM   DR-MODEL   OS   SYSBDS
A          Ultra-Enterprise-10000  all-A      2.0        5.8  2
B          Ultra-Enterprise-10000  all-A      3.0        5.8  3 4
C          Ultra-Enterprise-10000  all-A      2.0        5.7  5 6
D          Ultra-Enterprise-10000  all-A      3.0        5.9  7
```

According to this output, domain A is running Solaris version 8 software (OS 5.8) with DR model 2.0 enabled; domain B is running Solaris version 8 software with DR model 3.0 enabled; domain C is running Solaris version 7 software (OS 5.7) with DR model 2.0 enabled; and domain D is running Solaris version 9 (OS 5.9) with DR version 3.0 enabled..

Only certain commands are available in each model, and if you execute a command that is not supported, an error message appears on the console.

Caution – Before you switch to DR 3.0 in a domain that is running the Solaris 8 10/01 operating environment, you must upgrade the SSP software to version 3.5 because previous versions of SSP do not support DR 3.0 operations.

For more information about using DR 2.0, see the previous version of the *Sun Enterprise 10000 Dynamic Reconfiguration (DR) User Guide* (part number 806-7617-10). For more information about using DR 3.0, see the section "DR 3.0 Procedures" on page 9 of this book.

Enhancements in DR Model 3.0

The DR 3.0 model offers the following enhancements to DR 2.0:

- DR 3.0 has a framework that offers better integration with applications, through the reconfiguration coordination manager (RCM).
- DR 3.0 supports network multipathing using IPMP.

Where to Execute DR Commands

You execute DR operations from either of two places: from the system service processor (SSP) by using the SSP commands—`addboard`(1M), `moveboard`(1M), `deleteboard`(1M), `rcfgadm`(1M), and `showdevices`(1M); or from the domain, using the cfgadm(1M) command.

Requirements for Multipathing in DR 3.0

To use multipathing on DR model 3.0 domains, run IPMP (the IP multi-pathing software provided with the Solaris 8 operating environment) and MPxIO software, included in Solaris Kernal Update Patches 111412-02, 111413-02, 111095-02, 111096-02, and 111097-02.

CHAPTER 2

Using DR 3.0 Model

This chapter contains information about using DR model 3.0 on a Sun Enterprise 10000 system that is running version 3.5 of the SSP software and one of the following versions of the Solaris operating environment: Solaris 8 10/01, Solaris 8 02/02, or Solaris 9.

DR model 3.0 uses the domain configuration server, `dcs`(1M), to control DR operations. DR 3.0 includes Automated DR (ADR) commands such as `addboard`(1M), `deleteboard`(1M), and `moveboard`(1M). DR 3.0 also includes the following commands:

- `showdevices`(1M)—displays the usage of devices (see Section "Showing Device Information" on page 10 for more information).
- `rcfgadm`(1M)—displays the status of attachment points on the domain. (See also cfgadm_sbd(1M) for more information.)
- `cfgadm`(1M)—displays the status of attachment points on the domain. (See also `cfgadm_sbd`(1M) for more information.)

Note – For more information about using DR model 2.0, see the previous version of the *Sun Enterprise 10000 Dynamic Reconfiguration (DR) User Guide* (part number 806-7617-10).

Automatic DR

Automatic DR enables an application to perform DR operations without requiring user interaction. This ability is provided by an enhanced DR framework that includes the reconfiguration coordination manager (RCM) and a system event facility called `sysevent`. The RCM enables application-specific loadable modules to register with it for callbacks. The callbacks perform preparatory tasks before; error recovery during; and clean-up after a DR operation. The `sysevent` facility enables

applications to register for notification of system events. The automatic DR framework interfaces with the RCM and with `sysevent` to notify applications to give up resources automatically prior to unconfiguring them, and to capture new resources as they are configured into the domain.

Note – Automatic DR is a different feature than Automated DR (ADR)

For more information about RCM, refer to the *Solaris 8 System Administration Supplement* (part number 806-7502-10) in the Solaris 8 10/01 Update Collection.

Enhanced System Availability

The DR feature enables you to hot-swap system boards without bringing the system down. DR is used to unconfigure the resources on a faulty system board from a domain so that the system board can be removed from the system. The repaired (or replacement) board can be inserted into the domain while the Solaris operating environment is running. DR then configures the resources on the board into the domain.

DR and I/O Boards

You must use caution when you add or remove system boards with I/O devices. Before you can remove a board with I/O devices, all its devices must be closed, and all its file systems must be unmounted.

