![Sun microsystems logo]

Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual

Adobe PostScript™

Please Recycle

Contents

Maintenance Commands

NAME   Intro – Sun Enterprise 10000 DR administration

DESCRIPTION   This manual contains the commands, scripts, and programs executed in the
Sun Enterprise 10000 Dynamic Reconfiguration (DR) environment only.

Note - Execute the commands contained in this manual only in the SSP
environment.

**LIST OF
COMMANDS**   abort_attach**(1M)**

abort DR attach operation

abort_detach**(1M)**

abort DR detach operation

addboard**(1M)**

add a system board to a domain

complete_attach**(1M)**

complete DR attach operation

complete_detach**(1M)**

complete DR detach operation

deleteboard**(1M)**

delete a system board from a domain

dr**(1M)**

initiate dynamic reconfiguration shell

dr.service**(1M)**

abort DR attach system board operation

dr_cmd_a_attach**(1M)**

abort DR attach system board operation

dr_cmd_a_detach**(1M)**

abort DR detach system board operation

dr_cmd_auto_config**(1M)**

run Solaris reconfig sequence on target domain

dr_cmd_c_attach**(1M)**

complete DR attach system board operation

dr_cmd_c_detach**(1M)**

complete DR detach system board operation

dr_cmd_c_f_detach**(1M)**

force completion of DR detach system board operation

dr_cmd_cpu_info**(1M)**

show processors on a system board in Tcl encoding

dr_cmd_debug**(1M)**

toggle DR library-level debugging

dr_cmd_detach_allow**(1M)**

verify a system board can support DR detach

dr_cmd_dev_info**(1M)**

show devices on a system board in Tcl encoding

dr_cmd_drain**(1M)**

start memory drain on a system board

dr_cmd_drain_status**(1M)**

show state of in-progress memory drain

dr_cmd_eligible_attach**(1M)**

verify a system board is eligible for DR attach

dr_cmd_eligible_detach**(1M)**

verify a system board is eligible for DR detach

dr_cmd_init_attach**(1M)**

initiate DR attach system board operation

dr_cmd_mem_info**(1M)**

show memory configuration on a system board in Tcl encoding

dr_cmd_obp_info**(1M)**

show complete config on a system board in Tcl encoding

dr_cmd_print_brd_info**(1M)**

show board resource in tabular format

dr_cmd_print_obp_info**(1M)**

show system board info per OpenBoot(tm) PROM in tabular format

dr_cmd_print_unsafe_info**(1M)**

show the open unsafe devices in tabular format

dr_cmd_unsafe_dev_info**(1M)**

show the open unsafe devices in Tcl encoding

drain**(1M)**

start memory drain

drshow**(1M)**

display DR and board resource info

drview**(1M)**

DR Graphical User Interface

init_attach**(1M)**

initiate DR Attach operation

moveboard**(1M)**

move a system board from one domain to another

rcfgadm**(1M)**

perform remote reconfiguration from the SSP

reconfig**(1M)**

initiate auto-configuration sequence

NAME	abort_attach – abort a DR Attach operation
SYNOPSIS	**abort_attach** *sb*
DESCRIPTION	Execute this command at the dr(1M) shell prompt to return the specified board to its original condition after completion of an init_attach(1M) operation. abort_attach leaves the board present, powered-on, and in no domain. It instructs the operating system running on the target domain specified by the SUNW_HOSTNAME environment variable to abandon the in-progress attach operation, then removes the board from the domain_config file and resets the Enterprise 10000 centerplane cluster mask registers and board domain mask registers. Refer to domain_config(4) in the *Sun Enterprise 10000 SSP Reference Manual*.

You should run abort_attach after init_attach(1M) has successfully completed, and instead of the complete_attach(1M) command.

If executing abort_attach fails to abort the operation, try repeating the attempt at a later time, or contact your service provider.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS	The following operand is supported.
	sb The board number (0 to 15) of the system board not to be attached

EXAMPLES	**EXAMPLE 1** Using abort_attach(1M)

```
dr> abort_attach 5
Aborting attach board 5 to domain ts4.
Processors on board 5 reset.
Removing board 5 from domain_config file.
Board 5 placed into loopback.
Abort attach board successful.
dr>
```

DIAGNOSTICS	The following diagnostics are supported:

Failed to abort board attachment

Repeat the abort_attach command at a later time, or contact your service provider.

EXIT STATUS   If successful, abort_attach returns a 0 in the dr_return global variable; if
                    not, it returns a 1, along with one or more diagnostic messages.

NOTES   If DR detects a usage syntax error, it immediately aborts the dr(1M) command,
                    displays the dr(1M) shell prompt, and leaves dr_return unmodified. See
                    dr(1M).

SEE ALSO   dr(1M), init_attach(1M)

NAME   abort_detach – abort a DR Detach operation

SYNOPSIS   **abort_detach** *sb*

DESCRIPTION   Execute this command at the dr(1M) shell prompt to abort an attempt to DR Detach a board. You can execute abort_detach after the board has been successfully executed, resources on the designated system board are once again available to the operating system.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS   The following operand is supported.
sb   The board number (0 to 15) of the system board not to be detached

EXAMPLES   **EXAMPLE 1**   Using abort_detach(1M)

```
dr> abort_detach 4
Aborting detach board 4
Returning board to domain_config.
Adding board 4 to domain_config file.
Abort board detach completed successfully.
```

DIAGNOSTICS   The following diagnostics are supported:

FAILED to restore domain_config file

Retry the ABORT board detach at a later time

The attempt to restore the board number to the target domain board list in the domain_config(4) file (refer to the man page in the *Sun Enterprise 10000 SSP 3.5 Reference Manual*) has failed. This may be a temporary condition, so try the abort_detach again at a later time.

Failed to abort board detach

The operating system on the target domain was unable to restore the board to full operation. This may be a temporary condition, so try the abort_detach again at a later time.

EXIT STATUS   If successful, abort_detach returns a 0 in the dr_return global variable; if not, it returns a 1, along with one or more diagnostic messages.

NOTES   If DR detects a usage syntax error, it immediately aborts the dr(1M) command, displays the dr(1M) shell prompt, and leaves dr_return unmodified. See dr(1M).

SEE ALSO   complete_attach(1M), dr(1M), drain(1M)

NAME | addboard – attach a system board to a specific domain

SYNOPSIS | **addboard** −d *domain* [−f] [−r *retry_count* [−t *timeout*]] [−q] −b *board_number*

addboard −d *domain* [−f] [−r *retry_count* [−t *timeout*]] [−q] SB*x*

addboard −h

DESCRIPTION | The addboard(1M) command attaches a system board to a domain specified by *domain* (also referred to as the target domain). It also writes progress messages to the platform log on the SSP and to standard output.

If addboard fails, it writes an error message to standard error and returns a nonzero exit status. The board stays in the state it is in when the command fails. You can rerun addboard, or you can use deleteboard(1M) to return the board to its original state.

The −r option specifies the maximum number of retry attempts, *retry_count*, in the event that an attach operation fails. The −t option specifies a time period in seconds, *timeout*, between retry attempts. If you specify the number of retries, but no timeout, there is no delay between retry attempts. If you specify a timeout, you must also specify the number of retries.

To be eligible for attachment, a board must be powered on and in one of the following states:

■ Not in a domain

■ In an intermediate state due to an incomplete DR operation

If the specified board is already in the domain, addboard returns an exit status of zero, indicating success.

OPTIONS | The following options are supported:

−b *board_number*   Specifies the system board number, 0 through 15.

−d *domain*   Specifies the domain name.

−f   Attempts to force the operation. The −f option will override certain software constraints, but it will never override fundamental safety and availability constraints of the hardware and Solaris operating environment.

−h   Displays a usage message.

−r *retry_count*   Specifies the number of retry attempts. There are no limits to the possible retries.

−t *timeout*	Specifies the time in seconds between retry attempts. There is no limit to the amount of time.
−q	Turns on quiet mode, which means informational messages are not written to standard output.

OPERANDS The following operand is supported:

SB*x*	Specifies the system board number, where *x* equals an integer from 0 through 15.

EXAMPLES In the following examples, the command attaches system board 2 to a domain named xf1-b4. If the first attempt to attach the board fails, two retries are made, if necessary, with a wait time of 10 minutes between retries.

EXAMPLE 1   To Add a Board With the −b Option

```
# addboard -b 2 -d xf1-b4 -r 2 -t 600
```

EXAMPLE 2   To Add a Board Without the −b Option

```
# addboard -d xf1-b4 -r 2 -t 600 SB2
```

EXIT STATUS The following exit values are supported.

0	Successfully attached the board to the specified domain.
1	Command syntax error occurred.
2	Failed to acquire DR lock.
3	Failed to release DR lock.
4	Failed to initialize DR communications.
11	The board or target domain is not eligible for attachment.
12	A board in the target domain is in an intermediate attach state.
13	Failed in initial attachment stage.
14	Failed in complete attachment stage.
15	Failed to dynamically link with DR library.
16	bringup -C requested process termination.

17   A termination signal was received.

51   The command was not completed because of a negative
          acknowledgment from the `confp->confirm` function.

52   System configuration administration is not supported on this
          attachment point.

53   System configuration administration operation is not supported on this
          attachment point.

54   The caller does not have the required process privileges. For example,
          if configuration administration is performed through a device driver,
          the permissions on the device node are used to control access.

55   The command was not completed because an element of the system
          configuration administration system was busy.

56   The command required a service interruption and was not completed
          because part of the system could not be suspended (that is, quiesced).

57   A procedural error occurred in the data.

58   A procedural error occurred in the library, including a failure to obtain
          process resources such as memory and file descriptors.

