



Sun Enterprise 10000 Dynamic Reconfiguration Reference Manual

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part Number 806-7617-10
October 2001, Revision A

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

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

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

Sun, Sun Microsystems, the Sun logo, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

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

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

Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, Californie 94303-4900 Etats-Unis. Tous droits réservés.

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

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

Sun, Sun Microsystems, le logo Sun, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

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

Sun Enterprise 10000 DR attributions:

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

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



Please
Recycle

This is scotty, a simple tcl interpreter with some special commands to get information about TCP/IP networks. Copyright (c) 1993, 1994, 1995, J. Schoenwaelder, TU Braunschweig, Germany, Institute for Operating Systems and Computer Networks. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that this copyright notice appears in all copies. The University of Braunschweig makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Contents

Intro(1m)	8
abort_attach(1m)	12
abort_detach(1m)	14
addboard(1M)	16
complete_attach(1m)	20
complete_detach(1m)	22
deleteboard(1M)	24
dr(1m)	28
dr.service(1m)	31
dr_cmd_a_attach(1m)	32
dr_cmd_a_detach(1m)	33
dr_cmd_auto_config(1m)	34
dr_cmd_c_attach(1m)	35
dr_cmd_c_detach(1m)	36
dr_cmd_c_f_detach(1m)	37
dr_cmd_cpu_info(1m)	38
dr_cmd_debug(1m)	39
dr_cmd_detach_allow(1m)	40
dr_cmd_dev_info(1m)	41

dr_cmd_drain(1m) 42
dr_cmd_drain_status(1m) 43
dr_cmd_eligible_attach(1m) 44
dr_cmd_eligible_detach(1m) 45
dr_cmd_init_attach(1m) 46
dr_cmd_mem_info(1m) 48
dr_cmd_obp_info(1m) 49
dr_cmd_print_brd_info(1m) 50
dr_cmd_print_obp_info(1m) 52
dr_cmd_print_unsafe_info(1m) 53
dr_cmd_unsafe_dev_info(1m) 54
drain(1m) 55
drshow(1m) 57
drview(1m) 59
init_attach(1m) 60
moveboard(1M) 64
rcfgadm(1M) 68
reconfig(1m) 75

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	<p>abort_attach(1M)</p> <p>abort DR attach operation</p> <p>abort_detach(1M)</p> <p>abort DR detach operation</p> <p>addboard(1M)</p> <p>add a system board to a domain</p> <p>complete_attach(1M)</p> <p>complete DR attach operation</p> <p>complete_detach(1M)</p> <p>complete DR detach operation</p> <p>deleteboard(1M)</p> <p>delete a system board from a domain</p> <p>dr(1M)</p> <p>initiate dynamic reconfiguration shell</p> <p>dr.service(1M)</p> <p>abort DR attach system board operation</p> <p>dr_cmd_a_attach(1M)</p>

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 <i>sb</i>
DESCRIPTION	<p>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 <i>Sun Enterprise 10000 SSP Reference Manual</i>.</p> <p>You should run abort_attach after init_attach(1M) has successfully completed, and instead of the complete_attach(1M) command.</p> <p>If executing abort_attach fails to abort the operation, try repeating the attempt at a later time, or contact your service provider.</p> <hr/> <p>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.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board not to be attached</p>
EXAMPLES	<p>EXAMPLE 1 Using abort_attach(1M)</p> <pre>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></pre>
DIAGNOSTICS	<p>The following diagnostics are supported:</p> <p>Failed to abort board attachment</p> <p>Repeat the abort_attach command at a later time, or contact your service provider.</p>

EXIT STATUS	If successful, <code>abort_attach</code> returns a 0 in the <code>dr_return</code> 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 <code>dr(1M)</code> command, displays the <code>dr(1M)</code> shell prompt, and leaves <code>dr_return</code> unmodified. See <code>dr(1M)</code> .
SEE ALSO	<code>dr(1M)</code> , <code>init_attach(1M)</code>

