



Solaris 7 System Administration Supplement

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

Part No: 806-1650-10
November 1999

Copyright 1999 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, docs.sun.com, AnswerBook, AnswerBook2, Java, JDK, Power Management, 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 1999 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, docs.sun.com, AnswerBook, AnswerBook2, Java, JDK, Power Management, 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.

Contents

Preface vii

1. What's New at a Glance 1

2. Device Configuration Enhancements 7

SPARC: Dynamic Reconfiguration for Hot-Pluggable System Boards 7

Improved Device Configuration (devfsadm) 8

x86: PCI Hot-Plugging With cfgadm 8

 x86: Hardware That Supports Hot-Plugging 9

 x86: PCI Hot-Plugging With the cfgadm Command 9

▼ How to Display PCI Slot Configuration Information 10

▼ How to Remove a PCI Adapter Card 10

▼ How to Add a PCI Adapter Card 12

 x86: Troubleshooting PCI Configuration Problems 13

SCSI Hot-Plugging Features (cfgadm) 14

 Attachment Points 15

 SCSI Hot-Plugging With the cfgadm Command 16

▼ How to Display Configuration Information for All Devices 16

▼ How to Unconfigure a SCSI Controller 18

▼ How to Configure a SCSI Controller 18

▼ How to Configure a SCSI Device 19

- ▼ How to Disconnect a SCSI Controller 20
- ▼ How to Connect a SCSI Controller 21
- ▼ How to Add a SCSI Device to a SCSI Bus 22
- ▼ How to Replace an Identical Device on a SCSI Controller 23
- ▼ How to Remove a SCSI Device 25

3. File System Enhancements 27

DVD Support 27

Using the Universal Disk Format File System 28

UDF Features and Benefits 28

Hardware and Software Requirements 28

- ▼ How to Connect a DVD-ROM Device 29
- ▼ How to Access Files on a DVD-ROM Device 29
- ▼ How to Display UDF File System Parameters 30
- ▼ How to Create a UDF File System 30
- ▼ How to Identify the UDF File System Type 30
- ▼ How to Check a UDF File System 31
- ▼ How to Mount a UDF File System 31
- ▼ How to Unmount a UDF File System 31
- ▼ How to Label a Device With a UDF File System and Volume Name 32

Deferred Access Time Updates on UFS File Systems 32

4. Diagnostic Enhancements 33

Improved Core File Management (coreadm) 33

Configurable Core File Paths 34

Expanded Core File Names 34

Setting the Core File Name Pattern 35

Enabling setuid Programs to Produce Core Files 36

- ▼ How to Display the Current Core Dump Configuration 36
- ▼ How to Set a Core File Name Pattern 37

▼	How to Display a Core File Name Pattern	37
▼	How to Enable a Per-Process Core File Path	37
▼	How to Enable a Global Core File Path	38
	Troubleshooting Core File Problems	38
	New Console Features for Remote Systems	39
	Using Auxiliary Console Messaging During Run Level Transitions	40
	Using the consadm Command During an Interactive Login Session	41
▼	How to Enable an Auxiliary (Remote) Console	41
▼	How to Display a List of Auxiliary Consoles	42
▼	How to Enable an Auxiliary (Remote) Console Across System Reboots	43
▼	How to Disable an Auxiliary (Remote) Console	43
	Improved System Boot and Error Message Format	44
	System Boot and Error Message Format Changes	44
▼	How to Enable System Message IDs	45
▼	How to Disable System Message IDs	45
5.	System Administration Notes	47
	Migrating to BIND 8.1	47
	AnswerBook2 Server Software Updates	47
	SPARC: Running cron Jobs With Power Management Software	48

Preface

The *Solaris 7 System Administration Supplement* describes how to use new system administration features in the Solaris™ 7 11/99, 8/99, 5/99, and 3/99 software releases.

Related Books

This document describes new or changed functionality in Solaris update releases. The information here supplements or supersedes information in the previous releases of Solaris 7 documentation sets.

Ordering Sun Documents

The Sun Software Shop stocks select manuals from Sun Microsystems, Inc. You can purchase individual printed manuals and AnswerBook2™ CDs.

For a list of documents and how to order them, visit the Software Shop at <http://www.sun.com/software/shop/>.

Accessing Sun Documentation Online

The docs.sun.comSM web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is <http://docs.sun.com>.

What Typographic Conventions Mean

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

TABLE P-1 Typographic Conventions

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

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

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

What's New at a Glance

This chapter highlights new features added to the Solaris™ 7 operating environment in the 11/99, 8/99, 5/99, and 3/99 releases.

TABLE 1-1 Solaris 7 Features

Feature	First Released in . . .	Description
Installation (These features are described in the <i>Solaris 7 Installation Collection</i> .)		
AnswerBook2 ab2cd script updates	8/99	Updates to the ab2cd script enable users to read AnswerBook2™ documentation directly from the Solaris Documentation CD.
DNS configuration during system identification	5/99	Prior Solaris releases only allowed the configuration of a machine as a NIS or NIS+ client. With the Solaris 7 5/99 release, systems can be configured by the system identification utilities to be DNS clients.
Product Registry	3/99	The Solaris Product Registry is a tool to manage installed software. It enables you to: ■ View a list of installed and registered software and some software attributes ■ Uninstall software ■ Browse for and launch an installer
Patch Analyzer	3/99	The Patch Analyzer performs an analysis on your system to determine which (if any) patches will be removed or downgraded by upgrading from a Solaris 7 operating environment to the Solaris 7 11/99 operating environment. This analysis can be performed either at upgrade time or while the system is in production mode.

TABLE 1-1 Solaris 7 Features *(continued)*

Feature	First Released in . . .	Description
(Intel Platform Edition only) Added support for PAE mode	3/99	With the release of Pentium Pro, Intel introduced a mode called Physical Address Extension (PAE) on its advanced processors. PAE mode enables you to run multiple instances of databases and memory-intensive applications, and to support large numbers of online users on your machine. This feature is described in the <i>Solaris 7 11/99 Release Notes</i>.
System and Network Administration		
Universal Disk Format (UDF) file system	11/99	The UDF file system is the industry-standard format for storing information on optical media called digital versatile disc or digital video disc (DVD) that is both readable and re-writable.
Digital versatile disk or digital video disk	11/99	This release supports digital versatile disc or digital video disc (DVD) that is both readable and re-writable.
(Intel Platform Edition only) PCI hot-plugging with <code>cfgadm</code> command	11/99	The <code>cfgadm</code> command is updated to provide PCI hot-plugging for supported PCI controllers on x86 systems.
Improved device configuration (<code>devfsadm</code>)	11/99	The <code>devfsadm</code> command manages the special device files in the <code>/dev</code> and <code>/devices</code> directories and is updated to detect device configuration changes by any reconfiguration event.
(SPARC Platform Edition only) Inter-Domain Network (IDN)	11/99	Inter-Domain Network (IDN) supports high-speed networking between Dynamic System Domains (DSDs). This enables DSDs to communicate with each other using standard networking interfaces, such as TCP/IP, without the use of cabling or special hardware. See your hardware manufacturer's documentation for information about IDN and to see if IDN supports your server.
<code>coreadm</code>	8/99	The <code>coreadm</code> command provides improved core file management for Solaris system administrators.
Enhanced SCSI hot-plugging with the <code>cfgadm</code>	8/99	The <code>cfgadm</code> command now supports SCSI hot-plugging for supported SCSI hardware.
Migrating to BIND 8.1	8/99	This section updates Bind 8.1 in the <i>Solaris Naming Setup and Configuration Guide</i> and the <i>Solaris Naming Administration Guide</i> .
AnswerBook2 server software updates	8/99	Updates in AnswerBook2 version 1.4.1 server software improve navigation and performance.