59   A hardware-specific library could not be located by using the specified
          attachment point.

60   The operation failed because of the condition of the attachment point.

61   The system configuration administration operation requested is not
          supported on the specified attachment point.

62   An error occurred during the processing of the requested operation.
          This error code includes validation of the command arguments by the
          hardware-specific code.

63   No such attachment point exists.

64   No attachment point with the specified attributes exists.

65   An invalid board ID was specified.

ERROR   If `addboard` fails, it writes diagnostic messages to standard error.
MESSAGES

NOTES   The addboard command prevents more than one attach or detach operation
                from executing concurrently.

SEE ALSO   deleteboard(1M), moveboard(1M), rcfgadm(1M)

                domain_status(1M) and showdevices(1M) in the *Sun Enterprise 10000
                SSP 3.5 Reference Manual*

                dcs(1M) and dr_daemon(1M) in *man pages section 1M: System
                Administration Commands* in the *Solaris 8 Reference Manual*

                Sun Enterprise 10000 Dynamic Reconfiguration User Guide

NAME	complete_attach – complete a DR Attach operation

SYNOPSIS   **complete_attach** *sb*

DESCRIPTION   Execute this command at the dr(1M) shell prompt to complete an attempt to DR Attach a board after successful execution of the of the init_attach(1M) command. complete_attach causes the operating system running on the target domain to dynamically add the resources (processors, memory, and I/O devices) from the specified board to the running system. If a problem that prevents attachment of any device present on the board occurs, that problem is logged in the system message buffer of the target domain. To display a list of the devices that were successfully attached, execute the drshow(1M) command to display the current system configuration for the board.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS   The following operand is supported.

sb   The board number (0 to 15) of the system board to be attached to the target domain.

EXAMPLES   **EXAMPLE 1**   Using complete_detach(1M)

```
dr> complete_attach 5
Completing attach for board 5
Board attachment completed successfully.
```

DIAGNOSTICS   The following diagnostics are supported:

```
Failed during final state transition
```

The operation failed during the final stage of attachment. Check that the DR daemon is still running on the target domain, and that the network is operational. To recover from the failure, repeat the complete_attach operation or execute an abort_attach(1M).

```
Failed to complete attach board
```

The operating system on the target domain was unable to attach the board. Repeat the complete_attach operation at a later time or execute the abort_attach(1M) command.

EXIT STATUS   If successful, complete_attach returns a 0 in the dr_return global variable; if not, it returns a 1, along with one or more diagnostic messages.

NOTES   If DR detects a usage syntax error, it immediately aborts the dr(1M) command, displays the dr(1M) shell prompt, and leaves dr_return unmodified. See dr(1M).

SEE ALSO   dr(1M), drshow(1M), init_attach(1M)

NAME   complete_detach – complete a DR detach operation

SYNOPSIS   **complete_detach** *sb* [force]

DESCRIPTION   Execute this command at the dr(1M) shell prompt to complete an attempt to
                  DR Detach a board. The drain(1M) must have been previously executed and
                  the drain operation must have completed before complete_detach can
                  proceed. You can use the drshow(1M) command to check the status of the
                  domain operation.

                  A board can be detached only after all use of its devices has ceased. DR
                  automatically terminates the use of memory and network devices and, in
                  almost all cases, processors; but you must terminate use of the board's I/O
                  devices. You can use the drshow(1M) command to list the devices in use on
                  the board.

                  If the detaching board contains non-pageable kernel or OBP memory, the
                  domain is quiesced during the complete_detach operation. The quiesce
                  operation may fail due to *forcible* conditions. Refer to the *Sun Enterprise 10000
                  Dynamic Reconfiguration User Guide* for a description of such conditions. You
                  can use the force argument to force the quiesce in such situations.

                  Note - This command is available only for DR 2.0 domains (that is, domains
                  that use the DR daemon) on the Sun Enterprise 10000 server. You can use
                  domain_status(1M) to determine the DR version running on the domain.

OPERANDS   The following operands are supported.

                  sb   The board number (0 to 15) of the system board to be
                                    detached

                  force   Force the domain quiesce operation. Refer to the *Sun
                                    Enterprise 10000 Dynamic Reconfiguration User Guide* for a
                                    description of such conditions. You can use the force
                                    argument to force the quiesce in such situations.

EXAMPLES   **EXAMPLE 1**   Using complete_detach(1M)

                  ```
                  dr> complete_detach 5
                  Completing detach of board 5.
                  Operating System has detached the board.
                  Processors on board 5 reset.
                  Board 5 placed into loopback.
                  Board detachment completed successfully.
                  ```

DIAGNOSTICS   The following diagnostics are supported:

```
Cannot COMPLETE detach until drain completes
```

The drain operation is still in-progress. Use drshow(1M) to monitor the drain. After it has completed, repeat the complete_detach command.

```
Board detachment failed
```

```
Retry the COMPLETE or ABORT the operation
```

A condition in the operating system on the domain prevented the detach from completing. Retry the operation at a later time, or use abort_detach(1M) to abort the detach.

EXIT STATUS   If successful, complete_detach returns a 0 in the dr_return global variable; if not, it returns a 1, along with one or more diagnostic messages.

NOTES   If DR detects a usage syntax error, it immediately aborts the dr(1M) command, displays the dr(1M) shell prompt, and leaves dr_return unmodified. See dr(1M).

SEE ALSO   abort_detach(1M), dr(1M), drain(1M), drshow(1M)

NAME	deleteboard – detach a system board from a domain

SYNOPSIS   **deleteboard** −b *board_number* [−f] [−r *retry_count* [−t *timeout*]] [−q]

deleteboard [−f] [−r *retry_count* [−t *timeout*]] [−q] SB*x*

deleteboard −h

DESCRIPTION   The deleteboard command detaches a system board from its current domain. It also writes progress messages to the platform log on the SSP and to standard output.

If deleteboard fails, it writes error information to standard error and returns a nonzero exit status. The board stays in the state it is in when the command fails. You can rerun deleteboard, or you can use addboard(1M) to return the board to its original state.

The −r option specifies the maximum number of retry attempts, *retry_count*, in the event that a detach operation fails. The −t option specifies a time period in seconds, *timeout*, between retry attempts. If you specify the number of retries, but no timeout, there is no delay between retry attempts. If you specify a timeout, you must also specify the number of retries.

To be eligible for detachment, a board must be powered on and in one of the following states:

■ In a domain

■ In an intermediate state due to an incomplete DR operation

If the specified board is not in a domain or has already been detached from its source domain, deleteboard returns an exit value of zero, indicating success. A DR platform error message indicates that the board is already detached.

OPTIONS   The following options are supported:

−b *board_number*	Specifies the system board number, 0 through 15.
−f	Attempts to force the operation. The −f option will override certain software constraints, but it will never override fundamental safety and availability constraints of the hardware and Solaris operating environment.
−h	Displays a usage message.
−r *retry_count*	Specifies the number of retry attempts. There are no limits to the possible retries.

−t *timeout*	Specifies the time in seconds between retry attempts. There is no limit to the amount of time.
−q	Turns on quiet mode, which means informational messages are not written to standard output.

OPERANDS

The following operand is supported:

SB*x*   Specifies the system board, where *x* equals an integer from 0 through 15.

EXAMPLES

In the following examples, the command detaches system board 2 from its current domain. If the first attempt to detach the board fails, two retries are made, if necessary, with a wait time of 10 minutes between retries.

EXAMPLE 1   To Delete a Board With the −b Option

deleteboard -b 2 -r 2 -t 600

EXAMPLE 2   To Delete a Board Without the −b Option

deleteboard -r 2 -t 600 SB2

EXIT STATUS

The following exit values are supported.

0   Successfully detached the specified board from its domain.

1   Command syntax error occurred.

2   Failed to acquire DR lock.

3   Failed to release DR lock.

4   Failed to initialize DR communications.

5   Board or current domain is not eligible for detachment.

6   A board in the current domain is in an intermediate detach state.

7   DR daemon does not allow a detach operation when it removes the last processor from the domain or when it drops the memory in the domain below the minimum requirement.

8   Failed during memory drain operation.

9   Drain operation failed to make progress.

10   Failed in attempting to complete the detach.

15   Failed to dynamically link with DR library.

17   A termination signal was received.

51   The command was not completed because of a negative
          acknowledgment from the `confp->confirm` function.

52   System configuration administration is not supported on this
          attachment point.

53   System configuration administration operation is not supported on this
          attachment point.

54   The caller does not have the required process privileges. For example,
          if configuration administration is performed through a device driver,
          the permissions on the device node are used to control access.

55   The command was not completed because an element of the system
          configuration administration system was busy.

56   The command required a service interruption and was not completed
          because part of the system could not be suspended (that is, quiesced).

57   A procedural error occurred in the data.

58   A procedural error occurred in the library, including a failure to obtain
          process resources such as memory and file descriptors.

59   A hardware-specific library could not be located by using the specified
          attachment point.

60   The operation failed because of the condition of the attachment point.

61   The system configuration administration operation requested is not
          supported on the specified attachment point.

62   An error occurred during the processing of the requested operation.
          This error code includes validation of the command arguments by the
          hardware-specific code.

63   No such attachment point exists.

64   No attachment point with the specified attributes exists.

65   An invalid board ID was specified.

**ERROR
MESSAGES**

If deleteboard fails, it writes diagnostic messages to standard error.

NOTES

The deleteboard command prevents more than one attach or detach operation from executing concurrently.