NAME	abort_detach – abort a DR Detach operation
SYNOPSIS	abort_detach <i>sb</i>
DESCRIPTION	Execute this command at the dr(1M) shell prompt to abort an attempt to DR Detach a board. You can execute <code>abort_detach</code> after the board has been successfully executed, resources on the designated system board are once again available to the operating system.
	<hr/> 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 <code>domain_status(1M)</code> to determine the DR version running on the domain. <hr/>
OPERANDS	The following operand is supported. <i>sb</i> The board number (0 to 15) of the system board not to be detached
EXAMPLES	EXAMPLE 1 Using <code>abort_detach(1M)</code> <pre>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.</pre>
DIAGNOSTICS	The following diagnostics are supported: <pre>FAILED to restore domain_config file Retry the ABORT board detach at a later time</pre> <p>The attempt to restore the board number to the target domain board list in the <code>domain_config(4)</code> file (refer to the man page in the <i>Sun Enterprise 10000 SSP 3.5 Reference Manual</i>) has failed. This may be a temporary condition, so try the <code>abort_detach</code> again at a later time.</p> <pre>Failed to abort board detach</pre> <p>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 <code>abort_detach</code> again at a later time.</p>
EXIT STATUS	If successful, <code>abort_detach</code> returns a 0 in the <code>dr_return</code> 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	<p>addboard -d <i>domain</i> [-f] [-r <i>retry_count</i> [-t <i>timeout</i>]] [-q] -b <i>board_number</i></p> <p>addboard -d <i>domain</i> [-f] [-r <i>retry_count</i> [-t <i>timeout</i>]] [-q] SBx</p> <p>addboard -h</p>
DESCRIPTION	<p>The addboard(1M) command attaches a system board to a domain specified by <i>domain</i> (also referred to as the target domain). It also writes progress messages to the platform log on the SSP and to standard output.</p> <p>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.</p> <p>The -r option specifies the maximum number of retry attempts, <i>retry_count</i>, in the event that an attach operation fails. The -t option specifies a time period in seconds, <i>timeout</i>, 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.</p> <p>To be eligible for attachment, a board must be powered on and in one of the following states:</p> <ul style="list-style-type: none"> ■ Not in a domain ■ In an intermediate state due to an incomplete DR operation <p>If the specified board is already in the domain, addboard returns an exit status of zero, indicating success.</p>
OPTIONS	<p>The following options are supported:</p> <p>-b <i>board_number</i> Specifies the system board number, 0 through 15.</p> <p>-d <i>domain</i> Specifies the domain name.</p> <p>-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.</p> <p>-h Displays a usage message.</p> <p>-r <i>retry_count</i> Specifies the number of retry attempts. There are no limits to the possible retries.</p>

`-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
MESSAGES**

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

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 <i>sb</i>
DESCRIPTION	<p>Execute this command at the <code>dr(1M)</code> shell prompt to complete an attempt to DR Attach a board after successful execution of the <code>init_attach(1M)</code> command. <code>complete_attach</code> 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 <code>drshow(1M)</code> command to display the current system configuration for the board.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board to be attached to the target domain.</p>
EXAMPLES	<p>EXAMPLE 1 Using <code>complete_detach(1M)</code></p> <pre>dr> complete_attach 5 Completing attach for board 5 Board attachment completed successfully.</pre>
DIAGNOSTICS	<p>The following diagnostics are supported:</p> <p>Failed during final state transition</p> <p>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 <code>complete_attach</code> operation or execute an <code>abort_attach(1M)</code>.</p> <p>Failed to complete attach board</p> <p>The operating system on the target domain was unable to attach the board. Repeat the <code>complete_attach</code> operation at a later time or execute the <code>abort_attach(1M)</code> command.</p>
EXIT STATUS	<p>If successful, <code>complete_attach</code> returns a 0 in the <code>dr_return</code> global variable; if not, it returns a 1, along with one or more diagnostic messages.</p>

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 <i>sb</i> [force]
DESCRIPTION	<p>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.</p> <p>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.</p> <p>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 <i>Sun Enterprise 10000 Dynamic Reconfiguration User Guide</i> for a description of such conditions. You can use the force argument to force the quiesce in such situations.</p> <hr/> <p>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.</p> <hr/>
OPERANDS	<p>The following operands are supported.</p> <p>sb The board number (0 to 15) of the system board to be detached</p> <p>force Force the domain quiesce operation. Refer to the <i>Sun Enterprise 10000 Dynamic Reconfiguration User Guide</i> for a description of such conditions. You can use the force argument to force the quiesce in such situations.</p>
EXAMPLES	<p>EXAMPLE 1 Using complete_detach(1M)</p> <pre>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.</pre>
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	<p>deleteboard <i>-b board_number</i> [<i>-f</i>] [<i>-r retry_count</i>] [<i>-t timeout</i>] [<i>-q</i>]</p> <p>deleteboard [<i>-f</i>] [<i>-r retry_count</i>] [<i>-t timeout</i>] [<i>-q</i>] SB<i>x</i></p> <p>deleteboard <i>-h</i></p>
DESCRIPTION	<p>The <code>deleteboard</code> 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.</p> <p>If <code>deleteboard</code> 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 <code>deleteboard</code>, or you can use <code>addboard(1M)</code> to return the board to its original state.</p> <p>The <code>-r</code> option specifies the maximum number of retry attempts, <i>retry_count</i>, in the event that a detach operation fails. The <code>-t</code> option specifies a time period in seconds, <i>timeout</i>, 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.</p> <p>To be eligible for detachment, a board must be powered on and in one of the following states:</p> <ul style="list-style-type: none"> ■ In a domain ■ In an intermediate state due to an incomplete DR operation <p>If the specified board is not in a domain or has already been detached from its source domain, <code>deleteboard</code> returns an exit value of zero, indicating success. A DR platform error message indicates that the board is already detached.</p>
OPTIONS	<p>The following options are supported:</p> <p><i>-b board_number</i> Specifies the system board number, 0 through 15.</p> <p><i>-f</i> Attempts to force the operation. The <i>-f</i> option will override certain software constraints, but it will never override fundamental safety and availability constraints of the hardware and Solaris operating environment.</p> <p><i>-h</i> Displays a usage message.</p> <p><i>-r retry_count</i> Specifies the number of retry attempts. There are no limits to the possible retries.</p>