TABLE 1-1 Solaris 7 Features *(continued)*

Feature	First Released in . . .	Description
Remote console	5/99	New system console features, including the <code>consadm</code> command, enable system administrators and service providers to remotely troubleshoot systems by dialing in to a system using a modem connected to one of the serial ports and to redirect console messages to the auxiliary device.
(SPARC™ Platform Edition only) Dynamic reconfiguration	5/99	Dynamic reconfiguration allows the service provider to add or remove and replace hot-pluggable system boards in a running system, eliminating the time lost in rebooting. For more information, see your hardware manufacturer's documentation.
Deferred access time updates on UFS file systems	3/99	New mount options, <code>dfratime</code> and <code>nodfratime</code> , enable and disable deferred access time updates on UFS file systems.
Improved system boot and error message format	3/99	The system boot and error message format is improved by adding a numeric identifier, module name, and time stamp to messages generated by the <code>syslog</code> logging facility.
(SPARC Platform Edition only) Improved handling of <code>crontab</code> jobs on suspended systems	3/99	When you use Power Management™ software to suspend and resume a system, there is now a check on whether <code>crontab</code> jobs were scheduled during the time the system was suspended. If there were, the first instance of a <code>crontab</code> job is performed when the system resumes. (This improvement was available in the Solaris 7 release and first documented in the Solaris 7 3/99 release.)
Developer (These features are described in the <i>Solaris 7 Software Developer Supplement</i> .)		
(SPARC Platform Edition) Dynamic reconfiguration support for SCSI HBA drivers	11/99	In this release, dynamic reconfiguration (DR) support has been updated for SCSI devices. See your hardware manufacturer's documentation to see if your server supports DR.
PCI hot-plugging	11/99	PCI hot-plugging is updated to include information on writing device drivers for x86 servers.
8-bit visual support	8/99	The 8-bit visual shared library enables device drivers with only 24-bit hardware to display 8-bit visual applications.
<i>Solaris 7 64-bit Developer's Guide</i> updates	3/99	This is a list of updates in the <i>Solaris 7 64-bit Developer's Guide</i> .

TABLE 1-1 Solaris 7 Features *(continued)*

Feature	First Released in . . .	Description
Updated DDI interfaces for cluster-aware device drivers	3/99	This overview introduces the concept of device classes and the necessary interface modifications and additions for device driver writers.
Java Development Kit (These features are described in the <i>Solaris 7 Software Developer Supplement</i> .)		
JDK 1.1.7_08	11/99	The JDK™ 1.1.7_08 includes: ■ Enhanced Eurocurrency support ■ High-performance Just In Time compiler A Java™ Runtime Environment (JRE) corresponding to the latest version of the JDK is available for download from the Web. http://www.sun.com/solaris/java
Java 2 SDK 1.2.1_03	11/99	The Java 2 SDK 1.2.1_03 is the first Solaris release of Java technology based on Java 2. It includes: ■ Substantially increased scalability and performance ■ Enhanced memory management system ■ High-performing, scalable Java Virtual Machine (JVM) ■ Fast Java thread synchronization ■ Enhanced diagnostics with the heap inspection tool, which finds memory leaks in programs ■ Optimizing Just In Time compiler
Desktop User (These features are described in the <i>Solaris 7 User Supplement</i> .)		
Personal Digital Assistant (PDA) synchronization	11/99	PDA synchronization enables data to be synchronized from Sun applications, such as Calendar Manager, with data in a similar application on your PDA. Also, applications and databases can be installed from your workstation or server to your PDA.

TABLE 1-1 Solaris 7 Features *(continued)*

Feature	First Released in . . .	Description
Netscape™ application launcher	11/99	The Desktop-Apps folder window has four new icons: ■ Netscape ■ Netscape Composer ■ Netscape Mail ■ Netscape News
Xserver 11R6.4	11/99	The Xserver 11R6.4 provides the following new features: ■ XPrint ■ Xkeyboard ■ Display Power Management Signalling ■ Xinerama ■ Color Utilization Policy ■ WebEnabledX New man pages are available for each of these features.

Device Configuration Enhancements

This chapter describes the following new system administration information:

- “SPARC: Dynamic Reconfiguration for Hot-Pluggable System Boards” on page 7
- “Improved Device Configuration (devfsadm)” on page 8
- “x86: PCI Hot-Plugging With cfgadm ” on page 8
- “SCSI Hot-Plugging Features (cfgadm)” on page 14

Note - For the most up-to-date man pages, use the `man` command. The Solaris 7 11/99 man pages include new feature information not found in the *Solaris 7 Reference Manual Collection*.

SPARC: Dynamic Reconfiguration for Hot-Pluggable System Boards

This functionality was changed in the Solaris 7 5/99 software release.

Dynamic reconfiguration (DR) enables a server with a faulty system board to continue running while that board is logically detached, physically removed, physically replaced, then logically attached, eliminating downtime and the time lost in rebooting. Once a board has been detached the server can continue running even if a replacement board is not yet available. In servers that support domains, DR can be used to reassign a good board from one domain to another. On selected servers, CPU/memory boards can also be logically attached and detached.

The system administrator typically executes the DR commands, while the service provider does the actual board swapping. Only certain SPARC servers support this

version of DR. See your hardware manufacturer's documentation for information about whether your server supports DR.

Improved Device Configuration (devfsadm)

This functionality is updated in the Solaris 7 11/99 software release.

The `devfsadm` command provides an improved mechanism for managing the special device files in the `/dev` and `/devices` directories, including support for dynamic reconfiguration events.

In previous Solaris releases, device configuration was handled by `drvconfig`, which managed the physical device entries in the `/devices` directory, and five link generators, `devlinks`, `disks`, `tapes`, `ports`, and `audlinks`, which managed the logical device entries in the `/dev` directory.

These utilities were not aware of hot-pluggable devices nor were they flexible enough for devices with multiple instances. For compatibility purposes, `drvconfig` and the other link generators are symbolic links to the `devfsadm` utility. See *System Administration Guide, Volume I* for information about hot-pluggable devices.

Both reconfiguration boot processing and updating the `/dev` and `/devices` directories in response to dynamic reconfiguration events is handled by `devfsadmd`, the daemon version of the `devfsadm` command. This daemon is started from the `/etc/rc*` scripts when a system is booted.

Since `devfsadmd` automatically detects device configuration changes generated by any reconfiguration event, there is no need to run this command interactively.

See the man page `devfsadm(1M)` for more information.

x86: PCI Hot-Plugging With `cfgadm`

The `cfgadm` command is updated in the Solaris 7 11/99 software release to provide PCI hot-plugging for supported PCI adapter cards on x86 systems. This information supplements information on managing devices found in "Managing Devices" in *System Administration Guide, Volume I*.

Hot-plugging PCI adapter cards provides the ability to add and remove a hot-pluggable PCI adapter card on a running system without rebooting the system.

As part of Sun's high availability strategy, this feature is expected to be used in conjunction with additional layered products, such as alternate pathing or fail-over software, which provide fault tolerance in the event of a device failure.

Without any high availability software, you can replace a failed device by manually stopping the appropriate applications, unconfiguring the device, and then proceeding with the add or remove operations.

x86: Hardware That Supports Hot-Plugging

Currently, PCI hot-plugging is supported on the Intel platform. Please refer to *x86 Hardware Compatibility List* for a list of PCI hardware that supports hot-plugging.

x86: PCI Hot-Plugging With the `cfgadm` Command

The following section describes different hot-plugging operations and then provides step-by-step instructions for hot-plugging PCI adapter cards.