SEE ALSO

addboard(1M), moveboard(1M), rcfgadm(1M)

domain_status(1M) and showdevices(1M) in the *Sun Enterprise 10000 SSP 3.5 Reference Manual*

dcs(1M) and dr_daemon(1M) in *man pages section 1M: System Administration Commands* in the *Solaris 8 Reference Manual*

Sun Enterprise 10000 Dynamic Reconfiguration User Guide

NAME | dr – initiate dynamic reconfiguration shell

SYNOPSIS | **dr**

DESCRIPTION | The `dr` command initiates the Dynamic Reconfiguration (DR) shell, a Tcl application (see NOTES, below) with DR command extensions. You can use the `dr` shell to logically attach or detach a system board to or from a Sun Enterprise 10000 domain from the command line or by using a script.

Note - Whenever possible, use the DR GUI in Hostview to execute Dynamic Reconfiguration operations. Use the `dr` shell when you cannot run Hostview; for example, if you need to run DR over a dial-up connection. For more information, see the *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* and `hostview`(1M) in the *Sun Enterprise 10000 SSP 3.5 Reference Manual.*

When executed on the command line, `dr` connects to the domain specified by the SUNW_HOSTNAME environment variable. After this connection is established, `dr` displays the dr> prompt, which accepts the DR commands.

To see the list of DR commands if you not using AnswerBook2, execute `man Intro` on the SSP while logged in as user ssp.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use `domain_status`(1M) to determine the DR version running on the domain.

You can quit the `dr` shell at any time by typing `exit` or Control-d.

Caution: Do not execute any of the DR commands that begin with `dr_cmd_`; these are low-level commands that are for use only by authorized service personnel under special circumstances, as described in `dr.service`.

To minimize the risk of unintended DR operations, start this shell only when you are ready to execute DR commands and exit it as soon as you are done.

The DR commands return error status in the global Tcl variable `dr_return`. Normally, Tcl commands return both output and status together, which can be confusing and difficult to parse from within scripts. You can, however, execute the DR command `set dr_return` to display `dr_return` after executing each DR command, to determine command success or failure. Though, under most circumstances, the diagnostic messages output by the `dr` shell clearly indicate success or failure.

Note - Type `help` at the `dr` shell prompt (dr>) to access DR quick-reference help guide.

EXAMPLES | **EXAMPLE 1** Using dr(1M)

The following example performs a DR Attach of Board 2 to the domain named
"e100001". After complete_attach(1M) has successfully completed dr
displays the result code stored in dr_return

```
e100001-ssp% domain_switch e100001
e100001-ssp% dr
Checking environment...
Initializing SSP SNMP MIB...
Establishing communication with DR daemon...
e100001: System Status - Summary

BOARD #: 2 3 5 6 physically present.
BOARD #: 0 1 4 being used by the system.

dr> init_attach 2
Initiate attaching board 2

phase init_reset: Initial system resets...

phase jtag_integ: JTAG probe and integrity test...
phase mem_probe: Memory dimm probe...
phase jtag_bbsram: JTAG basic test of bootbus sram...
phase procl: Initial processor module tests...
phase pc/cic_reg: PC and CIC register tests...
phase dtag: CIC DTAG tests...
phase mem: MC register and memory tests...
phase procmem: Processor vs. memory tests...
phase xcall: Interprocessor interrupt tests...
phase io: I/O controller tests...
Skipping phase ecc: Proc ecc vs. memory tests...
phase final_config: Final configuration...
Creating OBP handoff structures...
Configured in 3F with 3 processors, 0 SBus cards, 1024 MBytes
memory.
Boot processor is 4.0 = 8
POST execution time 1:23
hpost is complete.
/opt/SUNWssp/bin/obp_helper
Master cpu is 8
Slave cpus initialization:
Slave cpus initialization OK
board debut utility complete.

Board attachment initiated successfully.

Ready to COMPLETE board attachment.

dr> complete_attach 2
Completing attach for board 2
Board attachment completed successfully.
dr> set dr_return
0
dr> exit
e100001-ssp%
```

NOTES   Tcl (Tool command language) is a simple scripting language forr controlling
and extending applications. You do not need Tcl knowledge to use the dr shell.

As a Tcl application, dr checks for certain types of syntax errors and, if it finds
one, aborts without executing the dr shell command. For example, if you
specify an argument with a command that does not require one, dr prints a
usage error message and aborts. dr updates dr_return only upon
completion of a dr command. If the command does not complete, as in our
example above, dr does not update dr_return.

SEE ALSO   *Sun Enterprise 10000 Dynamic Reconfiguration User Guide*

Sun Enterprise Server Alternate Pathing 2.3 User Guide

Sun Enterprise 10000 SSP 3.5 User Guide

domain_switch(1M), hostview(1M) in the *Sun Enterprise 10000 SSP 3.5
Reference Manual*

dr(7) in *man pages section 7: Device and Network Interfaces*

add_drv(1M), drvconfig(1M), devlinks(1M), disks(1M), inetd(1M),
ports(1M), prtconf(1M), tapes(1M) in *man pages section 1M: System
Administration Commands*

syslog(3C) in *man pages section 3: Basic Library Functions*

NAME	dr.service – low-level DR commands for service providers
DESCRIPTION	The low-level commands described here, which begin with `dr_cmd`, are available only in the DR shell and are for use by service providers only. Service providers should use them only when they need a finer level of control to debug failing DR operations, or when they cannot access the DR GUI.

The DR shell provides commands that directly map to `libdr.so` function calls. Executing this command set gives the caller a finer level of control over DR operations, but introduces additional risk of error due to fewer safeguards.

Note that DR operations can fail to be denied by the operating system for numerous reasons. Often, specific user action is required to complete a DR sequence. For this reason, Sun cautions against the use of automated DR scripts. The Hostview interface (refer to `hostview`(1M) in *Sun Enterprise 10000 SSP 3.5 Reference Manual*) is the preferred method of performing DR operations. Use the `dr`(1M) shell when the GUI-based Hostview application is unavailable.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use `domain_status`(1M) to determine the DR version running on the domain.

CAUTION	Customers should not use these low-level commands, but should access DR through the DR GUI, as described in the *Sun Enterprise 10000 Dynamic Reconfiguration User Guide*, or by using the high-level DR commands (those without the `dr_cmd` prefix) described in this reference manual.
SHELL COMMANDS	The low-level shell commands are those that begin with `dr_cmd`. See `Intro`(1M).
EXIT STATUS	The DR shell low-level command set generally returns an exit code in the `dr_return` global variable. Upon return from each of the DR commands, this variable can be tested for success or failure.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves `dr_return` uninitialized. In such cases, the `dr_return` error code is meaningless. See `dr`(1M) for more information concerning return codes.

NAME	dr_cmd_a_attach – abort DR attach system board operation
SYNOPSIS	**dr_cmd_a_attach** *sb*
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness. Instead, use abort_attach(1M), which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	If abort_attach(1M) were unavailable for some reason, you could run dr_cmd_init_attach(1M) and before the board has been completely attached by using the dr_cmd_c_attach(1M). dr_cmd_a_attach returns the board to the state it was in prior to the dr_cmd_init_attach(1M) operation; that is, present, powered-on, and in no domain.

dr_cmd_a_attach instructs the operating system running on the target domain to abandon the in-progress attach operation, removes the system board from the domain_config file, and resets the shared memory mask registers and board domain mask registers on the centerplane.

Some conditions that are transparent to the user may cause an abort failure. Therefore, if dr_cmd_a_attach fails to complete the abort successfully, try executing it again at a later time.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS	The following operand is supported.

sb   The system board number (0 to 15) for the abort attach operation

DIAGNOSTICS	See DIAGNOSTICS on abort_attach(1M).
EXIT STATUS	If abort_attach(1M) succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO	dr(1M), dr_cmd_init_attach(1M), dr_cmd_c_attach(1M)

NAME	dr_cmd_a_detach – abort DR detach system board operation
SYNOPSIS	**dr_cmd_a_detach** *sb*
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness. Instead, use abort_attach(1M), which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	You can run dr_cmd_a_detach after draining a system board by using dr_cmd_drain(1M) , but before that board has been completely detached.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS	The following operand is supported:

sb   The board number (0 to 15) of the system board whose detach is being aborted.

DIAGNOSTICS	See DIAGNOSTICS in abort_detach(1M).
EXIT STATUS	If dr_cmd_a_detach succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO	dr(1M), dr_cmd_drain(1M)

NAME | dr_cmd_auto_config – run Solaris reconfig sequence on target domain

SYNOPSIS | **dr_cmd_auto_config**

CAUTION | Do not use this command, use reconfig(1M) instead. Only authorized service providers should use dr_cmd_auto_config, which runs in the DR shell, and only when they cannot use reconfig(1M). Performing this operation may cause device files to be remapped and known devices to be renamed.

Note - As of the Solaris 8 GA release, manual reconfiguration is not needed. A new DDI subsystem, devfsadm, completes all of the reconfiguration tasks.

DESCRIPTION | The system administrator would normally run dr_cmd_auto_config after a new system board has been attached to a running domain to make the devices on the boards available immediately. The automatic configuration of the Solaris operating environment consists of the following commands, in the order shown:

drvconfig(1M), devlinks(1M), disks(1M), and tapes(1M).

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

DIAGNOSTICS | See DIAGNOSTICS in the reconfig(1M) man page.