- `-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	<p>The <code>dr</code> command initiates the Dynamic Reconfiguration (DR) shell, a Tcl application (see NOTES, below) with DR command extensions. You can use the <code>dr</code> 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.</p> <hr/> <p>Note - Whenever possible, use the DR GUI in Hostview to execute Dynamic Reconfiguration operations. Use the <code>dr</code> shell when you cannot run Hostview; for example, if you need to run DR over a dial-up connection. For more information, see the <i>Sun Enterprise 10000 Dynamic Reconfiguration User Guide</i> and <code>hostview(1M)</code> in the <i>Sun Enterprise 10000 SSP 3.5 Reference Manual</i>.</p> <hr/> <p>When executed on the command line, <code>dr</code> connects to the domain specified by the <code>SUNW_HOSTNAME</code> environment variable. After this connection is established, <code>dr</code> displays the <code>dr></code> prompt, which accepts the DR commands.</p> <p>To see the list of DR commands if you not using AnswerBook2, execute <code>man Intro</code> on the SSP while logged in as user <code>ssp</code>.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/> <p>You can quit the <code>dr</code> shell at any time by typing <code>exit</code> or <code>Control-d</code>.</p> <p>Caution: Do not execute any of the DR commands that begin with <code>dr_cmd_</code>; these are low-level commands that are for use only by authorized service personnel under special circumstances, as described in <code>dr.service</code>.</p> <p>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.</p> <p>The DR commands return error status in the global Tcl variable <code>dr_return</code>. 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 <code>set dr_return</code> to display <code>dr_return</code> after executing each DR command, to determine command success or failure. Though, under most circumstances, the diagnostic messages output by the <code>dr</code> shell clearly indicate success or failure.</p> <hr/> <p>Note - Type <code>help</code> at the <code>dr</code> shell prompt (<code>dr></code>) to access DR quick-reference help guide.</p> <hr/>

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 for 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	<p>The low-level commands described here, which begin with <code>dr_cmd</code>, 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.</p> <p>The DR shell provides commands that directly map to <code>libdr.so</code> 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.</p> <p>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 <code>hostview(1M)</code> in <i>Sun Enterprise 10000 SSP 3.5 Reference Manual</i>) is the preferred method of performing DR operations. Use the <code>dr(1M)</code> shell when the GUI-based Hostview application is unavailable.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
CAUTION	<p>Customers should not use these low-level commands, but should access DR through the DR GUI, as described in the <i>Sun Enterprise 10000 Dynamic Reconfiguration User Guide</i>, or by using the high-level DR commands (those without the <code>dr_cmd</code> prefix) described in this reference manual.</p>
SHELL COMMANDS	<p>The low-level shell commands are those that begin with <code>dr_cmd</code>. See <code>Intro(1M)</code>.</p>
EXIT STATUS	<p>The DR shell low-level command set generally returns an exit code in the <code>dr_return</code> global variable. Upon return from each of the DR commands, this variable can be tested for success or failure.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>

NAME	dr_cmd_a_attach – abort DR attach system board operation
SYNOPSIS	dr_cmd_a_attach <i>sb</i>
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness. Instead, use <code>abort_attach(1M)</code> , which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	<p>If <code>abort_attach(1M)</code> were unavailable for some reason, you could run <code>dr_cmd_init_attach(1M)</code> and before the board has been completely attached by using the <code>dr_cmd_c_attach(1M)</code>. <code>dr_cmd_a_attach</code> returns the board to the state it was in prior to the <code>dr_cmd_init_attach(1M)</code> operation; that is, present, powered-on, and in no domain.</p> <p><code>dr_cmd_a_attach</code> instructs the operating system running on the target domain to abandon the in-progress attach operation, removes the system board from the <code>domain_config</code> file, and resets the shared memory mask registers and board domain mask registers on the centerplane.</p> <p>Some conditions that are transparent to the user may cause an abort failure. Therefore, if <code>dr_cmd_a_attach</code> fails to complete the abort successfully, try executing it again at a later time.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The system board number (0 to 15) for the abort attach operation</p>
DIAGNOSTICS	See DIAGNOSTICS on <code>abort_attach(1M)</code> .
EXIT STATUS	<p>If <code>abort_attach(1M)</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>
SEE ALSO	<code>dr(1M)</code> , <code>dr_cmd_init_attach(1M)</code> , <code>dr_cmd_c_attach(1M)</code>