Detaching PCI Adapter Cards

A PCI adapter card hosting non-vital system resources can be removed if the device driver supports hot-plugging.

A PCI adapter card is not detachable if it is a vital system resource. For a PCI adapter card to be detachable:

- The device driver must support hot-plugging.
- Critical resources must be accessible through an alternate pathway.

For example, if a system has only one ethernet card installed in it, the ethernet card cannot be detached without losing network connection. This replacement requires additional layered software support to keep the network connection active.

See “x86: How to Remove a PCI Adapter Card” on page 10 for step-by-step instructions on removing a PCI adapter card.

Attaching PCI Adapter Cards

A PCI adapter card can be added to the system as long as:

- There are slots available.
- The device driver supports hot-plugging for this adapter card.

See “Attaching PCI Adapter Cards” on page 9 for step-by-step instructions on adding a PCI adapter card.

In the following examples, only PCI attachment points are listed, for brevity. The attachment points displayed on your system will depend on your system configuration.

▼ x86: How to Display PCI Slot Configuration Information

The `cfgadm (1M)` command displays the status of PCI hot-pluggable devices and slots on a system.

1. Become superuser.

2. Display PCI slot configuration information.

```
# cfgadm pci
Ap_Id                Receptacle  Occupant    Condition
pci1:hpc0_slot0      empty       unconfigured unknown
pci1:hpc0_slot1      empty       unconfigured ok
pci1:hpc0_slot2      empty       unconfigured ok
pci1:hpc0_slot3      connected   configured  ok
pci1:hpc0_slot4      empty       unconfigured ok
# cfgadm -s "cols=ap_id:type:info" pci
Ap_Id                Type        Information
pci1:hpc0_slot0      unknown     Slot 7
pci1:hpc0_slot1      unknown     Slot 8
pci1:hpc0_slot2      unknown     Slot 9
pci1:hpc0_slot3      Ethernet/HP Slot 10
pci1:hpc0_slot4      unknown     Slot 11
```

The logical `ap_id`, `pci1:hpc0_slot0`, is the logical `ap_id` for that particular hot-pluggable slot, Slot 7, (physical identification of this slot). The component `hpc0` indicates the hot-pluggable adapter card for this slot and `pci1` indicates the PCI bus instance. The `Type` field indicates the type of PCI adapter card present in the slot.

▼ x86: How to Remove a PCI Adapter Card

1. Become superuser.

2. Determine which slot the controller is in.

```
# cfgadm
Ap_Id                Receptacle  Occupant    Condition
pci1:hpc0_slot0      empty       unconfigured unknown
pci1:hpc0_slot1      empty       unconfigured unknown
pci1:hpc0_slot2      empty       unconfigured unknown
pci1:hpc0_slot3      connected   configured   ok
pci1:hpc0_slot4      empty       unconfigured unknown
```

3. Stop the application that has the device open.

For example, if this is an ethernet card, use `ifconfig(1M)` to bring down the interface and unplumb the interface.

4. Unconfigure the device.

```
# cfgadm -c unconfigure pci1:hpc0_slot3
```

5. Confirm the device has been unconfigured.

```
# cfgadm
Ap_Id                Receptacle  Occupant    Condition
pci1:hpc0_slot0      empty       unconfigured unknown
pci1:hpc0_slot1      empty       unconfigured unknown
pci1:hpc0_slot2      empty       unconfigured unknown
pci1:hpc0_slot3      connected   unconfigured unknown
pci1:hpc0_slot4      empty       unconfigured unknown
```

6. Disconnect the power to the slot.

```
# cfgadm -c disconnect pci1:hpc0_slot3
```

7. Confirm the device has been disconnected.

```
# cfgadm
Ap_Id                Receptacle  Occupant    Condition
pci1:hpc0_slot0      empty       unconfigured unknown
pci1:hpc0_slot1      empty       unconfigured unknown
pci1:hpc0_slot2      empty       unconfigured unknown
pci1:hpc0_slot3      disconnected unconfigured unknown
pci1:hpc0_slot4      empty       unconfigured unknown
```

8. Open the slot latches and remove the board.

▼ x86: How to Add a PCI Adapter Card

1. Become superuser.

2. Identify the hot-pluggable slot and open latches.

3. Insert the adapter card into a hot-pluggable slot.

4. Determine which slot the adapter card is in once it is inserted and the latches are closed.

# cfgadm	Ap_Id	Receptacle	Occupant	Condition
	pci1:hpc0_slot0	empty	unconfigured	unknown
	pci1:hpc0_slot1	empty	unconfigured	unknown
	pci1:hpc0_slot2	empty	unconfigured	unknown
	pci1:hpc0_slot3	disconnected	unconfigured	unknown
	pci1:hpc0_slot4	empty	unconfigured	unknown

5. Connect the power to the slot.

```
# cfgadm -c connect pci1:hpc0_slot3
```

6. Confirm the slot is connected.

# cfgadm	Ap_Id	Receptacle	Occupant	Condition
	pci1:hpc0_slot0	empty	unconfigured	unknown
	pci1:hpc0_slot1	empty	unconfigured	unknown
	pci1:hpc0_slot2	empty	unconfigured	unknown
	pci1:hpc0_slot3	connected	unconfigured	unknown
	pci1:hpc0_slot4	empty	unconfigured	unknown

7. Configure the PCI hot-pluggable adapter card.

```
# cfgadm -c configure pci1:hpc0_slot3
Ap_Id      Receptacle  Occupant    Condition
pci1:hpc0_slot3  connected  configured  ok
```

8. Verify the configuration of the adapter card in the slot.

```
# cfgadm pci1:hpc0_slot3
```

9. Configure any supporting software if this is a new device.

For example, if this is an ethernet card, use the `ifconfig(1M)` command to set up the interface.

x86: Troubleshooting PCI Configuration Problems

Error Message

```
cfgadm: Configuration operation invalid: invalid transition
```

Cause

An invalid transition was attempted.

Solution

Check whether the `cfgadm -c` command was issued appropriately. Use `cfgadm` to check the current receptacle and occupant state and make sure the `ap_id` is correct.

Error Message

```
cfgadm: Attachment point not found
```

Cause

Specified attachment point was not found.

Solution

Check whether the attachment point is correct. Use the `cfgadm` to display a list of available attachment points. Also check the physical path to see if the attachment point is still there.

Note - In addition to `cfgadm` command, several other commands are helpful during hot-plug operations. The `prtconf(1M)` command confirms whether or not Solaris recognizes the hardware. After a successful configuration of a hot-pluggable adapter card, use the `prtconf` command to verify that the specific PCI bus instance for the new hardware displays in the `prtconf` output. The `prtconf` output also indicates if the driver for the new hardware is attached.

SCSI Hot-Plugging Features (`cfgadm`)

The `cfgadm` command is updated in the Solaris 7 8/99 software release to provide SCSI hot-plugging for supported SCSI controllers. This information supplements information on managing devices found in “Managing Devices” in *System Administration Guide, Volume I*.

Hot-plugging is the ability to physically add, remove, or replace system components while the system is running. *Dynamic reconfiguration* refers to the ability to hot-plug system components and also the general ability to move system resources—both hardware and software—around in the system, or to disable them without removing them from the system.