EXIT STATUS | If dr_cmd_auto_config succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO | reconfig(1M) in this reference manual

drvconfig(1M), devlinks(1M), disks(1M), dr_daemon(1M), ports(1M), tapes(1M) in *man pages section 1M: System Administration Commands*

NAME	dr_cmd_c_attach – complete DR attach system board operation
SYNOPSIS	**dr_cmd_c_attach** *sb*
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness, and is dangerous. Instead, use complete_attach(1M), which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	dr_cmd_c_attach completes the DR attach board operation started by dr_cmd_init_attach(1M). The designated system board should already have been successfully Init Attached via dr_cmd_init_attach(1M). The complete attach operation causes the operating system on the target domain to dynamically add the resources from this system board (processors, memory, and I/O devices) to the running system. If a problem occurs, preventing attachment of any device present on the board, the problem is logged in the system message buffer of the target domain.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS	The following operand is supported.
	sb The board number (0 to 15) of the system board being attached
DIAGNOSTICS	See DIAGNOSTICS on the complete_attach(1M) man page.
EXIT STATUS	If dr_cmd_c_attach succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO	dr(1M), dr_cmd_init_attach(1M)

NAME | dr_cmd_c_detach – complete DR detach system board operation

SYNOPSIS | **dr_cmd_c_detach** *sb*

CAUTION | Do not use this command, which runs in the DR shell; it is included here only for completeness, and is dangerous. Instead, use complete_detach(1M), which performs the same functions, but with the added security of safeguards and checks.

DESCRIPTION | dr_cmd_c_detach completes a DR detach board operation. The designated system board should already have been drained via dr_cmd_drain(1M).

You can detach a system board only when none of its devices is in use. DR automatically terminates the use of memory, processors (in almost all cases), and network devices on the board. But the administrator must make certain that all use of the I/O devices has ceased. You can use drshow(1M) to list the devices in use on a given system board.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS | The following operand is supported.
sb   The board number (0 to 15) of the system board being detached

DIAGNOSTICS | See DIAGNOSTICS on complete_detach(1M).

EXIT STATUS | If dr_cmd_c_attach succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO | dr(1M), dr_cmd_drain(1M)

NAME	dr_cmd_c_f_detach – force completion of DR detach system board operation
SYNOPSIS	**dr_cmd_c_f_detach** *sb*
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness, and is dangerous. Instead, use complete_detach(1M), which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	dr_cmd_c_f_detach completes a DR detach board operation, using a forcible domain quiesce. See the CAUTION, above. Use this command when you need to force the system to complete a detach operation, when the system board to be detached contains unsafe devices that are open, but not in use. Refer to the *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* for more information about system quiesce and ways to increase the safety of this dangerous command.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS	The following operand is supported.
	sb　　　　　　　The board number (0 to 15) of the system board to be detached
DIAGNOSTICS	See DIAGNOSTICS on the complete_detach(1M) man page.
EXIT STATUS	If dr_cmd_c_f_detach succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO	dr(1M), dr_cmd_drain(1M)

NAME | dr_cmd_cpu_info – show processors on a system board in Tcl encoding

SYNOPSIS | **dr_cmd_cpu_info** *sb*

CAUTION | This command, which runs in the DR shell, produces output in a form suitable for the drview(1M) application, not the interactive user.

DESCRIPTION | dr_cmd_cpu_info queries the target domain and produces a list of the processors attached to the specified system board. This list is returned in a Tcl format, and is used by the drview(1M) application.

Since the Tcl list is not readily accessible to an interactive user, you should use drshow(1M) instead to acquire processor information.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS | The following operand is supported.
sb   The board number (0 to 15) of the target system board

EXIT STATUS | If dr_cmd_cpu_info succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO | dr(1M), dr_cmd_mem_info(1M), dr_cmd_dev_info(1M)

NAME	dr_cmd_debug – toggle DR library-level debugging
SYNOPSIS	**dr_cmd_debug**
CAUTION	Only authorized service providers should use this command, which runs in the DR shell.
DESCRIPTION	When switched on, dr_cmd_debug provides significantly more detailed information about DR operations performed by using dr(1M). dr_cmd_debug is set up as a toggle; execute it once to turn it on, and again to turn it off. Initially, it is set to 0, or off.

The service provider may find dr_cmd_debug very useful when diagnosing a DR-related failure. Activate debugging prior to executing any commands related to DR Attach or DR Detach.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

EXIT STATUS	dr_cmd_debug always returns a 0 character in the dr_return global Tcl variable.
SEE ALSO	dr(1M)

NAME | dr_cmd_detach_allow – verify a system board can support DR detach

SYNOPSIS | **dr_cmd_detach_allow** *sb*

CAUTION | Only authorized service providers should use this command, which runs in the DR shell.

DESCRIPTION | dr_cmd_detach_allow queries the operating system running on the target domain about any conditions that may prevent the system board from being successfully detach. If the board is not detachable, dr_cmd_detach_allow displays one or more diagnostic messages.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS | The following operand is supported.
sb   The board number (0 to 15) of the system board to be queried

EXIT STATUS | If dr_cmd_detach_allow succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1.

SEE ALSO | dr(1M)

NAME	dr_cmd_dev_info – show devices on a system board in Tcl list encoding
SYNOPSIS	**dr_cmd_dev_info** *sb*
DESCRIPTION	

Note - This command, which runs in the DR shell, produces output in a form suitable for the drview(1M) application, not the interactive user. Use drshow(1M) instead to view device information.

dr_cmd_dev_info checks the target domain for peripheral devices attached to the specified system board and returns the information in a Tcl list encoding, which is used by the drview(1M) application.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS   The following operand is supported.

sb   The board number (0 to 15) of the target system board

EXIT STATUS   If dr_cmd_dev_info succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnosic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO   dr(1M), dr_cmd_cpu_info(1M), dr_cmd_mem_info(1M)

NAME | dr_cmd_drain – start memory drain on a system board.

SYNOPSIS | **dr_cmd_drain** *sb*

CAUTION | This command, which runs in the DR shell, is dangerous; do not use it. It is included here only for completeness. Instead, use the drain(1M) command, which performs the same functions, but with the added security of safeguards and checks.

DESCRIPTION | dr_cmd_drain determines the best way to vacate memory physically located on the designated system board. It may simply flush the memory, or copy it to memory available on another system board in the same domain. If a suitable target memory for the memory copy is not available when the dr_cmd_drain command is invoked, the request is denied. If the unavailability is due to run-time conditions and system load, you should retry the dr_cmd_drain operation at a later time.

The dr_cmd_drain operation also removes the system board from the board list in the domain_config(4) file on the SSP. (Refer to the domain_config(4) man page in the *Sun Enterprise 10000 SSP 3.5 Reference Manual*.

dr_cmd_drain begins execution, then quickly exits. Use drshow(1M) to monitor its progress.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS | The following operand is supported.
sb   The board number (0 to 15) of the system board to be drained

DIAGNOSTICS | See DIAGNOSTICS on drain(1M).

EXIT STATUS | If dr_cmd_drain succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnosic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO | dr_cmd_mem_info(1M)

NAME   dr_cmd_drain_status – show state of in-progress memory drain.

SYNOPSIS   **dr_cmd_drain_status** *sb*

CAUTION   Only authorized service providers should use this command, which runs in
                    the DR shell.

DESCRIPTION   Use dr_cmd_drain_status to monitor a drain-in-progress. It displays a
                    table of current information about the drain. DR cannot complete a detach
                    until all the memory on a system board has been successfully drained.

                    Note - This command is available only for DR 2.0 domains (that is, domains
                    that use the DR daemon) on the Sun Enterprise 10000 server. You can use
                    domain_status(1M) to determine the DR version running on the domain.

OPERANDS   The following operand is supported.
                    sb   The board number (0 to 15) of the system board being
                                        drained

EXIT STATUS   If dr_cmd_drain_status succeeds it returns a 0 result code in the
                    dr_return global variable. If it fails, it returns a 1.

                    Note - Tcl parsing errors prevent DR commands from running which, in turn,
                    leaves dr_return uninitialized. In such cases, the dr_return error code is
                    meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO   dr(1M)

NAME dr_cmd_eligible_attach – verify a system board is eligible for DR attach

SYNOPSIS **dr_cmd_eligible_attach** *sb*

CAUTION Only authorized service providers should use this command, which runs in the DR shell. Be sure to run this eligibility check prior to initiating any DR attach activity when using the low-level DR shell command set. Initiating an attach operation on an ineligible board may cause a system failure.

DESCRIPTION Use dr_cmd_eligible_attach to verify that a system board is eligible for an attach operation before using dr_cmd_init_attach(1M) to begin the Init Attach.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS The following operand is supported.

sb   The board number (0 to 15) of the system board to be checked

EXIT STATUS dr_cmd_eligible_attach returns one of the following result codes to the dr_return global Tcl variable.

Y   The specified system board is eligible to be attached.

n   The specified system board is not eligible to be attached. dr_cmd_eligible_attach sends additional information to stdout.

board_number   The specified system board is not eligible to be attached because system board *board_number* (0 to 15), a different system board in the target domain, is in an intermediate DR Attach state. That DR Attach operation must be completed before you can initiate a DR operation on another board (such as the one specified).

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO dr(1M)

NAME	dr_cmd_eligible_detach – verify a system board is eligible for DR detach
SYNOPSIS	**dr_cmd_eligible_detach** *sb*
CAUTION	Only authorized service providers should use this command, which runs in the DR shell. Service providers: Be sure to run this eligibility check prior to initiating any DR detach activity when using the low-level DR shell command set. Initiating a detach operation on an ineligible board may cause a system failure.
DESCRIPTION	Use `dr_cmd_eligible_detach` to verify that a system board is eligible for a detach operation before using `dr_cmd_drain`(1M) to begin a DR drain operation.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use the `domain_status`(1M) to determine the DR version running on the domain.

OPERANDS	The following operand is supported.

sb	The board number (0 to 15) of the system board to be checked

EXIT STATUS	`dr_cmd_eligible_detach` returns one of the following result codes to the `dr_return` global Tcl variable.

Y	The specified system board is eligible to be detached.
n	The specified system board is not eligible to be detached. `dr_cmd_eligible_detach` sends additional information to stdout.
board_number	The specified system board is not eligible to be detached because system board *board_number* (0 to 15), a different system board in the target domain, is in an intermediate DR Detach state. That DR Detach operation must be completed before you can initiate a DR operation on another board (such as the one specified).