NAME	dr_cmd_a_detach – abort DR detach system board operation
SYNOPSIS	dr_cmd_a_detach <i>sb</i>
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness. Instead, use <code>abort_attach(1M)</code> , which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	<p>You can run <code>dr_cmd_a_detach</code> after draining a system board by using <code>dr_cmd_drain(1M)</code> , but before that board has been completely detached.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported:</p> <p><i>sb</i> The board number (0 to 15) of the system board whose detach is being aborted.</p>
DIAGNOSTICS	See DIAGNOSTICS in <code>abort_detach(1M)</code> .
EXIT STATUS	<p>If <code>dr_cmd_a_detach</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>
SEE ALSO	<code>dr(1M)</code> , <code>dr_cmd_drain(1M)</code>

NAME	dr_cmd_auto_config - run Solaris reconfig sequence on target domain
SYNOPSIS	dr_cmd_auto_config
CAUTION	Do not use this command, use <code>reconfig(1M)</code> instead. Only authorized service providers should use <code>dr_cmd_auto_config</code> , which runs in the DR shell, and only when they cannot use <code>reconfig(1M)</code> . Performing this operation may cause device files to be remapped and known devices to be renamed.
	<hr/> Note - As of the Solaris 8 GA release, manual reconfiguration is not needed. A new DDI subsystem, <code>devfsadm</code> , completes all of the reconfiguration tasks. <hr/>
DESCRIPTION	The system administrator would normally run <code>dr_cmd_auto_config</code> 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: <code>drvconfig(1M)</code> , <code>devlinks(1M)</code> , <code>disks(1M)</code> , and <code>tapes(1M)</code> . <hr/> 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 <code>domain_status(1M)</code> to determine the DR version running on the domain. <hr/>
DIAGNOSTICS	See DIAGNOSTICS in the <code>reconfig(1M)</code> man page.
EXIT STATUS	If <code>dr_cmd_auto_config</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1. <hr/> Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes. <hr/>
SEE ALSO	<code>reconfig(1M)</code> in this reference manual <code>drvconfig(1M)</code> , <code>devlinks(1M)</code> , <code>disks(1M)</code> , <code>dr_daemon(1M)</code> , <code>ports(1M)</code> , <code>tapes(1M)</code> in <i>man pages section 1M: System Administration Commands</i>

NAME	dr_cmd_c_attach – complete DR attach system board operation
SYNOPSIS	dr_cmd_c_attach <i>sb</i>
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness, and is dangerous. Instead, use <code>complete_attach(1M)</code> , which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	<p>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.</p> <hr/> <p>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.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board being attached</p>
DIAGNOSTICS	See DIAGNOSTICS on the <code>complete_attach(1M)</code> man page.
EXIT STATUS	<p>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.</p> <hr/> <p>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.</p> <hr/>
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 <i>sb</i>
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness, and is dangerous. Instead, use <code>complete_detach(1M)</code> , which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	<p><code>dr_cmd_c_detach</code> completes a DR detach board operation. The designated system board should already have been drained via <code>dr_cmd_drain(1M)</code>.</p> <p>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 <code>drshow(1M)</code> to list the devices in use on a given system board.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board being detached</p>
DIAGNOSTICS	See DIAGNOSTICS on <code>complete_detach(1M)</code> .
EXIT STATUS	<p>If <code>dr_cmd_c_attach</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>
SEE ALSO	<code>dr(1M)</code> , <code>dr_cmd_drain(1M)</code>

NAME	dr_cmd_c_f_detach – force completion of DR detach system board operation
SYNOPSIS	dr_cmd_c_f_detach <i>sb</i>
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness, and is dangerous. Instead, use <code>complete_detach(1M)</code> , which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	<p><code>dr_cmd_c_f_detach</code> 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 <i>Sun Enterprise 10000 Dynamic Reconfiguration User Guide</i> for more information about system quiesce and ways to increase the safety of this dangerous command.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board to be detached</p>
DIAGNOSTICS	See DIAGNOSTICS on the <code>complete_detach(1M)</code> man page.
EXIT STATUS	<p>If <code>dr_cmd_c_f_detach</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>
SEE ALSO	<code>dr(1M)</code> , <code>dr_cmd_drain(1M)</code>

NAME	dr_cmd_cpu_info – show processors on a system board in Tcl encoding
SYNOPSIS	dr_cmd_cpu_info <i>sb</i>
CAUTION	This command, which runs in the DR shell, produces output in a form suitable for the <code>drview(1M)</code> application, not the interactive user.
DESCRIPTION	<p><code>dr_cmd_cpu_info</code> 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 <code>drview(1M)</code> application.</p> <p>Since the Tcl list is not readily accessible to an interactive user, you should use <code>drshow(1M)</code> instead to acquire processor information.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the target system board</p>
EXIT STATUS	<p>If <code>dr_cmd_cpu_info</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>
SEE ALSO	<code>dr(1M)</code> , <code>dr_cmd_mem_info(1M)</code> , <code>dr_cmd_dev_info(1M)</code>