If you need to remove a board with I/O devices from a domain temporarily and then re-add it before any other boards with I/O devices are added, reconfiguration is unnecessary. In this case, device paths to the board devices remain unchanged. However, if you add another board with I/O devices after the first was removed, then re-add the first board, reconfiguration is required because the paths to the devices on the first board have changed.

Sun Enterprise 10000 Domains

The Sun Enterprise 10000 system can be divided into domains that contain system boards; and the components such as CPUs, memory chips, and CompactPCI cards that are connected to the boards. Each domain is electrically isolated into hardware partitions, which ensures that a hardware or software failure in one domain does not affect the other domains in the system.

DR 3.0 Procedures

This section contains procedures that describe how to use the DR 3.0 commands. The following procedures are included:

Showing Platform Information

Before you attempt to add, move, or delete a board to or from a specific domain, use the `domain_status`(1M) command to determine the domain name and board number.

▼ To Show Platform Information

1. **Use the** `domain_status`**(1M) command to obtain the domain information.**

```
% domain_status -m
```

Using the `domain_status` with the `-m` option command (in SSP version 3.5 only) displays the domain name, the DR model, and the number of the boards in the domain, as in the following example.

```
% domain_status -m

DOMAIN     TYPE                     PLATFORM   DR-MODEL   OS   SYSBDS
A          Ultra-Enterprise-10000   all-A      2.0        5.8  2
B          Ultra-Enterprise-10000   all-A      3.0        5.8  3 4
C          Ultra-Enterprise-10000   all-A      2.0        5.7  5 6
D          Ultra-Enterprise-10000   all-A      3.0        5.9  7
```

Showing Device Information

Before you attempt to perform any DR operation, use the `showdevices`(1M) command to display the device information, especially when removing devices.

▼ To Show Device Information

1. **Use the** showdevices**(1M) command to display the device information for a domain.**

```
% showdevices -v -d A
```

The above command displays the device information for all of the CPUs in domain A. Refer to the showdevices(1M) man page to learn how to display device-specific information.

```
CPU
---
domain    board  id     state    speed  ecache  usage
A         SB10   40     online   400    4
A         SB10   41     online   400    4
A         SB10   42     online   400    4
A         SB10   43     online   400    4
A         SB14   56     online   400    4
A         SB14   57     online   400    4
A         SB14   58     online   400    4
A         SB14   59     online   400    4
```

The following output is an example of the memory output for the showdevices(1M) command.

```
Memory
drain in progress:
-----------------
                board  perm    base          domain  target deleted remaining
domain   board  memMB  memMB  address      memMB   board  memMB   memMB
A        SB10   2048   933    0x800000000  4096    SB14   512     1536
A        SB14   2048   0      0x400000000  4096
```

The following is an example of the I/O devices output for the showdevices(1M) command.

```
IO Devices
----------
domain    board  device     resource            usage
A         SB14   sd0
A         SB14   sd1
A         SB14   sd2
A         SB14   sd3        /dev/dsk/c0t3d0s0   mounted filesystem "/"
A         SB14   sd3        /dev/dsk/c0t3d0s1   dump device (swap)
A         SB14   sd3        /dev/dsk/c0t3d0s1   swap area
A         SB14   sd3        /dev/dsk/c0t3d0s3   mounted filesystem "/var"
A         SB14   sd3        /var/run            mounted filesystem "/var/run"
A         SB14   sd4
A         SB14   sd5
```

Refer to the showdevices(1M) man page for a complete list of the options and arguments for this command.

Adding Boards

Adding a board to a domain moves the board through several state changes. First the board is connected to the domain, and then it is configured into the Solaris operating environment. After the board is connected, it is considered to be part of the physical domain and available for use by the operating system.

▼ To Add a Board to a Domain

1. **Use the** addboard**(1M) command to add the board to the domain.**

   The following example shows how the addboard(1M) command adds system board 2 to the domain specified by *domain_id*. Two retries are performed, if necessary, with a wait time of 10 minutes (600 seconds) between retries.

```
% addboard -d domain_id -r 2 -t 600 SB2
```

Deleting Boards

Deleting a board from a domain removes the board from the domain.

Always check the usage of the components on a board before you delete it from a domain. If the board hosts permanent memory, the memory is moved to another board within the same domain before the board is deleted from the domain. Likewise, if any busy devices are present, you must wait or ensure that the device is no longer being used by the system before you attempt to remove the board.