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves `dr_return` uninitialized. In such cases, the `dr_return` error code is meaningless. See `dr`(1M) for more information concerning return codes.

SEE ALSO	`dr`(1M)

NAME   dr_cmd_init_attach – initiate DR attach system board operation

SYNOPSIS   **dr_cmd_init_attach** *sb*

CAUTION   Do not use this command, which runs in the DR shell; it is dangerous, and is
                  included here only for completeness. Instead, use init_attach(1M), which
                  performs the same functions, but with the added security of safeguards and
                  checks.

DESCRIPTION   dr_cmd_init_attach begins a DR attach board operation. DR does not
                  screen the target domain for intermediate system boards as it does with the
                  init_attach(1M) command and through Hostview.

                  dr_cmd_init_attach is a low-level command for use only by trained
                  service personnel for diagnosing DR-related system problems. The designated
                  system board should be present, powered-on, and currently in no domain.
                  dr_cmd_init_attach diagnoses, then debuts the system board to the target
                  domain specified in the SUNW_HOSTNAME environment variable.

                  dr_cmd_init_attach adds the system board to the system board list in the
                  SSP domain_config file. (Refer to the domain_config(4) man page in the
                  Sun Enterprise 10000 SSP 3.5 Reference Manual.) DR then prepares the
                  resources (processors, memory, and I/O controllers) for attachment by the
                  operating system, and the centerplane is reconfigured such that the board is
                  visible to the target domain.

                  After dr_cmd_init_attach completes successfully, you can execute
                  dr_cmd_c_attach(1M) to complete the attach operation, or
                  dr_cmd_a_attach(1M) to abort it.

                  Note - This command is available only for DR 2.0 domains (that is, domains
                  that use the DR daemon) on the Sun Enterprise 10000 server. You can use
                  domain_status(1M) to determine the DR version running on the domain.

OPERANDS   The following operand is supported.
                  sb   The board number (0 to 15) of the system board to be
                                  attached

DIAGNOSTICS   See DIAGNOSTICS on the init_attach(1M) man page.

EXIT STATUS   If dr_cmd_init_attach succeeds it returns a 0 result code in the dr_return
                  global variable. If it fails, it returns a 1 and displays diagnostic messages.

> **Note -** Tcl parsing errors prevent DR commands from running which, in turn,
> leaves `dr_return` uninitialized. In such cases, the `dr_return` error code is
> meaningless. See `dr`(1M) for more information concerning return codes.

SEE ALSO   dr(1M), `init_attach`(1M)

NAME	dr_cmd_mem_info – show memory config on a system board in Tcl encoding
SYNOPSIS	**dr_cmd_mem_info** *sb*
CAUTION	Do not use this command. It returns information in Tcl encoding, which is understood by the drview(1M) application, but is not intended for direct viewing by users. Instead, use the drshow(1M) command.
DESCRIPTION	dr_cmd_mem_info queries the target domain for memory attached to this system board, returning the information in a Tcl list encoding, which then is used by the drview(1M) application.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS	The following operand is supported.
	sb The board number (0 to 15) of the system board to be checked
EXIT STATUS	If dr_cmd_mem_info succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

SEE ALSO	dr(1M)

NAME	dr_cmd_obp_info – show complete config of a system board in Tcl encoding
SYNOPSIS	**dr_cmd_obp_info** *sb*
CAUTION	Do not use this command, which runs in the DR shell; it displays information in Tcl encoding, which is understood by the drview(1M) application, but is not intended for direct viewing by the interactive user. Instead, use drshow(1M) to view this information.
DESCRIPTION	dr_cmd_obp_info displays the complete board configuration, including processors, memory and I/O devices, of a system board that has completed the Init Attached phase to a domain (that is, probed by OBP), but has not yet been completely attached.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS	The following operand is supported.
sb	The board number (0 to 15) of the target system board
EXIT STATUS	If dr_cmd_obp_info succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

NAME dr_cmd_print_brd_info – show board resource in tabular format

SYNOPSIS **dr_cmd_print_brd_info** *sb flags*

CAUTION Do not use this command, which runs in the DR shell; it is included here only for completeness. Instead, use drshow(1M), which presents the information in a more reabable format.

DESCRIPTION dr_cmd_print_brd_info obtains configuration information about the specified attached system board. The *flags* option specifies the information this command is to display, in the form of a bitstring, as follows:

```
Flag  Value    Display

-----------    -------

  1            Processor information

  2            Controller and peripheral information

  4            Memory configuration

  8            Memory cost information

 16            Memory drain status
```

You can obtain multiple displays by OR'ing (summing) the above decimal values. All displays are in a readable, tabular format.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS The following operands are supported.

sb   The board number (0 to 15) of the target system board

flags   A bitstring in decimal that represents the desired information

EXAMPLES **EXAMPLE 1** Displaying the Processor and Memory Configuration

To display the processor and memory configuration, use the following command.

```
dr> dr_cmd_print_brd_info 5
```

EXAMPLE 2   Displaying the Configuration Information

To display all configuration information, use the following command.

```
dr> dr_cmd_print_brd_info 31
```

EXIT STATUS   If dr_cmd_print_brd_info succeeds it returns a 0 result code in the
dr_return global variable. If it fails, it returns a 1 and displays diagnostic
messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn,
leaves dr_return uninitialized. In such cases, the dr_return error code is
meaningless. See dr(1M) for more information concerning return codes.

NAME   dr_cmd_print_obp_info – show system board info per OpenBoot(tm) Prom in
                    tabular format

SYNOPSIS   **dr_cmd_print_obp_info** *sb*

CAUTION   Do not use this command, which runs in the DR shell; it is included here only
                    for completeness. Instead, use the drshow(1M) command, which presents the
                    information in a more reabable format.

DESCRIPTION   dr_cmd_print_obp_info obtains configuration information from the
                    OpenBoot PROM, then displays that information in a tabular format. Use this
                    command to interrogate a system board that has passed the init attach stage,
                    but has not yet passed complete attach stage.

                    Note - This command is available only for DR 2.0 domains (that is, domains
                    that use the DR daemon) on the Sun Enterprise 10000 server. You can use
                    domain_status(1M) to determine the DR version running on the domain.

OPERANDS   The following operand is supported.
                    sb   The board number (0 to 15) of the target system board

EXIT STATUS   If dr_cmd_print_obp_info succeeds it returns a 0 result code in the
                    dr_return global variable. If it fails, it returns a 1 and displays diagnostic
                    messages.

                    Note - Tcl parsing errors prevent DR commands from running which, in turn,
                    leaves dr_return uninitialized. In such cases, the dr_return error code is
                    meaningless. See dr(1M) for more information concerning return codes.

NAME	dr_cmd_print_unsafe_info – show the open devices in tabular format
SYNOPSIS	**dr_cmd_print_unsafe_info**
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness. Instead, use drshow(1M), which presents the information in a more reabable format.
DESCRIPTION	dr_cmd_print_unsafe_info queries the target domain to determine if any unsafe peripheral devices are open. (Refer to the *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* for more information concerning DR unsafe devices.) If it finds that any such devices are open, it sends that information to stout.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

If dr_cmd_print_unsafe_info succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

NAME | dr_cmd_unsafe_dev_info – show the open unsafe devices in TCL encoding

SYNOPSIS | **dr_cmd_unsafe_dev_info**

CAUTION | Do not use this command, which runs in the DR shell; it is included here only for completeness. Instead, use drshow(1M), which presents the information in a more reabable format.

DESCRIPTION | dr_cmd_unsafe_dev_info queries the target domain to determine if any unsafe peripheral devices are open. (Refer to the *Sun Enterprise 10000 Dynamic Reconfiguration User Guide* for more information concerning DR unsafe devices.) If it finds that any such devices are open, it returns that information in a Tcl list encoding, which is used by the drview(1M) application.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

If dr_cmd_unsafe_dev_info succeeds it returns a 0 result code in the dr_return global variable. If it fails, it returns a 1 and displays diagnostic messages.

Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves dr_return uninitialized. In such cases, the dr_return error code is meaningless. See dr(1M) for more information concerning return codes.

NAME	drain – start memory drain
SYNOPSIS	**drain** *sb* [wait]
DESCRIPTION	The drain command, which you execute from the dr(1M) prompt, is the first of a two-step procedure for DR detaching a system board. The primary function of the drain command is to determine how the memory physically located on the designated board should be vacated. This memory may be simply flushed, or it may be copied to memory available on another system board in the same domain.

If a suitable target memory for the memory copy is not available when the drain command is invoked, the request is denied. If the unavailability is due to run-time conditions and system load, you can retry the drain operation at a later time.

The drain command starts the drain operation, and then returns. The drain may take several minutes to complete. You can execute drshow *sb* DRAIN to monitor its progress (see the drshow(1M) man page). Or, you can specify the wait option, and the drain returns only after the board has been fully drained, or drain detects an error. drain automatically displays the board status once before returning.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS	The following operand is supported.	
	sb	The board number (0 to 15) of the system board to be drained
	wait	Polls the DR daemon every 5 seconds and returns to the caller only after the drain completes. This option is useful when the drain is performed by a script. This option is case-insensitive.