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	<p>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.</p> <p>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.</p> <hr/> <p>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.</p> <hr/>
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 <i>sb</i>
CAUTION	Only authorized service providers should use this command, which runs in the DR shell.
DESCRIPTION	<p>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.</p> <hr/> <p>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.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board to be queried</p>
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 <i>sb</i>
DESCRIPTION	<hr/> <p>Note - This command, which runs in the DR shell, produces output in a form suitable for the <code>drview(1M)</code> application, not the interactive user. Use <code>drshow(1M)</code> instead to view device information.</p> <hr/> <p><code>dr_cmd_dev_info</code> 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 <code>drview(1M)</code> application.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the target system board</p>
EXIT STATUS	<p>If <code>dr_cmd_dev_info</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>
SEE ALSO	<code>dr(1M)</code> , <code>dr_cmd_cpu_info(1M)</code> , <code>dr_cmd_mem_info(1M)</code>

NAME	dr_cmd_drain – start memory drain on a system board.
SYNOPSIS	dr_cmd_drain <i>sb</i>
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 <code>drain(1M)</code> command, which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	<p><code>dr_cmd_drain</code> 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 <code>dr_cmd_drain</code> command is invoked, the request is denied. If the unavailability is due to run-time conditions and system load, you should retry the <code>dr_cmd_drain</code> operation at a later time.</p> <p>The <code>dr_cmd_drain</code> operation also removes the system board from the board list in the <code>domain_config(4)</code> file on the SSP. (Refer to the <code>domain_config(4)</code> man page in the <i>Sun Enterprise 10000 SSP 3.5 Reference Manual</i>.)</p> <p><code>dr_cmd_drain</code> begins execution, then quickly exits. Use <code>drshow(1M)</code> to monitor its progress.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board to be drained</p>
DIAGNOSTICS	See DIAGNOSTICS on <code>drain(1M)</code> .
EXIT STATUS	<p>If <code>dr_cmd_drain</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>
SEE ALSO	<code>dr_cmd_mem_info(1M)</code>

NAME	dr_cmd_drain_status – show state of in-progress memory drain.
SYNOPSIS	dr_cmd_drain_status <i>sb</i>
CAUTION	Only authorized service providers should use this command, which runs in the DR shell.
DESCRIPTION	<p>Use <code>dr_cmd_drain_status</code> 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.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board being drained</p>
EXIT STATUS	<p>If <code>dr_cmd_drain_status</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>
SEE ALSO	<code>dr(1M)</code>

NAME	dr_cmd_eligible_attach – verify a system board is eligible for DR attach
SYNOPSIS	dr_cmd_eligible_attach <i>sb</i>
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. <hr/> 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. <hr/>
OPERANDS	The following operand is supported. <i>sb</i> 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. <i>y</i> The specified system board is eligible to be attached. <i>n</i> The specified system board is not eligible to be attached. dr_cmd_eligible_attach sends additional information to stdout. <i>board_number</i> The specified system board is not eligible to be attached because system board <i>board_number</i> (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). <hr/> 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. <hr/>
SEE ALSO	dr(1M)

NAME	dr_cmd_eligible_detach – verify a system board is eligible for DR detach
SYNOPSIS	dr_cmd_eligible_detach <i>sb</i>
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 <code>dr_cmd_eligible_detach</code> to verify that a system board is eligible for a detach operation before using <code>dr_cmd_drain(1M)</code> to begin a DR drain operation. <hr/> 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 <code>domain_status(1M)</code> to determine the DR version running on the domain. <hr/>
OPERANDS	The following operand is supported. <i>sb</i> The board number (0 to 15) of the system board to be checked
EXIT STATUS	<code>dr_cmd_eligible_detach</code> returns one of the following result codes to the <code>dr_return</code> global Tcl variable. Y The specified system board is eligible to be detached. n The specified system board is not eligible to be detached. <code>dr_cmd_eligible_detach</code> sends additional information to stdout. <i>board_number</i> The specified system board is not eligible to be detached because system board <i>board_number</i> (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). <hr/> Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes. <hr/>
SEE ALSO	<code>dr(1M)</code>

NAME	dr_cmd_init_attach – initiate DR attach system board operation
SYNOPSIS	dr_cmd_init_attach <i>sb</i>
CAUTION	Do not use this command, which runs in the DR shell; it is dangerous, and is included here only for completeness. Instead, use <code>init_attach(1M)</code> , which performs the same functions, but with the added security of safeguards and checks.
DESCRIPTION	<p><code>dr_cmd_init_attach</code> begins a DR attach board operation. DR does not screen the target domain for intermediate system boards as it does with the <code>init_attach(1M)</code> command and through Hostview.</p> <p><code>dr_cmd_init_attach</code> 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. <code>dr_cmd_init_attach</code> diagnoses, then debuts the system board to the target domain specified in the <code>SUNW_HOSTNAME</code> environment variable.</p> <p><code>dr_cmd_init_attach</code> adds the system board to the system board list in the <code>SSP domain_config</code> file. (Refer to the <code>domain_config(4)</code> man page in the <i>Sun Enterprise 10000 SSP 3.5 Reference Manual</i>.) 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.</p> <p>After <code>dr_cmd_init_attach</code> completes successfully, you can execute <code>dr_cmd_c_attach(1M)</code> to complete the attach operation, or <code>dr_cmd_a_attach(1M)</code> to abort it.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board to be attached</p>
DIAGNOSTICS	See DIAGNOSTICS on the <code>init_attach(1M)</code> man page.
EXIT STATUS	If <code>dr_cmd_init_attach</code> succeeds it returns a 0 result code in the <code>dr_return</code> 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 <i>sb</i>
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	<p>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.</p> <hr/> <p>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.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board to be checked</p>
EXIT STATUS	<p>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.</p> <hr/> <p>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.</p> <hr/>
SEE ALSO	dr(1M)