Caution – You must use the `power` command to power off the board before physically removing it from the system. The `deleteboard`(1M) command does not power off the board. Refer to the `power`(1M) man page for information about the `power` command. Also see the section “To Physically Replace a System Board” on page 14. In addition, the *Sun Enterprise 10000 Systems Service Manual* contains complete information about physically removing and replacing boards.

▼ To Delete a Board From a Domain

1. **Use the `deleteboard`(1M) command to delete the board from the domain.**

   The following example of the `deleteboard`(1M) command deletes system board 2 from its current domain. Two retries are performed, if necessary, with a wait time of 15 minutes (900 seconds) between retries.

```
% deleteboard -r 2 -t 900 SB2
```

Moving Boards

Moving a board from one domain to another domain involves removing the board from the first domain; and then connecting and configuring it into the target domain.

Always check memory usage on a board, and the devices that are connected to it, before moving it out of a domain. If the board hosts permanent memory, the memory must be moved to another board within the same domain before the board can be moved to another domain. Likewise, if a busy device is present, you must wait until the device is no longer being used by the system before you attempt to move the board.

▼ To Move a Board

1. **Use the** `moveboard`**(1M) command to move the board from one domain to another domain.**

   The following example of the `moveboard`(1M) command moves system board 2 from its current domain to the domain specified by *domain_id*. Two retries are performed, if necessary, with a wait time of 15 minutes (900 seconds) between retries.

```
% moveboard -d domain_id -r 2 -t 900 SB2
```

Replacing System Boards

This section describes how to physically replace a board in a domain by using the commands described in this chapter.

▼ To Physically Replace a System Board

In the following steps, system board 2 is removed from its current domain and replaced by system board 3. Two retries are performed, if necessary, with a wait time of 15 minutes (900 seconds) between retries.

1. **Delete the board from the domain.**

```
% deleteboard -r 2 -t 900 SB2
```

2. **Power off system board 2.**

   Refer to the `power`(1M) man page for information about the `power` command.

```
% power -off -sb 2
```

Caution – For complete information about physically removing and replacing boards, refer to the *Sun Enterprise 10000 Systems Service Manual*. Failure to follow the procedures described therein can cause damage to system boards and other components.

3. **Physically remove system board 2 and replace it with system board 3.**

4. **Power on system board 3.**

```
% power -on -sb 3
```

5. **Add system board 3 to the domain.**

```
% addboard -r 2 -t 900 SB3
```

APPENDIX **A**

DR Error Messages

This appendix contains a list of error messages that you might see while performing DR operations. The list does not include Protocol Independent Module (PIM) layer errors, which are more generic than the error messages in the tables that follow.

All DR error messages are sent to the one or both of the following locations:

- SSP applications
- System error logs

Searching This Appendix

Before you use this appendix, be sure to read the following list of search tips.

- Search on a specific string of text in the error message.
- Avoid using numeric values. They are treated as replaceable text in this appendix.
- In this appendix, italicized strings such as the following are used to represent replaceable text in the error messages: *error_description*, *domain_name*, *path*, *line_number*, *operation*, *retry_value*, and *attachment_point*.
- The tables are presented by the type of error or failure.

Error-Type Links

The following are different types of errors:

SSP Errors

See the section “SSP Error Messages” on page 18.

Domain Errors

Use one of the following links to start a search of domain-related error messages:

“DCS Error Messages” on page 21

“DR Driver Error Messages” on page 24

“Plugin Error Messages” on page 30

SSP Error Messages

The following are SSP-related error messages:

TABLE 2-1 SSP-Related Error Messages

Error Message	Probable Cause	Suggested Action(s)
`Domain` *`domain_name`* `has unknown DR model`	SSP failed to determine which DR model is running on domain. Possible causes for this error message are: 1. The domain is down, or hung, or too busy to respond to the request from SSP 2. The SSP-to-domain link is down 3. DCS is not running 4. The DR driver is not loaded on domain.	1. Make sure the domain is up and running 2. Check the link between SSP and the domain 3. Retry at a later time when the domain is not as busy 4. Make sure DCS is running on the domain 5. Make sure the DR driver is properly loaded.
`Board` *`xx`* `is in the intermediate state`	A failed DR operation left the board in an intermediate state.	Re-run the command at a later time. If the error persist, check the domain message file for the root cause. Some DR operations are not allowed due to certain constraints. If the issue cannot be resolved, run the opposite DR command to restore the board to its original state.