EXAMPLES

EXAMPLE 1 Using drain(1M)

```
ts4-ssp% domain_switch ts4
ts4-ssp% dr
Checking environment...
Establishing Control Board Server connection...
Initializing SSP SNMP MIB...
Establishing communication with DR daemon...
                    ts4: System Status - Summary
BOARD #: 1 3 4 5 being used by the system.
```

```
dr> drain 5
Removing board 5 from domain_config file.
Start draining board 5.
Board drain started. Retrieving System Info...

            Bound Processes for Board 5

cpu   user  sys  procs
---   ----  ---  -----
 20    0     1
 21    0     1
 22    0     1
 23    0     1

No active IO devices.

            Memory Drain for Board 5 - IN PROGRESS

Reduction           = 1024 Mbytes
Remaining in System    = 2048 MBytes
Percent Complete      = 0% (1048576 KBytes remaining)

Drain operation started at Sun Sep 15 22:50:57 1996
Current time              Sun Sep 15 22:50:57 1996
Memory Drain is in progress. When Drain has finished,
you may COMPLETE the board detach.
```

EXIT STATUS   Upon successful initiation of the drain, drain returns a 0 in the dr_return
global variable; if the initiation fails, it returns a 1. If wait is specified, a 0 in
the dr_return indicates that the drain (not just initiation of it) has completed
successfully, and a 1 indicates that the drain has failed.

NOTES   If DR detects a usage syntax error, it immediately aborts the dr(1M) command,
displays the dr(1M) shell prompt, and leaves dr_return unmodified. See
dr(1M).

SEE ALSO   dr(1M) in this reference manual

domain_switch(1M) in the *Sun Enterprise 10000 SSP 3.5 Reference Manual*

NAME	drshow – display DR and board resource information
SYNOPSIS	**drshow** UNSAFE [*interval*	*count*]
	drshow *sb* [*report_type*][*interval*	*count*]
	drshow ALL [*report_type*][*interval*	*count*]
DESCRIPTION	drshow displays board-level and system-level resources and information about DR. It presents the displays in a tabular format.

drshow can sample at a specified interval (in seconds), for a given number of times. This polling capability is especially useful to monitor an in-progress drain operation.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS The following operands are supported.

UNSAFE	Displays all unsafe devices that are open throughout the domain.
sb	The board number (0 to 15) of the target domain
ALL	Reports the requested information for all active system boards in the domain.
report_type	Specifies a specific report type. All of the following keywords for report types are case-insensitive.

CPU – Displays processor information for the board (default)

DRAIN – Displays the progress of any active drain operation

IO – Shows the devices attached to this board

OBP – Displays tthe board configuration as OBP sees it. The OBP display can be used on a board that has completed the init-attach phase, but has not yet been completely attached. The OBP display may not be as accurate as the CPU/MEM/IO displays for boards in use.

MEM – Displays the memory configuration of this board

interval   The frequency, in seconds, with which drshow is
                                  to repeat the display

count   The number of times drshow is to repeat the
                                  display

EXAMPLES   **EXAMPLE 1**   Using drshow(1M)

```
dr> drshow 1 IO
                          I/O Bus Controllers and Devices for Board 1
-------------------------------- I/O Bus 1 : Slot 0 : esp0 ----------------------
device     opens   name                           usage
------     -----   ----                           -----
sd0          0     /dev/dsk/c0t0d0s0
sd1         26     /dev/dsk/c0t1d0s0              /
             0     /dev/dsk/c0t1d0s1              swap, /tmp
             9     /dev/dsk/c0t1d0s3              /var
             1     /dev/dsk/c0t1d0s5              /opt
            18     /dev/dsk/c0t1d0s6              /usr
             1     /dev/dsk/c0t1d0s7              /export
sd2          0     /dev/dsk/c0t2d0s0
sd3          0     /dev/dsk/c0t3d0s1              swap, /tmp
             0     /dev/dsk/c0t3d0s7              /xfer
-------------------------------- I/O Bus 1 : Slot 1 : qec0 ----------------------

device     opens   name                           usage
------     -----   ----                           -----
qe0                qe0                            ts4 (129:153:49:118)
qe1                qe1
qe2                qe2
qe3                qe3
```

NOTES   Exercise caution when using repeating displays. The only way to prematurely
                  stop one is by hitting Control-C, which terminates the DR shell.

EXIT STATUS   drshow returns a character 0 result code in dr_return.

SEE ALSO   dr(1M)

                  Sun Enterprise 10000 Dynamic Reconfiguration User Guide

NAME	drview – DR Graphical User Interface
SYNOPSIS	**drview**
DESCRIPTION	drview is the graphical user interface for the Dynamic Reconfiguration feature. Do not invoke it directly; it is automatically initiated by Hostview. Refer to the hostview(1M) man page in the *Sun Enterprise 10000 SSP 3.5 Reference Manual*.

For more information about Hostview refer to the *Sun Enterprise 10000 SSP 3.5 User Guide*, and for more information about drview, refer to the *Sun Enterprise 10000 Dynamic Reconfiguration User Guide*.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

SEE ALSO   hostview(1M) in the *Sun Enterprise 10000 SSP 3.5 Reference Manual*

Sun Enterprise 10000 SSP 3.5 User Guide

NAME | init_attach – initiate a DR attach operation

SYNOPSIS | **init_attach** *sb*

DESCRIPTION | Execute this command at the dr(1M) shell prompt to begin a DR Attach operation. The system board to be attached must be present, powered-on, and currently not attached to a domain. It is diagnosed and debuted to the target domain specified by the SUNW_HOSTNAME environment variable. Upon completion of the init_attach, the resources (processors, memory, and I/O controllers) are prepared for attachment by the operating system. The board is added to the board list in the SSP domain_config(4) file, and the centerplane is reconfigured such that the board is visible to the target domain.

Upon successful completion of init_attach you can use complete_attach(1M) to complete the attach operation or abort_attach(1M) to abort it.

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

OPERANDS | The following operand is supported.

sb   The board number (0 to 15) of the system board to be attached

EXAMPLES | **EXAMPLE 1**   Using init_attach(1M)

```
ts4-ssp% domain_switch ts4
ts4-ssp% dr
Checking environment...
Establishing Control Board Server connection...
Initializing SSP SNMP MIB...
Establishing communication with DR daemon...
        ts4: System Status - Summary
BOARD #: 5 physically present.
BOARD #: 1 3 4 being used by the system.


dr> init_attach 5
Initiate attaching board 5 to domain ts4..
Adding board 5 to domain_config file.
/opt/SUNWssp/bin/hpost -H20,4
Opening SNMP server library...

Reading centerplane asics to obtain bus configuration...
Bus configuration established as 3F.
phase cplane_isolate: CP domain cluster mask clear...
phase init_reset: Initial system resets...
phase jtab_integ: JTAG probe and integrity test...
phase mem_probe: Memory dimm probe...
```

```
phase iom_probe: I/O module type probe...
phase jtag_bbsram: JTAG basic test of bootbus sram...
phase procl: Initial processor module tests...
phase pc/cic_reg: PC and CIC register tests...
phase dtag: CIC DTAG tests...
phase mem: MC register and memory tests...
phase io: I/O controller tests...
phase procmem2: Processor vs. memory II tests...
phase ibexit: Centerplane connection tests...
phase final_config: Final configuration...
Configuring in 3F with 4 processors, 2 SBus cards, 1024 MBytes memory.
Interconnect frequency is  83.273 MHz, from SNMP MIB.
Processor    frequency is 166.589 MHz, from SNMP MIB.
Boot processor is 5.0 = 20
POST (level=16, verbose=20, -H4,0020) execution time 3:50
hpost is complete.
obp_helper -H -m20
Board debut complete.
Reconfiguring domain mask registers.
Probing board resources.
Board attachment initiated successfully.

Ready to COMPLETE board attachment.

dr>
```

DIAGNOSTICS   The following diagnostics are supported.

`add_board_to_domain returns entry not found`

  The target domain specified by the SUNW_HOSTNAME environment
  variable is not properly listed in the `domain_config`(4) file. Check the
  `domain_config`(4) file, then try the operation again at a later time.

`add_board_to_domain returns entry not found Unable to`
`locate domain` *target domain* `in domain_config file.`

  DR was unable to locate an entry for the current target domain. Use the
  `domain_status`(1M) command to verify the contents of the
  `domain_config`(4) file. Refer to the *Sun Enterprise 10000 SSP 3.5*
  Reference Manual.

`Board debut failed - return =` *value*

  The debut utility has failed (refer to the `obp_helper`(1M) man page in the
  Sun Enterprise 10000 SSP 3.5 Reference Manual). Consult the SSP message
  files for information regarding the failure.

`Board` *brd* **is a member of a foreign hardware domain.**

The board you are trying to attach has been identified as a member of
another domain on this platform, which prevents it from being attached to
the designated target domain. You must remove this board from the other
domain before initiating an attach.

Board *brd* is not eligible for attach

One or more conditions is preventing this board from being attached to the
target domain. The board must be physically present, powered on, and not a
member of any domain to be eligible for attachment.

Board may be black or red listed.

If this board is blacklisted or redlisted, it cannot be attached. Check the
postrc(1M) file for the location of the blacklist(1M) and redlist(4)
files.

DR Error: State for board *brd* cannot be determined.

During initial domain contact an unexpected board condition was detected
by dr_daemon(1M). (Refer to the dr_daemon(1M) man page in *man pages
section 1M: System Administration Commands*.) Check the system log on the
host for more information.

Error executing *command*

dr(1M) executed the indicated command, but it returned a failure
indication. If the error message specifies a specific action you must take, do
so, then retry the command. Otherwise, simply retry the init_attach
operation at a later time. If that attempt fails, call your service provider.

FAD error detected, retrying...

A transient failure occurred during updating of the domain_config(4) file
has been. init_attach will retry the operation. If all retries fail, consult
the SSP messages files for more information.

Failed to initiate board attachment

The init_attach operation on the target domain has failed.

Unable to execute *command*

dr(1M) could not execute the indicated command. Check that the program
file exists and is assigned the appropriate modes.