NAME	dr_cmd_obp_info – show complete config of a system board in Tcl encoding
SYNOPSIS	dr_cmd_obp_info <i>sb</i>
CAUTION	Do not use this command, which runs in the DR shell; it displays information in Tcl encoding, which is understood by the <code>drview(1M)</code> application, but is not intended for direct viewing by the interactive user. Instead, use <code>drshow(1M)</code> to view this information.
DESCRIPTION	<p><code>dr_cmd_obp_info</code> 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.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the target system board</p>
EXIT STATUS	<p>If <code>dr_cmd_obp_info</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>

NAME	dr_cmd_print_brd_info – show board resource in tabular format																					
SYNOPSIS	dr_cmd_print_brd_info <i>sb flags</i>																					
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness. Instead, use <code>drshow(1M)</code> , which presents the information in a more readable format.																					
DESCRIPTION	<p><code>dr_cmd_print_brd_info</code> obtains configuration information about the specified attached system board. The <i>flags</i> option specifies the information this command is to display, in the form of a bitstring, as follows:</p> <table border="0"> <thead> <tr> <th>Flag</th> <th>Value</th> <th>Display</th> </tr> </thead> <tbody> <tr> <td>-----</td> <td>-----</td> <td></td> </tr> <tr> <td></td> <td>1</td> <td>Processor information</td> </tr> <tr> <td></td> <td>2</td> <td>Controller and peripheral information</td> </tr> <tr> <td></td> <td>4</td> <td>Memory configuration</td> </tr> <tr> <td></td> <td>8</td> <td>Memory cost information</td> </tr> <tr> <td></td> <td>16</td> <td>Memory drain status</td> </tr> </tbody> </table> <p>You can obtain multiple displays by OR'ing (summing) the above decimal values. All displays are in a readable, tabular format.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>	Flag	Value	Display	-----	-----			1	Processor information		2	Controller and peripheral information		4	Memory configuration		8	Memory cost information		16	Memory drain status
Flag	Value	Display																				
-----	-----																					
	1	Processor information																				
	2	Controller and peripheral information																				
	4	Memory configuration																				
	8	Memory cost information																				
	16	Memory drain status																				
OPERANDS	<p>The following operands are supported.</p> <p><i>sb</i> The board number (0 to 15) of the target system board</p> <p><i>flags</i> A bitstring in decimal that represents the desired information</p>																					
EXAMPLES	<p>EXAMPLE 1 Displaying the Processor and Memory Configuration</p> <p>To display the processor and memory configuration, use the following command.</p> <pre>dr> dr_cmd_print_brd_info 5</pre>																					

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 <i>sb</i>
CAUTION	Do not use this command, which runs in the DR shell; it is included here only for completeness. Instead, use the <code>drshow(1M)</code> command, which presents the information in a more readable format.
DESCRIPTION	<p><code>dr_cmd_print_obp_info</code> 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.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the target system board</p>
EXIT STATUS	<p>If <code>dr_cmd_print_obp_info</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>

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 <code>drshow(1M)</code> , which presents the information in a more readable format.
DESCRIPTION	<p><code>dr_cmd_print_unsafe_info</code> queries the target domain to determine if any unsafe peripheral devices are open. (Refer to the <i>Sun Enterprise 10000 Dynamic Reconfiguration User Guide</i> for more information concerning DR unsafe devices.) If it finds that any such devices are open, it sends that information to <code>stout</code>.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/> <p>If <code>dr_cmd_print_unsafe_info</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>

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 <code>drshow(1M)</code> , which presents the information in a more readable format.
DESCRIPTION	<p><code>dr_cmd_unsafe_dev_info</code> queries the target domain to determine if any unsafe peripheral devices are open. (Refer to the <i>Sun Enterprise 10000 Dynamic Reconfiguration User Guide</i> 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 <code>drview(1M)</code> application.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/> <p>If <code>dr_cmd_unsafe_dev_info</code> succeeds it returns a 0 result code in the <code>dr_return</code> global variable. If it fails, it returns a 1 and displays diagnostic messages.</p> <hr/> <p>Note - Tcl parsing errors prevent DR commands from running which, in turn, leaves <code>dr_return</code> uninitialized. In such cases, the <code>dr_return</code> error code is meaningless. See <code>dr(1M)</code> for more information concerning return codes.</p> <hr/>