Features of the `cfgadm` command include:

- Displaying system component status
- Testing system components
- Changing component configurations
- Displaying configuration help messages

The benefit of using the `cfgadm` command to reconfigure SCSI components is that you can add, remove, or replace components while the system is running. An added benefit is that the `cfgadm` command guides you through the steps needed to add, remove, or replace SCSI components. See the man page `cfgadm(1M)` and “SCSI Hot-Plugging With the `cfgadm` Command” on page 16 for step-by-step instructions on hot-plugging SCSI components.

Note - Not all SCSI controllers support hot-plugging with the `cfgadm` command.

Attachment Points

The `cfgadm` command displays information about *attachment points*, which are locations in the system where dynamic reconfiguration operations can occur.

An attachment point consists of:

- An *occupant*, which represents a hardware resource that may be configured into the system, and
- A *receptacle*, which is the location that accepts the occupant resource

Attachment points are named by logical and physical attachment point IDs (`ap_id`). The physical `ap_id` is the physical pathname of the attachment point. The logical `ap_id` is a user-friendly name that represents the physical attachment point pathname. Refer to *cfgadm(1M)* for more information on `ap_id`.

The logical `ap_id` for a SCSI host bus adapter (HBA), or SCSI controller, is usually represented by the controller number, such as `c0`.

In cases where no controller number has been assigned to a SCSI HBA, then an internally-generated unique identifier is provided. An example of a unique identifier for a SCSI controller is:

```
fas1:scsi
```

The logical `ap_id` for a SCSI device usually looks like this:

```
HBA-logical-apid::device-identifier
```

In the example below, `c0` is the logical `ap_id` for the SCSI HBA:

```
c0::dsk/c0t3d0
```

The device identifier is typically derived from the logical device name for the device in the `/dev` directory. For example, a tape device with logical device name, `/dev/rmt/1`, has the following logical `ap_id`:

```
c0::rmt/1
```

If a logical `ap_id` of a SCSI device cannot be derived from the logical name in the `/dev` directory, then an internally-generated unique identifier is provided. An example of an identifier for the tape device `/dev/rmt/1` is:

```
c0::st4
```

Refer to the man page *cfgadm_scsi(1M)* for more information on SCSI `ap_ids`.

The `cfgadm` command represents all resources and dynamic reconfiguration operations in terms of a common set of states (such as configured, unconfigured) and set of operations (connect, configure, unconfigure). Refer to the man page *cfgadm(1M)* for more information on these generic states and operations.

The receptacle and occupant states for the SCSI HBA attachment points are:

Receptacle	Description	Occupant	Description
empty	N / A to SCSI HBA	configured	One or more devices configured on the bus
disconnected	Bus quiesced	unconfigured	No devices configured
connected	Bus active		

Receptacle and occupant state mappings for SCSI devices are:

Receptacle	Description	Occupant	Description
empty	N / A to SCSI devices	configured	Device is configured
disconnected	Bus quiesced	unconfigured	Device is not configured
connected	Bus active		

The condition of SCSI attachment points is "unknown" unless there is special hardware to indicate otherwise. See “How to Display Configuration Information for All Devices” on page 16 for instructions on displaying SCSI component configuration information.

SCSI Hot-Plugging With the `cfgadm` Command

The following section describes various SCSI hot-plugging tasks with the `cfgadm` command.

The procedures in this section use specific devices as examples to illustrate how to use the `cfgadm` command to hot plug SCSI components. The device information you supply, and that is displayed with the `cfgadm` command, will depend on your system configuration.

▼ How to Display Configuration Information for All Devices

SCSI controllers `c0` and `c1` and the devices attached to them are examples of the type of device configuration information that can be displayed with the `cfgadm` command.

Note - If the SCSI device is not supported by the `cfgadm` command, it will not display in the `cfgadm` command output.

1. Become superuser.

2. Display information about attachment points on the system.

# cfgadm -l			
Ap_Id	Receptacle	Occupant	Condition
ac0:bank0	connected	configured	ok
ac0:bank1	connected	configured	ok
c0	connected	configured	unknown
c1	connected	configured	unknown
sysctrl0:slot0	connected	configured	ok
sysctrl0:slot1	connected	configured	ok

In this example, `c0` and `c1` represent two SCSI controllers.

The other `ap_ids` such as `ac0:bank1` and `sysctrl0:slot0` in the command output represent other types of system components.

3. Display information about a system's SCSI controllers and their attached devices.

# cfgadm -al			
Ap_Id	Receptacle	Occupant	Condition
ac0:bank0	connected	configured	ok
ac0:bank1	connected	configured	ok
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown
c1::dsk/c1t4d0	connected	configured	unknown
c2	connected	unconfigured	unknown
sysctrl0:slot0	connected	configured	ok
sysctrl0:slot1	connected	configured	ok

Notice that SCSI HBAs are displayed in the `cfgadm -l` output but not SCSI devices. The `cfgadm -al` command displays information about SCSI devices such as disks and tapes.

▼ How to Unconfigure a SCSI Controller

SCSI controller c1 provides an example of unconfiguring a SCSI controller.

Only SCSI attachment points are listed. The attachment points displayed on your system will depend on your system configuration.

1. **Become superuser.**
2. **Unconfigure a SCSI controller.**

```
# cfgadm -c unconfigure c1
```

3. **Verify the SCSI controller is unconfigured.**

```
# cfgadm -al
Ap_Id                      Receptacle  Occupant    Condition
c0                          connected   configured  unknown
c0::dsk/c0t0d0             connected   configured  unknown
c0::rmt/0                   connected   configured  unknown
c1                          connected   unconfigured unknown
```

Notice that the Occupant column specifies unconfigured for c1, indicating that the SCSI bus has no configured occupants.

▼ How to Configure a SCSI Controller

SCSI controller c1 provides an example of configuring a SCSI controller.

Only SCSI attachment points are listed. The attachment points displayed on your system will depend on your system configuration.

1. **Become superuser.**
2. **Configure a SCSI controller.**

```
# cfgadm -c configure c1
```

3. Verify the SCSI controller is configured.

```
# cfgadm -al
```

Ap_Id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown
c1::dsk/c1t4d0	connected	configured	unknown

The previous unconfigure procedure removed all devices on the SCSI bus.
Now all the devices are configured back into the system.

▼ How to Configure a SCSI Device

SCSI disk c1t4d0 provides an example of configuring a SCSI device.

Only SCSI attachment points are listed. The attachment points displayed on your system will depend on your system configuration.

1. Become superuser.

2. Identify the device to be configured.

```
cfgadm -al
```

Ap_Id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown
c1::dsk/c1t4d0	connected	unconfigured	unknown

3. Configure a specific SCSI device.

```
# cfgadm -c configure c1::dsk/c1t4d0
```

4. Verify the SCSI device is configured.

```
# cfgadm -al
```

Ap_Id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown
c1::dsk/c1t4d0	connected	configured	unknown

▼ How to Disconnect a SCSI Controller

Disconnecting a SCSI device must be done with caution, particularly when dealing with controllers for disks containing critical file systems such as root /, usr, var, and the swap partition. The dynamic reconfiguration software cannot detect all cases where a system hang may result. Use this command with caution.

SCSI controller c1 provides an example of disconnecting a SCSI device.

1. Become superuser.

2. Verify the device is connected before disconnecting it.

```
# cfgadm -al
```

Ap_Id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown
c1::dsk/c1t4d0	connected	configured	unknown

3. Disconnect a SCSI controller.

```
# cfgadm -c disconnect c1
WARNING: Disconnecting critical partitions may cause system hang.
Continue (yes/no)? y
```



Caution - This command suspends all I/O activity on the SCSI bus until the `cfgadm -c connect` command is used. The `cfgadm` command does some basic checking to prevent critical partitions from being disconnected, but it cannot detect all cases. Inappropriate use of this command may result in a system hang and could require a system reboot.