EXIT STATUS   If successful, init_attach returns a 0 in the dr_return global variable; if
                  not, it returns a 1, along with one or more diagnostic messages.

NOTES   If DR detects a usage syntax error, it immediately aborts the dr(1M) command,
                  displays the dr(1M) shell prompt, and leaves dr_return unmodified. See
                  dr(1M).

SEE ALSO   dr(1M) in this reference manual

                  blacklist(4), domain_config(4), domain_status(1M),
                  domain_switch(1M), postrc(4), redlist(4) in the *Sun Enterprise 10000
                  SSP 3.5 Reference Manual*

                  dr_daemon(1M) in the *man pages section 1M: System Administration
                  Commands*

NAME | moveboard – detach a board from its domain and attach it to a specific domain

SYNOPSIS | **moveboard** −b *board_number* −d *domain* [−f] [−r *retry_count* [−t *timeout*]] [−q]

moveboard −d *domain* [−f] [−r *retry_count* [−t *timeout*]] [−q] SB*x*

moveboard −h

DESCRIPTION | The moveboard command detaches a system board from its current domain (referred to as the source domain) and attaches it to the domain specified by *domain* (referred to as the target domain). It also writes progress messages to the platform log on the SSP and to standard output.

If moveboard fails, it writes an error message to standard error and returns a nonzero exit status. If the specified board is already detached and powered on, moveboard attempts to attach it to the specified domain. If the board is already in the specified domain, moveboard returns an exit status of zero, indicating success.

The −r option specifies the maximum number of retry attempts, *retry_count*, in the event that the moveboard command fails. The −t option specifies a time period in seconds, *timeout*, between retry attempts. If you specify the number of retries, but no timeout, there is no delay between retry attempts. If you specify a timeout, you must also specify the number of retries.

To move a board, it must be powered on and in one of the following states:

■ In a domain

■ In an intermediate state due to an incomplete DR operation

If the board is not already in the target domain or if it has already been detached from its source domain, moveboard proceeds with the attach operation.

OPTIONS | The following options are supported:

−b *board_number*   Specifies the system board number, 0 through 15.

−d *domain*   Specifies the domain name.

−f   Attempts to force the operation. The −f option will override certain software constraints, but it will never override fundamental safety and availability constraints of the hardware and Solaris operating environment.

−h   Displays a usage message.

−r *retry_count*	Specifies the number of retry attempts. There are no limits to the possible retries.
−t *timeout*	Specifies the time in seconds between retry attempts. There is no limit to the amount of time.
−q	Turns on quiet mode, which means informational messages are not written to standard output.

OPERANDS The following operand is supported:
SB*x*   Specifies the system board, where *x* equals an
                                    integer 0 through 15.

EXAMPLES In the following examples, the command moves system board 2 from its
current domain to a domain named xf3-b2. If the first attempt to move the
board fails, two retries are made, if necessary, with a wait time of 15 minutes
between retries.

EXAMPLE 1   To Move a Board With the −b Option

moveboard -b 2 -d xf3-b2 -r 2 -t 900

EXAMPLE 2   To Move a Board Without the −b Option

moveboard -d xf3-b2 -r 2 -t 900 SB2

EXIT STATUS The following exit values are supported.

0   Successfully moved the board to the specified domain.

1   Command syntax error occurred.

2   Failed to acquire DR lock.

3   Failed to release DR lock.

4   Failed to initialize DR communications.

5   Board or target domain is not eligible for detachment.

6   A board in the target domain is in an intermediate detach state.

7   DR daemon does not allow a detach operation when it removes the
         last processor from the domain or when it drops the memory in the
         domain below the minimum requirement.

8   Failed during the memory drain operation.

9   Failed to make progress during the drain operation.

10   Failed to complete the drain operation.

11   The board or target domain is not eligible for detachment.

12   A board in the target domain is in an intermediate attach state.

13   Failed in initial attachment stage.

14   Failed in complete attachment stage.

15   Failed to dynamically link with DR library.

17   A termination signal was received.

51   The command was not completed because of a negative
         acknowledgment from the `confp->confirm` function.

52   System configuration administration is not supported on this
         attachment point.

53   System configuration administration operation is not supported on this
         attachment point.

54   The caller does not have the required process privileges. For example,
         if configuration administration is performed through a device driver,
         the permissions on the device node are used to control access.

55   The command was not completed because an element of the system
         configuration admistration system was busy.

56   The command required a service interruption and was not completed
         because part of the system could not be suspended (that is, quiesced).

57   A procedural error occurred in the data.

58   A procedural error occurred in the library, including a failure to obtain
         process resources such as memory and file descriptors.

59   A hardware-specific library could not be located by using the specified
         attachment point.

60   The operation failed because of the condition of the attachment point.

61   The system configuration administration operation requested is not
        supported on the specified attachment point.

62   An error occurred during the processing of the requested operation.
        This error code includes validation of the command arguments by the
        hardware-specific code.

63   No such attachment point exists.

64   No attachment point with the specified attributes exists.

65   An invalid board ID was specified.

ERROR   If moveboard fails, it writes diagnostic messages to standard error.
MESSAGES

NOTES   The moveboard command prevents more than one attach or detach operation
                  from executing concurrently.

SEE ALSO   addboard(1M), deleteboard(1M), rcfgadm(1M)

                  domain_status(1M) and showdevices(1M) in the *Sun Enterprise 10000
                  SSP 3.5 Reference Manual*

                  dcs(1M) and dr_daemon(1M) in *man pages section 1M: System
                  Administration Commands* in the *Solaris 8 Reference Manual*

                  Sun Enterprise 10000 Dynamic Reconfiguration User Guide

NAME	rcfgadm – remote configuration administration
SYNOPSIS	**rcfgadm** −d *domain_name* [−v] [−a] [−s *listing_options*] [−o *hardware_options*] [−l [*ap_id*	*ap_type*]]
	rcfgadm −d *domain_name* [−v] [−o *hardware_options*] −h [*ap_id*	*ap_type*]
DESCRIPTION	The rcfgadm(1M) command provides remote configuration administration operations on dynamically reconfigurable hardware resources from the System Service Processor (SSP). These operations include displaying status, (−l), and obtaining configuration administration help messages (−h). They are performed on attachment points, which are places where system software supports dynamic reconfiguration of hardware resources during continued operation of the Solaris operating environment.

Configuration administration makes a distinction between hardware resources that are physically present in the server and hardware resources that are configured and visible to the Solaris operating environment. The nature of configuration administration functions are hardware-specific and are performed by calling hardware-specific libraries.

Attachment points have associated state and condition information. A receptacle can exist in one of three states: empty, disconnected, or connected, while an occupant can exist in one of two states: configured or unconfigured.

The empty state is the normal state of a receptacle when the attachment point contains no occupant. A receptacle can also be in the disconnected state if its occupant is isolated from normal system access. Typically this state is used for hardware-specific testing prior to bringing the resources into full use by the system or as a step in preparing an occupant for physical removal or reconfiguration. The connected state enables normal access to hardware resources on the occupant. The connected state is the normal state of a receptacle that contains an occupant and that is not currently undergoing configuration administration operations. In the unconfigured state, the hardware resources are not exposed in the Solaris software data structures; thus, they are not available for use by the Solaris operating environment. In the configured state, hardware resources are exposed in software data structures; thus, they can be in use by the Solaris operating environment.

An attachment point can be in one of five conditions: unknown, ok, failing, failed, or unusable. An attachment point can enter the system in any condition, depending upon results of power-on tests and nonvolatile record keeping. With a configured occupant, the attachment point can be in one of four conditions: unknown, ok, failing, failed. An attachment point can change to failing during the course of operation if a hardware-dependent recoverable error threshold is exceeded. An attachment point can change to

`failed` as a result of an unrecoverable error. With an `unconfigured` occupant, an attachment point can be in any of the defined conditions. The condition can change from `ok` to `unknown` after a system-dependent time threshold.

An attachment point can be in the `unusable` condition for a variety of reasons, such as inadequate power, inadequate cooling, or an occupant that is unidentifiable, unsupported, incorrectly configured. In the `unusable` condition, an attachment point cannot be used by the system. It typically remains in this condition until the physical cause is remedied.

The server also maintains information about in-progress state changes and condition re-evaluations. When they occur on an attachment point, they are represented as being `busy` in the `rcfgadm` output.

Attachment points are represented by hardware-specific identifiers (*ap_id*s) that are related to the type and location of the attachment points in the system device hierarchy. An *ap_id* cannot be ambiguous; it must identify a single attachment point. Two types of *ap_id* specifications are supported: physical and logical. A physical *ap_id* contains a fully specified path name, while a logical *ap_id* contains a shorthand notation that identifies an attachment point in a more user-friendly way.

For example, an attachment point representing slot number 7 could have a physical *ap_id* of `/devices/central/fhc/sysctrl:SB7` while the logical *ap_id* could be `SB7`. Another example, the third receptacle on the second PCI I/O bus on a system could have a logical *ap_id* of `pci2:plug3`.

EXAMPLE 1   Physical *ap_ids*

```
/devices/pci@71,2000/pci@2/SUNW,isptwo@4:scsi
/devices/pci@71,2000/pci@2/SUNW,isptwo@4:scsi::dsk/c0t0d0
/devices/pci@71,2000/pci@2/SUNW,isptwo@4:scsi::dsk/c0t1d0
/devices/pci@71,2000/pci@2/SUNW,isptwo@4:scsi::dsk/c0t2d0
/devices/pci@71,2000/pci@2/SUNW,isptwo@4:scsi::dsk/c0t3d0
/devices/pseudo/ngdr@0:SB0
/devices/pseudo/ngdr@0:SB1
/devices/pseudo/ngdr@0:SB2
/devices/pseudo/ngdr@0:SB3
```

Attachment points can also be created dynamically. A dynamic attachment point is named relative to a base attachment point which is present in the system. *ap_ids* for dynamic attachment points consist of a base component followed by two colons (::) and a dynamic component. The base component is the base attachment point *ap_id*. The dynamic component is hardware-specific and generated by the corresponding hardware-specific library.