TABLE 2-1 SSP-Related Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action(s)
`Board xx is in intermediate attachment state`	Another board *xx* in the target domain is in an intermediate attachment state and has to be fully attached to the domain or fully detached from the domain before the current DR operation can proceed.	Use addboard(1M) to attach board *xx* to the target domain or deleteboard(1M) to detach it from the target domain, and re-run the current command.
`Failed in complete attachment stage for board xx`	The board could not be connected and configured into the domain.	Check the domain message file for detailed errors; the file may indicate the cause of the failure. Resolve the issue and re-run the command.
`Board xx is not a member of a domain`	The SSP software shows that the specified board does not belong to any domain.	Make sure the right board number is specified; and use domain_status(1M) to to find out whether the board belongs to any domain.
`Board xx is in intermediate detachment state`	Another board *xx* in the target domain is in an intermediate detachment state and must be fully attached to the domain, or fully detached from the domain, before the current DR operation can proceed.	Use addboard(1M) to attach board *xx* to the target domain, or deleteboard(1M) to detach it from the target domain. Then re-run the current command.
`Failed in complete detachment stage for board xx`	The board could not be unconfigured and disconnected from the domain.	Check the domain message file for detailed errors; it may indicate the cause of the failure. Resolve the issue and re-run the command. If the board cannot be detached due to certain restraints, run addboard(1M) to return the board to its original state.
`Unable to connect to SNMP agent`	The DR thread failed to establish a connection with snmpd, which may be down or too busy.	• Make sure snmpd is up and running. • Make sure there are not too many jobs running at the same time, which requires snmpd attention. • Reboot the SSP machine if none of the previous actions succeed.

TABLE 2-1 SSP-Related Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action(s)
RDR_ERROR	There was an error communicating with the dcs process.	Make sure the internal network for the domain is working properly. The SSP should be working and you should be able to ping the domain.
RDR_NET_ERR	There was an error setting up a socket connection with the dcs process.	Make sure the internal network for the domain is working properly. The SSP should be working and you should be able to ping the domain.
RDR_TIMEOUT	A poll() system call timed out due to a loss of communication to the dcs; or the domain is busy.	Verify the network connection or retry the DR operation.
RDR_ABORTED	A poll(), read(), or write() system call was interrupted.	Retry the DR operation.
RDR_MSG_INVAL	A message being sent to, or received from, dcs is invalid.	No user intervention is needed or possible.
RDR_MEM_ALLO	Unable to allocate memory.	No user intervention is needed or possible.

Domain Error Messages

DCS Error Messages

The following table contains DCS error messages that are sent to the console window, to the `/var/adm/messages` directory, and to the `$SSPLOGGER/`*`domain_name`*`/messages` directory.

TABLE 2-2 DCS Error Messages

Error Message	Probable Cause	Suggested Action
`dcs: permission denied`	ERROR: Only the superuser of the domain can run the DCS.	Check the `inetd.conf` file on the domain to ensure that the DCS is started with superuser ID.
`dcs: internal error: `*`operation`*`: `*`error_description`*	ERROR: An internal error occurred within the DCS.	Use the *error_description*, which corresponds with the *errno_value*, to diagnose the error. The operation field refers to the function call that caused the error.
`dcs: unrecognized error reported`	NOTICE: The DCS reported an unknown error condition.	Use the log file on the domain to help determine what caused the error.
`dcs: network initialization failed`	ERROR: The DCS failed to initialize the network connection used to accept DR requests from the SSP.	Retry the DR operation.

TABLE 2-2 DCS Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`dcs: failed to acquire reserved port`	ERROR: The DCS uses port 665, which is reserved through sun-dr. The error occurred because another process is using the port.	Determine if another process is still using the port. If so, kill the process, if possible, then retry the DR operation.
`dcs: connection attempt failed`	ERROR: The DCS failed to establish a connection with the SSP.	Retry the DR operation.
`dcs: unable to receive message`	ERROR: The DCS failed to receive a message from the SSP.	Retry the DR operation.
`dcs: unable to send message for` *`operation_name`* *`operation`*	ERROR: The DCS failed to send a message to the SSP.	Retry the DR operation.
`dcs: sun-dr service not found, using reserved port 665`	ERROR: The DCS failed to find the sun-dr service in `/etc/services`.	None
`DCS NOTICE: client disconnected`	NOTICE: The client unexpectedly disconnected.	None
`dcs: unknown operation requested`	ERROR: The SSP requested an operation that is not recognized by the DCS.	Retry the DR operation.