NAME	drain – start memory drain
SYNOPSIS	drain <i>sb</i> [wait]
DESCRIPTION	<p>The <code>drain</code> command, which you execute from the <code>dr(1M)</code> prompt, is the first of a two-step procedure for DR detaching a system board. The primary function of the <code>drain</code> 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.</p> <p>If a suitable target memory for the memory copy is not available when the <code>drain</code> 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.</p> <p>The <code>drain</code> command starts the drain operation, and then returns. The drain may take several minutes to complete. You can execute <code>drshow sb DRAIN</code> to monitor its progress (see the <code>drshow(1M)</code> man page). Or, you can specify the <code>wait</code> option, and the <code>drain</code> returns only after the board has been fully drained, or <code>drain</code> detects an error. <code>drain</code> automatically displays the board status once before returning.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board to be drained</p> <p><i>wait</i> 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.</p>
EXAMPLES	<p>EXAMPLE 1 Using <code>drain(1M)</code></p> <pre>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.</pre>

```

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	<p>drshow UNSAFE [<i>interval</i> <i>count</i>]</p> <p>drshow <i>sb</i> [<i>report_type</i>][<i>interval</i> <i>count</i>]</p> <p>drshow ALL [<i>report_type</i>][<i>interval</i> <i>count</i>]</p>
DESCRIPTION	<p>drshow displays board-level and system-level resources and information about DR. It presents the displays in a tabular format.</p> <p>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.</p> <hr/> <p>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.</p> <hr/>
OPERANDS	<p>The following operands are supported.</p> <p>UNSAFE Displays all unsafe devices that are open throughout the domain.</p> <p><i>sb</i> The board number (0 to 15) of the target domain</p> <p>ALL Reports the requested information for all active system boards in the domain.</p> <p><i>report_type</i> Specifies a specific report type. All of the following keywords for report types are case-insensitive.</p> <p> CPU – Displays processor information for the board (default)</p> <p> DRAIN – Displays the progress of any active drain operation</p> <p> IO – Shows the devices attached to this board</p> <p> 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.</p> <p> MEM – Displays the memory configuration of this board</p>

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          0      qe0                                ts4 (129:153:49:118)
qe1          0      qe1
qe2          0      qe2
qe3          0      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	<p>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 <code>hostview(1M)</code> man page in the <i>Sun Enterprise 10000 SSP 3.5 Reference Manual</i>.</p> <p>For more information about Hostview refer to the <i>Sun Enterprise 10000 SSP 3.5 User Guide</i>, and for more information about drview, refer to the <i>Sun Enterprise 10000 Dynamic Reconfiguration User Guide</i>.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
SEE ALSO	<p><code>hostview(1M)</code> in the <i>Sun Enterprise 10000 SSP 3.5 Reference Manual</i> <i>Sun Enterprise 10000 SSP 3.5 User Guide</i></p>

NAME	init_attach – initiate a DR attach operation
SYNOPSIS	init_attach <i>sb</i>
DESCRIPTION	<p>Execute this command at the <code>dr(1M)</code> 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 <code>SUNW_HOSTNAME</code> environment variable. Upon completion of the <code>init_attach</code>, 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 <code>domain_config(4)</code> file, and the centerplane is reconfigured such that the board is visible to the target domain.</p> <p>Upon successful completion of <code>init_attach</code> you can use <code>complete_attach(1M)</code> to complete the attach operation or <code>abort_attach(1M)</code> to abort it.</p> <hr/> <p>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 <code>domain_status(1M)</code> to determine the DR version running on the domain.</p> <hr/>
OPERANDS	<p>The following operand is supported.</p> <p><i>sb</i> The board number (0 to 15) of the system board to be attached</p>
EXAMPLES	<p>EXAMPLE 1 Using <code>init_attach(1M)</code></p> <pre>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.</pre> <p><code>dr> init_attach 5</code> Initiate attaching board 5 to domain ts4.. Adding board 5 to domain_config file. /opt/SUNWssp/bin/hpost -H20,4 Opening SNMP server library...</p> <pre>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...</pre>

```

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, <code>init_attach</code> returns a 0 in the <code>dr_return</code> 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 <code>dr(1M)</code> command, displays the <code>dr(1M)</code> shell prompt, and leaves <code>dr_return</code> unmodified. See <code>dr(1M)</code> .
SEE ALSO	<code>dr(1M)</code> in this reference manual <code>blacklist(4)</code> , <code>domain_config(4)</code> , <code>domain_status(1M)</code> , <code>domain_switch(1M)</code> , <code>postrc(4)</code> , <code>redlist(4)</code> in the <i>Sun Enterprise 10000 SSP 3.5 Reference Manual</i> <code>dr_daemon(1M)</code> in the <i>man pages section 1M: System Administration Commands</i>