4. Verify the SCSI bus is disconnected.

```
# cfgadm -al
Ap_Id                      Receptacle  Occupant    Condition
c0                          connected   configured  unknown
c0::dsk/c0t0d0              connected   configured  unknown
c0::rmt/0                    connected   configured  unknown
c1                           disconnected configured  unknown
c1::dsk/c1t10d0              disconnected configured  unknown
c1::dsk/c1t4d0              disconnected configured  unknown
```

The controller and all the devices attached to it are disconnected from the system.

▼ How to Connect a SCSI Controller

SCSI controller c1 provides an example of connecting a SCSI controller.

Only SCSI attachment points are listed. The attachment points displayed on your system will depend on your system configuration.

1. Become superuser.

2. Verify the device is disconnected before connecting it.

```
# cfgadm -al
Ap_Id                      Receptacle  Occupant    Condition
c0                          connected   configured  unknown
c0::dsk/c0t0d0              connected   configured  unknown
c0::rmt/0                    connected   configured  unknown
c1                           disconnected configured  unknown
c1::dsk/c1t10d0              disconnected configured  unknown
c1::dsk/c1t4d0              disconnected configured  unknown
```

3. Connect a SCSI controller.

```
# cfgadm -c connect c1
```

4. Verify the SCSI controller is connected.

```
# cfgadm -al
```

Ap_Id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown
c1::dsk/c1t4d0	connected	configured	unknown

▼ How to Add a SCSI Device to a SCSI Bus

SCSI controller c1 provides an example of adding a SCSI device to a SCSI bus.

Only SCSI attachment points are listed. The attachment points displayed on your system will depend on your system configuration.

Note - When adding devices, the `ap_id` of the SCSI HBA (controller) to which the device is attached is specified, not the `ap_id` of the device itself.

1. Become superuser.

2. Identify the current SCSI configuration.

```
# cfgadm -al
```

ap_id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown

3. Add a SCSI device to a SCSI bus.

```
# cfgadm -x insert_device c1
Adding device to SCSI HBA: /devices/sbus@1f,0/SUNW,fas@1,8800000
This operation will suspend activity on SCSI bus: c1
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)? y
```

a. Type y at the Continue (yes/no)? prompt to proceed.

I/O activity on the SCSI bus will be suspended while the hot-plug operation is in progress.

b. Connect the device and then power it on.

c. Type y at the

Enter y if operation is complete or n to abort (yes/no)?
prompt after the new device has been inserted.

4. Verify the device has been added.

```
# cfgadm -al
```

Ap_Id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown
c1::dsk/c1t4d0	connected	configured	unknown

A new disk has been added to controller c1.

▼ How to Replace an Identical Device on a SCSI Controller

SCSI disk c1t4d0 provides an example replacing an identical device on a SCSI controller.

Only SCSI attachment points are listed. The attachment points displayed on your system will depend on your system configuration.

1. Become superuser.

2. Identify the current SCSI configuration.

```
# cfgadm -al
Ap_Id                      Receptacle  Occupant    Condition
c0                          connected   configured  unknown
c0::dsk/c0t0d0             connected   configured  unknown
c0::rmt/0                   connected   configured  unknown
c1                          connected   configured  unknown
c1::dsk/c1t10d0            connected   configured  unknown
c1::dsk/c1t4d0             connected   configured  unknown
```

3. Replace a device on the SCSI bus with another device of the same type.

```
# cfgadm -x replace_device c1::dsk/c1t4d0
Replacing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,88000000/sd@4,0
This operation will suspend activity on SCSI bus: c1
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)? y
```

a. Type **y** at the Continue (yes/no)? prompt to proceed.

I/O activity on the SCSI bus will be suspended while the hot-plug operation is in progress.

b. Power off the device to be removed and remove it. Add the replacement device, which should be of the same type and at the same address (target and lun) as the device to be removed. Then power it on.

c. Type **y** at the

Enter y if operation is complete or n to abort (yes/no)? prompt after the device has been replaced.

4. Verify the device has been replaced.

```
# cfgadm -al
```

Ap_Id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown
c1::dsk/c1t4d0	connected	configured	unknown

▼ How to Remove a SCSI Device

SCSI disk c1t4d0 provides an example of removing a device on a SCSI controller.

Only SCSI attachment points are listed. The attachment points displayed on your system will depend on your system configuration.

1. **Become superuser.**
2. **Identify the current SCSI configuration.**

```
# cfgadm -al
```

Ap_Id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown
c1::dsk/c1t4d0	connected	configured	unknown

3. **Remove a SCSI device from the system.**

```
# cfgadm -x remove_device c1::dsk/c1t4d0
Removing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
This operation will suspend activity on SCSI bus: c1
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
Enter y if operation is complete or n to abort (yes/no)? y
```

- a. **Type y at the Continue (yes/no)? prompt to proceed.**

I/O activity on the SCSI bus will be suspended while the hot-plug operation is in progress.

b. Power off the device to be removed and remove it.

c. Type y at the

Enter y if operation is complete or n to abort (yes/no)?
prompt after the device has been removed.

4. Verify the device has been removed from the system.

# cfgadm -al			
Ap_Id	Receptacle	Occupant	Condition
c0	connected	configured	unknown
c0::dsk/c0t0d0	connected	configured	unknown
c0::rmt/0	connected	configured	unknown
c1	connected	configured	unknown
c1::dsk/c1t10d0	connected	configured	unknown

File System Enhancements

This chapter describes the following new system administration information:

- “DVD Support” on page 27
- “Using the Universal Disk Format File System ” on page 28
- “Deferred Access Time Updates on UFS File Systems” on page 32

Note - For the most up-to-date man pages, use the `man` command. The Solaris 7 11/99 man pages include new feature information not found in the *Solaris 7 Reference Manual Collection*.

DVD Support

The Solaris 7 11/99 release provides support for digital versatile disc or digital video disc (DVD devices), the optical media technology that is both readable and re-writable, on certain SPARC and Intel based systems.

See the following section for information on accessing information on DVD devices through the UDF file system. This information supplements information found in *System Administration Guide, Volume I*.

Using the Universal Disk Format File System

The Solaris 7 11/99 release provides support for Universal Disk Format (UDF) , the industry-standard format for storing information on DVD devices. This information supplements information found in *System Administration Guide, Volume I*.

The UDF file system is provided as dynamically loadable, 32-bit and 64-bit modules, with system administration utilities for creating, mounting, and checking the file system on both SPARC and Intel platforms. The Solaris UDF file system works with supported ATAPI and SCSI DVD drives, CD-ROM devices, and disk and diskette drives. In addition, the Solaris UDF file system is fully compliant with the UDF 1.50 specification.

The UDF file system support is provided in the following new packages:

- SUNWudfr — 32-bit kernel component
- SUNWudfxr — 64-bit kernel component
- SUNWudf — /usr component

UDF Features and Benefits

In the Solaris 7 11/99 release, the UDF file system provides the following features:

- Ability to access the industry standard CD-ROM and DVD-ROM media when they contain a UDF file system.
- Flexibility in exchanging information across platforms and operating systems.
- A mechanism for implementing new applications rich in broadcast-quality video, high-quality sound along with the richness in interactivity using DVD video specification based on UDF format.

The following features are not included in this UDF file system release:

- Support for write-once media, CD-RW and DVD-RAM, with both the sequential disk-at-once and incremental recording.
- UFS components such as quotas, ACLs, transaction logging, file system locking, and file system threads, which are not UDF 1.50 specification requirements.