`dcs: operation failed`	ERROR: The current DCS operation failed to complete. The DR operation can still succeed, if the DCS failed only to send the results to the SSP.	Check the status of the operation manually. If the DR operation did not succeed, retry the operation.
`dcs: invalid session establishment sequence`	ERROR: The session establishment sequence and (the initialization handshake) between the SSP and the DCS failed.	Retry the DR operation.
`dcs:` *`operation_name`* `operation issued before session established`	ERROR: A DR operation was requested before the session was established.	Retry the DR operation.

TABLE 2-2 DCS Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`dcs: received an invalid message`	ERROR: The DCS received unexpected information in the message.	Retry the DR operation.
`dcs: confirm callback failed, aborting operation`	ERROR: The DCS was unable to display the confirmation prompt to the user.	None
`dcs: message callback failed, continuing`	NOTICE: The DCS was unable to display a message to the user.	None
`dcs: retry value invalid (`*`retry_value`*`)`	NOTICE: The value given for the *retry_value* was invalid, so the operation proceeded with the retry value set to zero.	None
`dcs: timeout value invalid (`*`timeout_value`*`)`	NOTICE: The value given for the *timeout_value* was invalid, so the operation proceeded with the retry value set to zero.	None
`dcs: retrying operation, attempt` *`attempt_number`*	INFO: The DCS is retrying the operation. The *attempt_number* field represents the current attempt.	None
`dcs: failed to start a new session handler`	ERROR: The DCS failed to start a concurrent session handler to process the incoming DR request.	Retry the DR operation.
`dcs: abort attempt of session,` *`session_id`*`, unsuccessful`	ERROR: The DCS failed to abort session, *session_id*.	Retry the abort request.
`dcs: unsupported message protocol version:` *`version_number`*	ERROR: The DCS does not support the reported protocol version, *version_number*.	Check the DR software on the domain and the SSP. Reinstall the proper version of the software on the domain if they are not compatible.

TABLE 2-2 DCS Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`dcs: session aborted`	INFO: The current DR operation was aborted by the user.	None
`dcs: illegal option` *`option`*`, exiting`	ERROR: The DCS was passed an illegal option name.	Check the `inetd.conf` file on the domain and remove the illegal option from the entries for the DCS.
`dcs: illegal argument to` *`option`* `flag (`*`argument`*`),` *`action`*	NOTICE: The option *option* was given the illegal argument *argument*. The DCS will perform the action specified by *action*.	Check the `inetd.conf` file on the domain and fix the entries for the DCS.
`dcs: resource info init error (`*`error_code`*`)`	ERROR: The DCS failed to initialize the module responsible for providing resource usage information.	Retry the operation.

DR Driver Error Messages

The following table contains DR driver error messages that are sent to the console window, to the `/var/adm/messages` directory, and to the `$SSPLOGGER/`*`domain_name`*`/messages` directory.

TABLE 2-3 DR Driver Error Messages

Error Message	Probable Cause	Suggested Action
`ngdr: Internal error: dr.c` *`line_number`*	An internal error has occurred in the DR driver.	Retry the operation that failed. If the error persists, exit and restart various DR software components, then retry the operation. If the problem still persists, reboot the domain. Check the console or the system logs for additional information.

TABLE 2-3 DR Driver Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`ngdr: Insufficient memory: resource`	The DR framework was unable to configure or unconfigure resources because a KPHYSM_ERESOURCE error or `cpu_configure()`/`cpu_unconfigure()` error with the ENOMEM *errno* occurred.	This condition might be transient. Retry the DR operation. If the error persists and if the operation that is failing is the unconfigure operation, then try configuring more memory into the domain from a different domain. If the error still persists, reboot the domain.
`ngdr: Device busy: resource`	Translation of possible EBUSY errno messages from `cpu_configure()` or `cpu_unconfigure()`; or an I/O device cannot be detached because it is busy. This error message is also returned if a CPU to be detached is online when `dr_pre_detach_cpu` is called. A CPU cannot be detached while a memory drain is in progress.	Use `showdevices`(1M) on the system controller to find out why the resource is busy. Or, on the domain, use `fuser`(1M), `psrinfo`(1M), `prtdiag`(1M), or similar tools to find out why the device is busy. Also check whether another memory drain is already in progress. Either reconfigure or shut down whatever is consuming the resource, or wait for the previous memory drain to complete, depending on the cause of the error. Then retry the DR operation.