NAME	moveboard – detach a board from its domain and attach it to a specific domain
SYNOPSIS	<p>moveboard <i>-b board_number -d domain [-f] [-r retry_count [-t timeout]] [-q]</i></p> <p>moveboard <i>-d domain [-f] [-r retry_count [-t timeout]] [-q] SBx</i></p> <p>moveboard <i>-h</i></p>
DESCRIPTION	<p>The <code>moveboard</code> command detaches a system board from its current domain (referred to as the source domain) and attaches it to the domain specified by <i>domain</i> (referred to as the target domain). It also writes progress messages to the platform log on the SSP and to standard output.</p> <p>If <code>moveboard</code> 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, <code>moveboard</code> attempts to attach it to the specified domain. If the board is already in the specified domain, <code>moveboard</code> returns an exit status of zero, indicating success.</p> <p>The <code>-r</code> option specifies the maximum number of retry attempts, <i>retry_count</i>, in the event that the <code>moveboard</code> command fails. The <code>-t</code> option specifies a time period in seconds, <i>timeout</i>, 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.</p> <p>To move a board, it must be powered on and in one of the following states:</p> <ul style="list-style-type: none"> ■ In a domain ■ In an intermediate state due to an incomplete DR operation <p>If the board is not already in the target domain or if it has already been detached from its source domain, <code>moveboard</code> proceeds with the attach operation.</p>
OPTIONS	<p>The following options are supported:</p> <p><i>-b board_number</i> Specifies the system board number, 0 through 15.</p> <p><i>-d domain</i> Specifies the domain name.</p> <p><i>-f</i> Attempts to force the operation. The <code>-f</code> option will never override fundamental safety and availability constraints of the hardware and Solaris operating environment.</p> <p><i>-h</i> Displays a usage message.</p>

<code>-r <i>retry_count</i></code>	Specifies the number of retry attempts. There are no limits to the possible retries.
<code>-t <i>timeout</i></code>	Specifies the time in seconds between retry attempts. There is no limit to the amount of time.
<code>-q</code>	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 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 `moveboard` fails, it writes diagnostic messages to standard error.

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	<pre>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]</pre>
DESCRIPTION	<p>The <code>rcfgadm(1M)</code> command provides remote configuration administration operations on dynamically reconfigurable hardware resources from the System Service Processor (SSP). These operations include displaying status, (<code>-l</code>), and obtaining configuration administration help messages (<code>-h</code>). 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.</p> <p>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.</p> <p>Attachment points have associated state and condition information. A receptacle can exist in one of three states: <code>empty</code>, <code>disconnected</code>, or <code>connected</code>, while an occupant can exist in one of two states: <code>configured</code> or <code>unconfigured</code>.</p> <p>The <code>empty</code> state is the normal state of a receptacle when the attachment point contains no occupant. A receptacle can also be in the <code>disconnected</code> 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 <code>connected</code> state enables normal access to hardware resources on the occupant. The <code>connected</code> state is the normal state of a receptacle that contains an occupant and that is not currently undergoing configuration administration operations. In the <code>unconfigured</code> 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 <code>configured</code> state, hardware resources are exposed in software data structures; thus, they can be in use by the Solaris operating environment.</p> <p>An attachment point can be in one of five conditions: <code>unknown</code>, <code>ok</code>, <code>failing</code>, <code>failed</code>, or <code>unusable</code>. An attachment point can enter the system in any condition, depending upon results of power-on tests and nonvolatile record keeping. With a <code>configured</code> occupant, the attachment point can be in one of four conditions: <code>unknown</code>, <code>ok</code>, <code>failing</code>, <code>failed</code>. An attachment point can change to <code>failing</code> during the course of operation if a hardware-dependent recoverable error threshold is exceeded. An attachment point can change to</p>

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_ids*) 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,isp4:scsi
/devices/pci@71,2000/pci@2/SUNW,isp4:scsi::dsk/c0t0d0
/devices/pci@71,2000/pci@2/SUNW,isp4:scsi::dsk/c0t1d0
/devices/pci@71,2000/pci@2/SUNW,isp4:scsi::dsk/c0t2d0
/devices/pci@71,2000/pci@2/SUNW,isp4: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_types* 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	SUNWssp

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 "$type" = CPU ]
    then
        then
            if [ -z "$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	<p>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.</p> <hr/> <p>Note - As of the Solaris 8 GA release, manual reconfiguration is not needed. A new DDI subsystem, devfsadm, completes all of the reconfiguration tasks.</p> <hr/> <p>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).</p> <hr/> <p>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.</p> <hr/>
EXAMPLES	<p>EXAMPLE 1 Using reconfig(1M)</p> <pre>dr> reconfig Reconfiguration of devices in progress... Reconfiguration completed successfully.</pre>
DIAGNOSTICS	<p>The following diagnostics are supported.</p> <p>Reconfiguration failed</p> <p>One or more of the reconfiguration commands failed. Check the /var/adm/messages file on the domain.</p>
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*