Hardware and Software Requirements

The UDF file system requires the following:

- The Solaris 7 11/99 release
- Supported SPARC or Intel platforms
- Supported CD-ROM or DVD-ROM device

UDF Compatibility Issues

This first Solaris UDF file system implementation provides:

- Support for industry-standard read-write UDF version 1.50.
- Fully internationalized file system utilities.

▼ How to Connect a DVD-ROM Device

1. **Become superuser.**
2. **Create the `/reconfigure` file.**

```
# touch /reconfigure
```

3. **Shut down the system and turn off power.**

```
# init 0
```

4. **Connect the DVD-ROM device.**
5. **Turn on power to the system.**

▼ How to Access Files on a DVD-ROM Device

1. **Verify the DVD-ROM device is automatically mounted.**

```
$ ls /cdrom
```

Note - If the system has both a CD-ROM and DVD-ROM device, the CD-ROM might be named `/cdrom/cdrom0` and the DVD-ROM might be named `/cdrom/cdrom1`. If the system only has a DVD-ROM device, then try using `/cdrom/cdrom0`.

2. Display content with the `ls` command.

```
$ ls /cdrom/cdrom1  
Copyright filea fileb
```

Automatic display with the Common Desktop Environment (CDE) file manager is not implemented yet. All other CDE file manager functions—drag and drop for copying and ImageTool features—are available.

▼ How to Display UDF File System Parameters

Display UDF file system parameters by using the `mkfs` command.

1. **Become superuser.**
2. **Display UDF file system parameters.**

```
# mkfs -F udfs -m /dev/rdisk/device-name
```

▼ How to Create a UDF File System

Create a UDF file system by using the `mkfs` command.

1. **Become superuser.**
2. **Create a UDF file system.**

```
# mkfs -F udfs /dev/rdisk/device-name
```

3. **Verify the UDF file system is created by mounting it.**

See the man page, *mkfs_udfs.1m*, for more information.

▼ How to Identify the UDF File System Type

Identify the UDF file system type by using the `fstyp` command.

1. **Become superuser.**

2. **Determine whether a file system is a UDF file system.**

```
# fstyp -v /rdev/dsk/device-name
```

▼ How to Check a UDF File System

Check the integrity of a UDF file system by using the `fsck` command.

1. **Become superuser.**
2. **Check a UDF file system.**

```
# fsck -F udfs /dev/rdisk/device-name
```

See the man page, *fsck_udfs.1m*, for more information.

▼ How to Mount a UDF File System

Mount a UDF file system.

1. **Become superuser.**
2. **Mount a UDF file system.**

```
# mount -F UDFS /dev/dsk/device-name /mount-point
```

3. **Verify the UDF file system is mounted.**

```
# ls /mount-point
```

See the man page, *mount_udfs.1m*, for more information.

▼ How to Unmount a UDF File System

Unmount a UDF file system.

1. **Become superuser.**
2. **Unmount a UDF file system.**

```
# umount /dev/dsk/device-name
```

▼ How to Label a Device With a UDF File System and Volume Name

Create a file system and volume name for a UDF file system.

1. **Become superuser.**
2. **Create a file system and volume name for the UDF file system.**

```
# labelit -F UDFS /dev/rdisk/device-name fssize volume
```

See the man page, *labelit_udfs.1m*, for more information.

Deferred Access Time Updates on UFS File Systems

This feature was new in the Solaris 7 3/99 software release.

This information supplements information on mounting UFS file systems found in “The UFS File System” in *System Administration Guide, Volume I* and referenced in the *Solaris Transition Guide*.

Two new mount options, *dfratime* and *nodfratime* enable and disable deferred access time updates on UFS file systems.

When enabled, writing access time updates for the file system may be deferred until the disk is accessed for a reason other than updating access times. The default behavior is *dfratime*. Use the *nodfratime* option to disable this feature.

In addition, the mount option, *noatime*, turns off access time recording regardless of the *dfratime*/*nodfratime* value.

For specific information about UFS mount options, see the man page *mount_ufs(1M)*.

Diagnostic Enhancements

This chapter describes the following new system administration information:

- “Improved Core File Management (coreadm)” on page 33
- “New Console Features for Remote Systems” on page 39
- “Improved System Boot and Error Message Format” on page 44

Note - For the most up-to-date man pages, use the `man` command. The Solaris 7 11/99 man pages include new feature information not found in the *Solaris 7 Reference Manual Collection*.

Improved Core File Management (coreadm)

This functionality is updated in the Solaris 7 8/99 software release.

Enhanced process core file features that include the `coreadm` command update the Solaris 7 software release. This information supplements information on troubleshooting software problems found in “Troubleshooting Software Problems (Overview)” in the *System Administration Guide, Volume II*.

This release introduces the `coreadm` command, which provides flexible core file naming conventions and better core file retention. For example, you can use the `coreadm` command to configure a system so that all process core files are placed in a single system directory. This means it is easier to track problems by examining the core files in a specific directory whenever a Solaris process or daemon terminates abnormally.

Limitations of the previous Solaris process core dump features are:

- Process `core` files are placed in their current working directory, and thus all Solaris daemons, which typically `chdir` to the root (`/`) directory as part of their initialization, overwrite each other's `core` files.
- Many system daemons, such as `statd`, perform `setuid` operations but do not, for security reasons, produce core files in the event of a problem.

Configurable Core File Paths

Two new configurable `core` file paths that can be enabled or disabled independently of each other are:

- A per-process core file path, which defaults to `core` and is enabled by default. If enabled, the per-process core file path causes a `core` file to be produced when the process terminates abnormally. The per-process path is inherited by a new process from its parent process.

When generated, a per-process core file is owned by the owner of the process with read/write permissions for the owner. Only the owning user can view this file.

- A global core file path is disabled by default. If enabled, an *additional* core file with the same content as the per-process core file is produced by using the global core file path.

When generated, a global core file is owned by the superuser with read/write permissions for the superuser only. Non-privileged users cannot view this file.

When a process terminates abnormally, it produces a `core` file in the current directory as in previous Solaris releases. But if the global core file path is enabled and set to `/corefiles/core`, for example, then each process that expires produces *two* core files: one in the current working directory and one in the `/corefiles` directory.

By default, the Solaris core paths and core file retention remain the same:

- A `setuid` process does not produce core files using either the global or per-process path.
- The global core file path is disabled.
- The per-process core file path is enabled.
- The per-process core file path is set to `core`.

Expanded Core File Names

If a global core file directory is enabled, `core` files can be distinguished from one another by using the variables described in the following table.

Variable Name	Variable Definition
%p	Process ID
%u	Effective user ID
%g	Effective group ID
%f	Executable file name
%n	System node name, equivalent to the <code>uname -n</code> output
%m	Machine name, equivalent to the <code>uname -m</code> output
%t	Decimal value of <code>time(2)</code> system call
%%	Literal %

For example, if the global core file path is set to:

```
/var/core/core.%f.%p
```

and a `sendmail` process with PID 12345 terminates abnormally, it produces the following core file:

```
/var/core/core.sendmail.12345
```

Setting the Core File Name Pattern

You can set a core file name pattern on a global basis or a per-process basis. You can specify whether you want these settings saved across a system reboot.

For example, the following `coreadm` command sets the global core file pattern for all processes started by the `init` process. This pattern will persist across system reboots.

```
$ coreadm -i /var/core/core.%f.%p
```

Global core values are stored in the `/etc/coreadm.conf` file, which means these settings are saved across a system reboot.