`ngdr: Operation already in progress: resource`	Translation of possible EALREADY errno from `cpu_configure()` or `cpu_unconfigure()`.	Use `showdevices`(1M) on the system controller to examine the configuration of the specified resource. Or, on the domain, use `cfgadm`(1M), `pbind`(1M), `psrinfo`(1M), and similar commands to examine the configuration of the resource. Determine what operations are already in progress on this resource, and either wait for them to complete or cancel them. Then, retry the DR operation. The operation already in progress may already have terminated, so retrying to the operation might succeed, or may produce another error.
`ngdr: I/O error: resource`	An unexpected error code resulted from a call to `kphysm_del_start`. A more verbose `cmn_err message` is also printed.	Check the verbose error message from `cmn_err` in the system logs, and/or on the console for a more specific condition and suggested action.
`ngdr: Bad address: resource`	`kphysm_add_memory_dynamic` returned `KPHYSM_EFAULT`.	Retry the DR operation. If this error persists, contact your Sun Service representative.

TABLE 2-3 DR Driver Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`ngdr: No device(s) on board: board_path`	The board is connected or disconnected with no devices (I/O, memory, or CPU).	If devices were expected to be on the board, then disconnect the board and remove it from the system. The board's components should be reseated by a qualified technician.
`ngdr: Invalid argument: attachment_point`	DR was passed an invalid argument.	Retry the DR operation. If this error persists, contact your Sun service representative.
`ngdr: Invalid state transition: attachment_point`	A DR operation was sequenced out of order. This could be operator error if the `cfgadm`(1M) commands were issued out of order. Or, the DR driver could be confused due to some internal error conditions.	Retry the DR operation. If this error persists, stop and restart (or unload and load) DR software components to recover from this error condition. If the error persists, reboot the domain.
`ngdr: Device in fatal state`	The device could not be suspended, or it refused to be suspended.	Retry the DR operation. If this error persists, the device could be in suspend-unsafe mode. Check the list of suspend-unsafe devices. If the device is unsafe, use `showdevices`(1M) or `fuser`(1M) to show whether the device is in use, and manually reconfigure the resource. Then, manually unload the driver, or if needed, disconnect the cables attached to the device. The device should now be safe to retry the operation. Do not reconnect the cables to the device, reload its driver, or reconfigure its resources before the DR operation has completed successfully.
`ngdr: Device failed to resume: path`	A previously suspended device could not be resumed.	
`ngdr: Cannot stop user thread`	DR could not stop a user thread(s) in preparing a device to be suspended.	Retry the DR operation. If this error persists, examine the user threads that failed to suspend, and determine why they could not be suspended. You might have to kill the threads to enable the DR operation to proceed.
`ngdr: Cannot stop kernel thread: name`	DR could not stop a kernel thread.	Retry the DR operation. If this error persists, examine the user threads that failed to suspend, and determine why they could not be suspended. Kill the kernel threads, if possible, to enable the DR operation to proceed.

TABLE 2-3 DR Driver Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`ngdr: Failed to off-line: cpu`	A CPU could not be brought off-line, which prevents it from being unconfigured. The CPU might have a thread(s) bound to it. An additional `cmn_err` message is logged if there are threads bound to the CPU. DR must be able to off-line CPUs and/or to power off CPUs before the board can be disconnected.	Check the console and system log messages to determine if threads are bound to the CPU. If they are, they can be manually unbound or rebound to CPUs on other boards in the domain. If threads are not bound to the CPU, use `psrset`(1M), `pbind`(1M), and `psrinfo`(1M) to determine what changes are required to enable DR to off-line the CPU. For example, you might have to add more CPUs to the domain from different boards. Or, you may have to online other CPUs. Finally, you might have to add more CPU boards to take over the CPU workload.
`ngdr: Failed to on-line: cpu`	DR could not online a CPU on a newly-connected or previously unconfigured board.	
`ngdr: Failed to start CPU: cpu`	DR could not start a CPU on a newly-connected or previously unconfigured board.	
`ngdr: Failed to stop CPU: cpu`	DR could not power off a CPU on a board to be unconfigured. All of the CPUs on a board to be unconfigured must be taken offline and powered off before the operation can succeed.	
`ngdr: Kernel cage is disabled: resource`	When the kernel cage is disabled, boards hosting permanent memory cannot be detached.	Enable the kernel cage in `/etc/system` and reboot the domain.