EXAMPLE 2   Logical *ap_ids*

```
c0
c0::dsk/c0t0d0
c0::dsk/c0t1d0
c0::dsk/c0t2d0
c0::dsk/c0t3d0
c0::dsk/c0t4d0
c0::dsk/c0t5d0
c0::dsk/c0t6d0
SB0
SB1
SB2
SB3
```

For example, consider a base attachment point, which represents a SCSI HBA,
with the physical *ap_id*/devices/sbus@1f,0/SUNW,fas@e,8800000:scsi
and logical *ap_id* c0. A disk attached to this SCSI HBA could be represented
by a dynamic attachment point with logical *ap_id* c0::dsk/c0t0d0 where c0
is the base component and dsk/c0t0d0 is the hardware-specific dynamic
component. Similarly the physical *ap_id* for this dynamic attachment point
would be:

/devices/sbus@1f,0/SUNW,fas@e,8800000:scsi::dsk/c0t0d0.

An *ap_type* is a partial form of a logical *ap_id* that can be ambiguous and not
specify a particular attachment point. An *ap_type* is a substring of the portion
of the logical *ap_id*, up to but not including, the colon (:) separator. For
example, an *ap_type* of pci would show all attachment points whose logical
ap_ids begin with pci.

EXAMPLE 3   *ap_types*

```
c0
sbd
```

The use of *ap_type*s is discouraged. The new select suboption to the −s option
provides a more general and flexible mechanism for selecting attachment
points. See OPTIONS.

rcfgadm interacts primarily with hardware-dependent functions contained in
hardware-specific libraries and, thus, its behavior is hardware-dependent.
Hardware-specific options are supplied as suboptions using the −o option.

The arguments for this command conform to the getopt(3C) and
getsubopt(3C) syntax conventions.

OPTIONS   The following options are supported:

−a

Specifies that the −l option must also list dynamic attachment points.

−d *domain_name*

  Specifies the ID for a domain.

−h **[***ap_id* | *ap_type***]**

  Prints out the help message text. If *ap_id* or *ap_type* is specified, the help
  routine of the hardware-specific library for the attachment point indicated
  by the argument is called.

−l **[***ap_id* | *ap_type***]**

  Lists the state and condition of attachment points specified. Attachment
  points can be filtered by using the −s option and select suboption. Invoking
  rcfgadm without one of the action options is equivalent to −l without an
  argument. The format of the list display is controlled by the −v and −s
  options. When the −a option is specified, attachment points are dynamically
  expanded.

−o *hardware_options*

  Supplies hardware-specific options to the main command option. The
  format and content of the hardware option string is completely
  hardware-specific. The option string *hardware_options* conforms to the
  getsubopt(3C) syntax convention.

−s *listing_options*

Supplies listing options to the list (−l) command. *listing_options* conforms to the getsubopt(3C) syntax convention. The suboptions are used to specify the attachment point selection criteria (select=select_string), the type of matching desired (match=match_type), order of listing (sort=field_spec), the data that is displayed (cols=field_spec and cols2=field_spec), the column delimiter (delim=string) and whether to suppress column headings (noheadings).

When the select suboption is specified, only attachment points which match the specified criteria are listed. The select suboption has the following syntax:

rcfgadm −s select=attr1(value1):attr2(value2)...

where an attr is one of *ap_id*, *class* or *type*. *ap_id* refers to the logical *ap_id* field, *class* refers to attachment point class and *type* refers to the type field. *value1*, *value2*, and so on, are the corresponding values to be matched. The type of match can be specified by the match suboption as follows:

rcfgadm −s match=match_type,select=attr1(value1)...

where *match_type* can be either exact or partial. The default value is exact. Arguments to the select suboption can be quoted to protect them from the UNIX C shell interpretation.

A *field_spec* is one or more data-fields concatenated using a colon (:), as in data-field:data-field:data-field. A data-field is one of *ap_id*, *physid*, *r_state*, *o_state*, *condition*, *type*, *busy*, *status_time*, *status_time_p* and *info*. The *ap_id* field output is the logical name for the attachment point, while the *physid* field contains the physical name. The *r_state* field can be empty, disconnected, or connected. The *o_state* field can be configured or unconfigured. The *busy* field can be either y if the attachment point is *busy*, or n if it is not. The *type* and *info* fields are hardware-specific. The *status_time_p* field is a parsable version of the *status_time* field. If an attachment point has an associated class, the *class* field lists the class name.

The order of the fields in *field_spec* is significant. For the sort suboption, the first field given is the primary sort key. For the cols and cols2 suboptions, the fields are printed in the order requested. The order of sorting on a data-field can be reversed by placing a minus (-) before the data-field name within the *field_spec* for the sort suboption. The default value for sort is *ap_id*. The default values for cols and cols2 depend on whether the −v option is given: Without it, cols is ap_id:r_state:o_state:condition and cols2 is not set; with −v, cols is ap_id:r_state:o_state:condition:info and cols2 is status_time:type:busy:physid. The default value for delim is a single space. The value of delim can be a string of arbitrary length. The

delimiter cannot include a comma (,) character,; see getsubopt(3C). These listing options can be used to create parsable output. See NOTES.

−v

Executes in verbose mode. For the −c, −t, and −x options, outputs a message giving the results of each attempted operation. Outputs detailed help information for the −h option if there is any. Outputs verbose information for each attachment point for the −l option.

ENVIRONMENT VARIABLES

See environ(5) for descriptions of the following environment variables that affect the execution of command_name: LC_TIME,LC-MESSAGES,TZ.

LC_MESSAGES	Determines how rcfgadm displays column headings and error messages. Listing output data is not affected by the setting of this variable.
LC_TIME	Determines how rcfgadm displays human-readable status changed time (*status_time*).
TZ	Specifies the time zone used when converting the status changed time. This applies to both the human-readable (*status_time*) and parsable (*status_time_p*) formats.

EXIT STATUS

The following exit values are returned:

0	Successful completion.
1	An error occurred.
2	Configuration administration not supported on specified target.
3	Usage error.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWsspop

SEE ALSO

cfgadm(1M)

NOTES   Hardware resources enter the `unconfigured` pool in a hardware-specific
manner. This can occur at various times such as system initialization or as a
result of an `unconfigure` operation. An occupant that is in the
`unconfigured` state is not available for use by the system until specific
intervention occurs. This intervention can be manifested as an
operator-initiated command, or it can be by way of an automatic configuring
mechanism.

The listing option of the `rcfgadm` command can be used to provide parsable
input for another command, for example, within a shell script. For parsable
output, the −s option must be used to select the fields required. The −s option
can also be used to suppress the column headings. The following fields always
produce parsable output: *ap_id*, *physid*, *r_state*, *o_state*, *condition*, *busy*,
status_time_p, and *type*. Parsable output never has white-space characters
embedded in the field value.

The following shell script fragment finds the first good unconfigured occupant
of type CPU.

EXAMPLE 4   Finding the First Good Occupant of Type CPU

```
found=
    rcfgadm -l -s "noheadings,cols=ap_id:r_state:condition:type" | \
    while read ap_id r_state cond type
    do
            if [ "$r_state" = unconfigured -a "$cond" = ok -a "$type" = CPU ]
                    then
                    if [ -z "$found" ]
                            then
                            found=$ap_id
                            fi
                    fi
    done
    if [ -n "$found" ]
    then
                            echo "Found CPU $found"
    fi
```

The format of the parsable time field (*status_time_p*) is *YYYYMMDDhhmmss*,
giving the year, month, day, hour, minute, and second in a form suitable for
string comparison.

Reference should be made to the hardware-specific documentation for details
of system configuration administration support.

NAME	reconfig – initiate auto-configuration sequence
SYNOPSIS	**reconfig**
CAUTION	This command can remap device files and cause the renaming of known devices. Use it with extreme caution.
DESCRIPTION	Execute this command at the dr(1M) shell prompt after a new board has been attached to a running domain to make the devices immediately available for use.

Note - As of the Solaris 8 GA release, manual reconfiguration is not needed. A new DDI subsystem, devfsadm, completes all of the reconfiguration tasks.

reconfig executes the standard Solaris configuration sequence in the target domain. This sequence consists of the following commands, shown here in the proper order: drvconfig(1M), devlinks(1M), disks(1M), ports(1M), and tapes(1M).

Note - This command is available only for DR 2.0 domains (that is, domains that use the DR daemon) on the Sun Enterprise 10000 server. You can use domain_status(1M) to determine the DR version running on the domain.

EXAMPLES	**EXAMPLE 1** Using reconfig(1M)

```
dr> reconfig
Reconfiguration of devices in progress...
Reconfiguration completed successfully.
```

DIAGNOSTICS	The following diagnostics are supported.

```
Reconfiguration failed
```

One or more of the reconfiguration commands failed. Check the /var/adm/messages file on the domain.

EXIT STATUS	reconfig returns a 0 in the dr_return global variable upon success, or a 1 upon failure.
NOTES	If DR detects a usage syntax error, it immediately aborts the dr(1M) command, displays the dr(1M) shell prompt, and leaves dr_return unmodified. See dr(1M).
SEE ALSO	dr(1M) in this reference manual

drvconfig(1M), devlinks(1M), disks(1M), ports(1M), tapes(1M) in *man
pages section 1M: System Administration Commands*