This `coreadm` command sets the per-process core file name pattern for all processes:

```
$ coreadm -p /var/core/core.%f.%p $$
```

The \$\$ symbols represent a placeholder for the process ID of the currently running shell. The per-process core file name pattern is inherited by all child processes.

Once a global or per-process core file name pattern is set, it must be enabled with the `coreadm -e` command. See the procedures below for more information.

You can set the core file name pattern for all processes run during a user's login session by putting the command in a user's \$HOME/.profile or .login file.

Enabling setuid Programs to Produce Core Files

You can use the `coreadm` command to enable or disable `setuid` programs to produce core files for all system processes or on a per-process basis by setting the following paths:

- If the global `setuid` option is enabled, a global core file path allows all `setuid` programs on a system to produce core files.
- If the per-process `setuid` option is enabled, a per-process core file path allows specific `setuid` processes to produce core files.

By default, both flags are disabled. For security reasons, the global core file path must be a full pathname, starting with a leading /. If superuser disables per-process core files, individual users cannot obtain core files.

The `setuid` core files are owned by the superuser with read/write permissions for the superuser only. Ordinary users cannot access them even if the process that produced the `setuid` core file was owned by an ordinary user.

See `coreadm(1M)` for more information.

▼ How to Display the Current Core Dump Configuration

Use the `coreadm` command without any options to display the current core dump configuration.

```
$ coreadm
      global core file pattern: /var/core/core.%f.%p
      init core file pattern: core
      global core dumps: enabled
      per-process core dumps: enabled
      global setid core dumps: enabled
      per-process setid core dumps: disabled
      global core dump logging: disabled
```

▼ How to Set a Core File Name Pattern

1. **Determine whether you want to set a per-process or global core file and select one of the following:**

- a. **Set a per-process core file name pattern.**

```
$ coreadm -p $HOME/corefiles/%f.%p $$
```

- b. **Set a global core file name pattern.**

Become superuser first.

```
# coreadm -g /var/core/core.%f.%p
```

▼ How to Display a Core File Name Pattern

Use the following `coreadm` command to inquire about the core file settings of the current process. The `$$` symbols represent a placeholder for the process ID of the running shell.

```
$ coreadm $$  
278:    core.%f.%p
```

Superuser can inquire about any user's core file settings by using `coreadm process ID`. Ordinary users can only inquire about the core file settings of their own processes.

▼ How to Enable a Per-Process Core File Path

1. **Become superuser.**
2. **Enable a per-process core file path.**

```
# coreadm -e process
```

3. **Display the current process core file path to verify the configuration.**

```
$ coreadm $$  
1180: /home/kryten/corefiles/%f.%p
```

▼ How to Enable a Global Core File Path

1. **Become superuser.**
2. **Enable a global core file path.**

```
# coreadm -e global -g /var/core/core.%f.%p
```

3. **Display the current process core file path to verify the configuration.**

```
# coreadm  
global core file pattern: /var/core/core.%f.%p  
init core file pattern: core  
global core dumps: enabled  
per-process core dumps: enabled  
global setid core dumps: disabled  
per-process setid core dumps: disabled  
global core dump logging: disabled
```

Troubleshooting Core File Problems

Error Message

```
NOTICE: 'set allow_setid_core = 1' in /etc/system is obsolete  
NOTICE: Use the coreadm command instead of 'allow_setid_core'
```

Cause

You have an obsolete parameter that allows setuid core files in your `/etc/system` file.

Solution

Remove `allow_setid_core=1` from the `/etc/system` file. Then use the `coreadm` command to enable global setuid core file paths.

New Console Features for Remote Systems

This feature was new in the Solaris 7 5/99 software release.

The new console features in the Solaris 7 5/99 software release update the Solaris operating environment. This information supplements information on troubleshooting Solaris systems found in “Troubleshooting Software Problems (Overview)” in *System Administration Guide, Volume II* and in the *Solaris Transition Guide*.

The following new console features improve your ability to troubleshoot remote systems:

- The `consadm` command enables you to select a serial device as an *auxiliary* (or remote) console. Using the `consadm` command, a system administrator can configure one or more serial ports to display redirected console messages and to host `sulogin` sessions when the system transitions between run levels. This feature enables you to dial in to a serial port with a modem to monitor console messages and participate in `init` state transitions. (See `sulogin(1M)` and the step-by-step procedures below for more information.)

While you can log in to a system using a port configured as an auxiliary console, it is primarily an output device displaying information that is also displayed on the default console. If boot scripts or other applications read and write to and from the default console, the write output will display on all the auxiliary consoles, but the input will only be read from the default console. Auxiliary consoles cannot be used to provide input to boot scripts. (See “Using the `consadm` Command During an Interactive Login Session” on page 41 for using the `consadm` command during an interactive login session.)

- Console output now consists of kernel and `syslog` messages written to a new pseudo device, `/dev/sysmsg`. In addition, `rc` script startup messages are written to `/dev/msglog`. Previously, all of these messages were written to `/dev/console`.

Scripts that direct console output to `/dev/console` need to be changed to `/dev/msglog` if you want to see script messages displayed on the auxiliary consoles. Source code referencing `/dev/console` should be explicitly modified to use `syslog()` or `strlog()` if you want messages to be redirected to an auxiliary device.

- The `consadm` command runs a daemon to monitor auxiliary console devices. Any display device designated as an auxiliary console that disconnects—hangs up or loses carrier—is removed from the auxiliary console device list and is no longer active. Enabling one or more auxiliary consoles does not disable message display on the default console; messages continue to display on `/dev/console`.

Using Auxiliary Console Messaging During Run Level Transitions

Keep the following in mind when using auxiliary console messaging during run level transitions:

- Input cannot come from an auxiliary console if user input is expected for an `rc` script that is run when a system is booting. The input must come from the default console.
- The `sulogin` program, invoked by `init` to prompt for the superuser password when transitioning between run levels, has been modified to send the superuser password prompt to each auxiliary device in addition to the default console device.
- When the system is in single-user mode and one or more auxiliary consoles are enabled using the `consadm` command, a console login session runs on the first device to supply the correct superuser password to the `sulogin` prompt. When the correct password is received from a console device, `sulogin` disables input from all other console devices.
- A message is displayed on the default console and the other auxiliary consoles when one of the consoles assumes single-user privileges. This message indicates which device has become the console by accepting a correct superuser password. If there is a loss of carrier on the auxiliary console running the single-user shell, one of two actions may occur:
 - If the auxiliary console represents a system at run level 1, the system proceeds to the default run level.
 - If the auxiliary console represents a system at run level `S`, the system displays the `ENTER RUN LEVEL (0-6, s or S):` message on the device where the `init s` or `shutdown` command had been entered from the shell. If there isn't any carrier on that device either, you will have to reestablish carrier and enter the correct run level. The `init` or `shutdown` command will not redisplay the run-level prompt.
- If you are logged in to a system using a serial port, and an `init` or `shutdown` command is issued to transition to another run level, the login session is lost whether this device is the auxiliary console or not. This situation is identical to Solaris releases without auxiliary console capabilities.
- Once a device is selected as an auxiliary console using the `consadm` command, it remains the auxiliary console until the system is rebooted or the auxiliary console is unselected. However, the `consadm` command includes an option to set a device as the auxiliary console across system reboots. (See the procedure below for step-by-step instructions.)

Using the consadm Command During an Interactive Login Session

If you want to run an interactive login session by logging in to a system using a terminal that is connected to a serial port, and then using the `consadm` command to see the console messages from the terminal, note the following behavior.