`ngdr: No available memory target: resource`	DR could not detach the board because it hosts permanent memory and there is no available target for the memory. Permanent memory must be moved to another memory component within the domain before the DR operation can succeed.	Configure an additional memory component that contains an adequate amount of memory to act as a target for this board. Then, retry the DR operation.
`ngdr: VM viability test failed: resource`	Translation of error code returned by `kphysm_del_start`.	Configure additional memory components into the domain to relieve memory resource pressure. Then, retry the DR operation.

TABLE 2-3 DR Driver Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`ngdr: Memory operation refused: resource`	Translation of error code returned by `kphysm_del_start`.	Configure additional memory components into the domain to relieve memory resource pressure. Then, retry the DR operation.
`ngdr: Non-relocatable pages in span: resource`		
`ngdr: Memory operation cancelled: resource`		
`ngdr: Memory operation failed: resource`	DR failed to attach the memory on a newly attached board.	
`ngdr: Can't unconfig cpu if mem online`	DR cannot unconfigure a CPU if the memory on the board is online.	You must bring memory offline before you can unconfigure the board.
`ngdrmach: Cannot read property value: Device Node node_address: property property_name`	DR could not get the specified property of a particular device node.	
`ngdrmach: Cannot determine property length: board::slot:property`	DR could not get the length of the specified property for a particular device node.	
`ngdrmach: No CPU specified for connect: slot`	A board connect operation requires that a CPU on the board to connect is specified as part of the DR request.	Retry the addboard operation.
`ngdrmach: Cannot move SIGB assignment`	No CPU could be found as a target for relocating the boot proc of the domain.	Make sure there are CPUs present and online on another board in the domain.
`ngdrmach: Cannot disconnect CPU; SIGB is currently assigned: slot::board`	The disconnect operation is attempting to remove the board that has the boot proc.	Retry the deleteboard operation. It may be necessary to run addboard before retrying deleteboard.
`ngdrmach: Device driver failure: path`	Operations to online or offline a device failed.	Retry the operation.
`ngdrmach: Must specify a CPU on the given board: cpu_id`	A board connect operation requires that a CPU on the board to connect is specified as part of the DR request.	Retry the addboard operation.
`ngdrmach: No such device: board::slot`	The specified device does not exist on the specified board.	

TABLE 2-3 DR Driver Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`ngdrmach: Memory configured with inter-board interleaving: board::slot`	Memory that is interleaved across boards cannot be unconfigured from the system.	Configure the system without memory interleaving.
`ngdrmach: Invalid board number: board_number`	An invalid board number was specified for the connect board operation.	Retry the deleteboard operation.
`ngdrmach:: Cannot proceed; Board is configured or busy: component_name`	DR cannot disconnect a board that is configured or busy.	Unconfigure the board, or wait for any previous DR operations on the board to complete. Then, retry the DR operation
`ngdrmach: Firmware probe failed: attachment_point`	OBP failed to probe the board during board disconnect.	Retry the deleteboard operation. It may be necessary to run addboard before retrying the deleteboard
`ngdrmach: Firmware deprobe failed: attachment_point`	OBP failed to deprobe the board.	
`ngdrmach: Operation not supported`	The operation you attempted is not supported.	None
`ngdrmach: Unrecognized platform command: command/options`	An unrecognized command was passed to DR.	Refer to the `cfgadm_sbd`(1M) man page to ensure that you use a valid argument. If you used a valid argument and this error persists, contact your Sun service representative.
`ngdrmach: drmach parameter is not a valid ID`	An invalid `drmachid_t` value was encountered.	
`ngdrmach: drmach parameter is inappropriate for operation`	The wrong type of `drmachid_t` was passed to a function.	
`ngdrmach: Unexpected internal condition: drmach.c line_number`	An internal `drmach` error occurred.	Use `modunload`(1M) and `modload`(1M) to unload then to load the drmach driver. Then, retry the DR operation. If this error persists, then you must reboot the domain.
`ngdrmach: Firmware move_cpu0 failed: CPU cpu_id`	OBP failed to move the boot proc during the unconfigure operation.	Retry the deleteboard operation. It may be necessary to run addboard before retrying the deleteboard.

Plugin Error Messages

The following error messages are generated by the `libcfgadm` system board plugin. The messages are sent to the `netcon`(1M) console window, to the `/var/adm/messages` directory, and to the `$SSPLOGGER/`*`domain_name`*`/messages` directory.