- If you use the terminal for an interactive login session while the auxiliary console is active, the console messages are sent to the `/dev/sysmsg` or `/dev/msglog` devices.
- While you issue commands on the terminal, input goes to your interactive session and not to the default console (`/dev/console`).
- If you run the `init` command to change run levels, the remote console software kills your interactive session and runs the `sulogin` program. At this point, input is accepted only from the terminal and is treated like it is coming from a console device. This allows you to enter your password to the `sulogin` program as described in “Using Auxiliary Console Messaging During Run Level Transitions” on page 40.

Then, if you enter the correct password on the (auxiliary) terminal, the auxiliary console runs an interactive `sulogin` session, locks out the default console and any competing auxiliary console. This means the terminal essentially functions as the system console.

- From here you can change to run level 3 or go to another run level. If you change run levels, `sulogin` runs again on all console devices. If you exit or specify that the system should come up to run level 3, then all auxiliary consoles lose their ability to provide input. They revert to being display devices for console messages.

As the system is coming up, you must provide information to `rc` scripts on the default console device. After the system comes back up, the `login` program runs on the serial ports and you can log back in to another interactive session. If you have designated the device to be an auxiliary console, you will continue to get console messages on your terminal, but all input from the terminal goes to your interactive session.

▼ How to Enable an Auxiliary (Remote) Console

The `consadm` daemon does not start monitoring the port until after you add the auxiliary console with the `consadm` command. As a security feature, console messages are only redirected until carrier drops, or the auxiliary console device is unselected. This means carrier must be established on the port before you can successfully use the `consadm` command.

See the man page `consadm(1M)` for more information on enabling an auxiliary console.

1. **Log in to the system as superuser.**

2. Enable the auxiliary console.

```
# consadm -a devicename
```

3. Verify that the current connection is the auxiliary console.

```
# consadm
```

Example—Enabling an Auxiliary (Remote) Console

```
# consadm -a /dev/term/a  
# consadm  
/dev/term/a
```

▼ How to Display a List of Auxiliary Consoles

1. Log in to the system as superuser.

2. Select one of the following steps:

a. Display the list of auxiliary consoles.

```
# consadm  
/dev/term/a
```

or

b. Display the list of persistent auxiliary consoles.

```
# consadm -p  
/dev/term/b
```

▼ How to Enable an Auxiliary (Remote) Console Across System Reboots

1. Log in to the system as superuser.
2. Enable the auxiliary console across system reboots.

```
# consadm -a -p devicename
```

This adds the device to the list of persistent auxiliary consoles.

3. Verify that the device has been added to the list of persistent auxiliary consoles.

```
# consadm
```

Example—Enabling an Auxiliary (Remote) Console Across System Reboots

```
# consadm -a -p /dev/term/a  
# consadm  
/dev/term/a
```

▼ How to Disable an Auxiliary (Remote) Console

1. Log in to the system as superuser.
2. Select one of the following steps:
 - a. Disable the auxiliary console.

```
# consadm -d devicename
```

or

- b. Disable the auxiliary console and remove it from the list of persistent auxiliary consoles.

```
# consadm -p -d devicename
```

3. Verify that the auxiliary console has been disabled.

```
# consadm
```

Example—Disabling an Auxiliary (Remote) Console

```
# consadm -d /dev/term/a  
# consadm
```

Improved System Boot and Error Message Format

This functionality was new in the Solaris 7 3/99 software release.

This supplements information on system boot and error messages found in “Viewing System Messages” in *System Administration Guide, Volume II*, and referenced in the *Solaris Transition Guide*.

The Solaris 7 3/99 release improves the system boot and error message format by providing a numeric identifier, module name, and time stamp to messages generated by the syslog logging facility. In addition, messages that were previously lost after a system panic and reboot are now saved.

The new message format can be enabled or disabled by setting the msgid property in the log.conf file. The new message format is not enabled by default. See the man page, log(7D), and the procedure below for more information.

For general information about system error logging, see the *System Administration Guide, Volume II*.

System Boot and Error Message Format Changes

If msgid is set to 0 in the log.conf file, there are no changes in the message format. If msgid is set to 1, there are two changes in the message format:

1. The text of the message is preceded by a message ID that looks like this:

[ID msgid facility.priority]

For example:

[ID 123456 kern.notice]

The *msgid* identifier is described in the man page *msgid(1M)*. The *facility* and *priority* identifiers are described in the man page, *syslog.conf(4)*.

2. If the message originated in the kernel, the kernel module name is displayed instead of just 'unix.'

Previous message format:

```
Oct 1 14:07:24 mars unix: alloc: /: file system full
```

New message format:

```
Oct 1 14:07:24 mars ufs: [ID 845546 kern.notice] alloc: /:
file system full
```

▼ How to Enable System Message IDs

1. **Become superuser.**
2. **Enable system message IDs by adding the following line to the /platform/'uname -i'/kernel/drv/log.conf file, if it exists. Otherwise, add the *msgid* property to the /kernel/drv/log.conf file.**

```
msgid=1
```

3. **Save and close the file.**
4. **Reboot the system by using the following command.**

```
# init 6
```

Note - To enable system message IDs without rebooting the system, use the following adb command:

```
# echo log_msgid/W1 | adb -kw
```

▼ How to Disable System Message IDs

1. **Become superuser.**
2. **Disable system message IDs by changing the *msgid* line in the /platform/'uname -i'/kernel/drv/log.conf file to the following, if it**

exists. Otherwise, change the `msgid` property to the `/kernel/drv/log.conf` file.

```
msgid=0
```

3. Save and close the file.

4. Reboot the system by using the following command.

```
# init 6
```

Note - To disable system message IDs without rebooting the system, use the following `adb` command:

```
# echo log_msgid/W0 | adb -kw
```

System Administration Notes

This chapter describes the following new system administration information:

- “Migrating to BIND 8.1” on page 47
- “AnswerBook2 Server Software Updates” on page 47
- “SPARC: Running cron Jobs With Power Management Software” on page 48

Note - For the most up-to-date man pages, use the `man` command. The Solaris 7 11/99 man pages include new feature information not found in the *Solaris 7 Reference Manual Collection*.

Migrating to BIND 8.1

The “Migration from BIND 4.9.x to BIND 8.1” section of the *Solaris Naming Setup and Configuration Guide* and the *Solaris Naming Administration Guide* has been updated for the Solaris 7 8/99 software release. The shell script name as well as the pathname have changed from `/etc/named-boot2conf` to `/usr/sbin/named-bootconf`.

AnswerBook2 Server Software Updates

This functionality is updated in the Solaris 7 8/99 software release.

The AnswerBook2 version 1.4.1 server software provides improved performance, a textual, rather than graphical, navigation interface, and the ability to view and search information by collection.

For detailed information about changes to the AnswerBook2 interface, see *Viewing Online Documentation Using the AnswerBook2 System*.

SPARC: Running cron Jobs With Power Management Software

This functionality was new in the Solaris 7 3/99 software release.

This information supersedes the information on cron jobs found in “Choosing to Use Power Management” in *Using Power Management*.

Beginning with the Solaris 7 release, cron jobs scheduled to run during the time a system is suspended with Power Management software are handled in the following way:

- The first cron job (the one scheduled to run at the earliest point during the time the system was suspended) is started as soon as the system resumes.
- The second and subsequent cron jobs that might have taken place during the time the system was suspended are not run at all.

Normally, the cron command identifies the next job to run and calculates the interval to wait until it is time to run that job. When a system resumes, the cron command calculates the interval until the first job has expired; then this already-queued cron job is run.

cron then rechecks its queue of jobs. It only considers jobs whose execution time has not yet passed, so any additional jobs whose run time is due during the time the system was suspended are not run.