TABLE 2-4 Plugin Error Messages

Error Message	Probable Cause	Suggested Action
`Configuration operation cancelled: command` *`ap_id`*	You did not confirm a configuration operation that requires confirmation.	See the `cfgadm`(1M) and/or the `cfgadm_sbd`(1M) man page for more information about which configuration operations require confirmation.
`Hardware specific failure:` *`command ap_id`*`:` *`error`*`:` *`resource`*	A system error occurred during the execution of the command. The error message, *error*, can be a standard error, or it can be a more specific error message that is returned by the DR driver. (See the DR Driver error messages for more information.) The name of the resource, *resource*, that is causing the error (for example, a busy device) can also be returned by the DR driver.	For busy devices, identify and stop usage of the device. For other errors, refer to the driver's documentation for recovery options.
`Library Error: command invalid:` *`command`*	The specified command is invalid for system boards.	Refer to the `cfgadm_sbd`(1M) man page for a list of valid commands.
`Library Error: command not supported: command` *`ap_id`*	The command that was executed is not supported for the attachment point specified by ap_id. For example, the `connect` operation is not allowed for CPUs.	Refer to the `cfgadm_sbd`(1M) man page for a list of supported commands.
`Library Error: command aborted:` *`command`*	You aborted the command.	N/A
`Library Error: option invalid:` *`option`*	The specified option, *option*, is invalid.	Refer to the `cfgadm_sbd`(1M) man page for a list of the valid options.
`Library Error: option requires value:` *`option`*	The specified option, *option*, requires a value.	Refer to the `cfgadm_sbd`(1M) man page for a list of the option values.
`Library Error: option requires no value:` *`option`*	The specified option, *option*, does not require a value.	Refer to the `cfgadm_sbd`(1M) man page for a list of options that do not require values.
`Library Error: option value invalid:` *`option value`*	The specified value, *value*, for the option, *option*, is invalid.	Refer to the `cfgadm_sbd`(1M) man page for a list of valid option values.

TABLE 2-4 Plugin Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`Library Error: attachment point invalid: ap_id`	The specified attachment point, *ap_id*, could not be parsed correctly. This error is rare and could indicate an internal error.	Refer to the `cfgadm_sbd`(1M) man page for a list of valid attachment points. If this error persists, contact your Sun service representative.
`Library Error: component invalid: ap_id`	The specified component, *ap_id*, is invalid.	Refer to the `cfgadm_sbd`(1M) man page for a list of valid dynamic attachment points.
`Library Error: sequence invalid: command (rstate ostate) ap_id`	The specified command, *command*, is invalid for the receptacle and / or occupant state of the specified attachment point. For example, trying to connect an empty slot results in an invalid sequence error.	Refer to the `cfgadm_sbd`(1M) man page for a list of valid operations.
`Library Error: offline ap_id (path): error`	The Reconfiguration Coordination Manager (RCM) failed to take the resource, *ap_id*, offline. The error message, *error*, returned by the RCM will indicate the reason for the failure. Usually, the reason is a busy device.	For busy devices, identify and remove the usage of the device.
`Library Error: suspend ap_id (path): error`	The Reconfiguration Coordination Manager (RCM) failed to suspend the resource, *ap_id*. The error message, *error*, returned by the RCM will indicate the reason for the failure. Usually, the reason is a busy device.	For busy devices, identify and remove the usage of the device.
`Library Error: not enough memory`	The plugin operation failed due to a lack of memory.	Check the memory usage.
`Library Error: change signal disposition failed`	The plugin operation failed to set up signals before it started the DR operation.	None
`Library Error: cannot get RCM handle`	The Reconfiguration Coordination Manager (RCM) failed to initialize.	None
`Library Error: cannot open library: error`	The Reconfiguration Coordination Manager (RCM) library, *library*, was found, but an error occurred when it was opened. The error message, *error*, is returned by `dlopen`(3DL).	Check for proper installation of the RCM.

TABLE 2-4 Plugin Error Messages *(Continued)*

Error Message	Probable Cause	Suggested Action
`Library Error: cannot find symbol` *`symbol`* `in` *`library`*	A required symbol, *symbol*, was not found in the Reconfiguration Coordination Manager (RCM) library, *library*.	Check for proper installation of the RCM.
`Library Error: cannot stat` *`library`*`:` *`error`*	The Reconfiguration Coordination Manager (RCM) library, *library*, exists, but the `stat`(2) function failed to get the file status. The error message, *error*, will be returned by the Solaris operating environment.	None

Index

M

N

Q

R

S

U