System Administration Guide: Basic Administration

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817–1985–10
January 2005

041213@10536

Contents

Preface

System Administration Guide: Basic Administration is part of a set that includes a significant part of the Solaris™ system administration information. This guide contains information for both SPARC® based and x86 based systems.

This book assumes you have completed the following tasks:

- Installed the SunOS™ 5.10 Operating System (Solaris OS)
- Set up all the networking software that you plan to use

The SunOS 5.10 OS is part of the Solaris product family, which also includes many features, including the Solaris Common Desktop Environment (CDE). The SunOS 5.10 OS is compliant with AT&T's System V, Release 4 operating system.

For the Solaris 10 release, new features that might be interesting to system administrators are covered in sections called *What's New in ... ?* in the appropriate chapters.

Note – This Solaris release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC®, SPARC64, AMD64, Pentium, and Xeon EM64T. The supported systems appear in the *Solaris 10 Hardware Compatibility List* at `http://www.sun.com/bigadmin/hcl`. This document cites any implementation differences between the platform types.

In this document the term "x86" refers to 64-bit and 32-bit systems manufactured using processors compatible with the AMD64 or Intel Xeon/Pentium product families. For supported systems, see the *Solaris 10 Hardware Compatibility List*.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the Solaris 10 release. To use this book, you should have 1-2 years of UNIX® system administration experience. Attending UNIX system administration training courses might be helpful.

How the System Administration Volumes Are Organized

Here is a list of the topics that are covered by the volumes of the System Administration Guides.

Book Title	Topics
System Administration Guide: Basic Administration	User accounts and groups, server and client support, shutting down and booting a system, managing services, and managing software (packages and patches)
System Administration Guide: Advanced Administration	Printing services, terminals and modems, system resources (disk quotas, accounting, and crontabs), system processes, and troubleshooting Solaris software problems
System Administration Guide: Devices and File Systems	Removable media, disks and devices, file systems, and backing up and restoring data
System Administration Guide: IP Services	TCP/IP network administration, IPv4 and IPv6 address administration, DHCP, IPsec, IKE, Solaris IP filter, Mobile IP, IP network multipathing (IPMP), and IPQoS
System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)	DNS, NIS, and LDAP naming and directory services, including transitioning from NIS to LDAP and transitioning from NIS+ to LDAP
System Administration Guide: Naming and Directory Services (NIS+)	NIS+ naming and directory services
System Administration Guide: Network Services	Web cache servers, time-related services, network file systems (NFS and Autofs), mail, SLP, and PPP

Book Title	Topics
System Administration Guide: Security Services	Auditing, device management, file security, BART, Kerberos services, PAM, Solaris cryptographic framework, privileges, RBAC, SASL, and Solaris Secure Shell
System Administration Guide: Solaris Containers—Resource Management and Solaris Zones	Resource management topics projects and tasks, extended accounting, resource controls, fair share scheduler (FSS), physical memory control using the resource capping daemon (rcapd), and dynamic resource pools; virtualization using Solaris Zones software partitioning technology

Related Third-Party Web Site References

Note – Sun™ is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Accessing Sun Documentation Online

The docs.sun.com℠ 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 conventions 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 `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. Do *not* save changes yet.

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	`#`

General Conventions

Be aware of the following conventions used in this book.

- When following steps or using examples, be sure to type double-quotes (`"`), left single-quotes (`‘`), and right single-quotes (`’`) exactly as shown.

- The key referred to as Return is labeled Enter on some keyboards.
- The root path usually includes the `/sbin`, `/usr/sbin`, `/usr/bin`, and `/etc` directories, so the steps in this book show the commands in these directories without absolute path names. Steps that use commands in other, less common, directories show the absolute paths in the examples.
- The examples in this book are for a basic SunOS software installation without the Binary Compatibility Package installed and without `/usr/ucb` in the path.

Caution – If `/usr/ucb` is included in a search path, it should always be at the end of the search path. Commands like `ps` or `df` are duplicated in `/usr/ucb` with different formats and options from the SunOS commands.

CHAPTER 1

Solaris Management Tools (Road Map)

This chapter provides a roadmap to Solaris management tools.

What's New in Solaris Management Tools?

These tools are new or changed in the Solaris 10 release:

- `admintool` – Not available in this release
- Package and Patch Tool Enhancements
- Sun Patch Manager
- Solaris Print Manager

The following table provides a brief description of each tool and where to find more information about these tools.

TABLE 1–1 New or Changed Solaris Management Tools in the Solaris 10 Release

Solaris Administration Tool	Description	For More Information
`admintool`	This tool is no longer available. Alternative tools include the following: ■ Solaris Management Console to manage users and groups. ■ Solaris Product Registry to manage software. ■ Solaris Print Manager to manage printers. ■ Solaris Management Console to manage terminals and modems.	"Setting Up User Accounts (Task Map)" on page 101 "Managing Software With the Solaris Product Registry GUI (Task Map)" on page 264 "Setting Up Printing (Task Map)" in *System Administration Guide: Advanced Administration* "Setting Up Terminals and Modems With Serial Ports Tool (Overview)" in *System Administration Guide: Advanced Administration*
Package and Patch Tool Enhancements	In this release, the package and patch tools have been enhanced. You can now use the `pkchk` command with the `-P` option instead of `grep pattern /var/sadm/install/contents`. The `-P` option enables you to use a partial path.	"Package and Patch Tool Enhancements" on page 250
Sun Patch Manager	The following new features are included in this version of Sun Patch Manager: ■ PatchPro analysis engine ■ Local-mode command-line interface ■ Patch list operations	Chapter 18 Chapter 19 Chapter 20

TABLE 1–1 New or Changed Solaris Management Tools in the Solaris 10 Release *(Continued)*

Solaris Administration Tool	Description	For More Information
Solaris Print Manager	The expanded printer support includes the following new or modified features: ■ Support for raster image processor (RIP). ■ Support for PostScript Printer Description (PPD) files. ■ The new `-n` option to the `lpadmin` command, which enables you to specify a PPD file when creating a new print queue or modifying an existing print queue. ■ The `lpstat` command output will display the PPD for a print queue that was creating by specifying a PPD file.	"What's New in Printing?" in *System Administration Guide: Advanced Administration*

Matrix of Solaris Management Tools and Supported Releases

This section provides information about tools that are primarily used to manage users, groups, clients, disks, printers, and serial ports.

This table lists the various Solaris management GUI tools and whether they are currently supported.

TABLE 1–2 Matrix of Solaris Management Tool Support

	Solaris 7	Solaris 8	Solaris 9	Solaris 10
`admintool`	Supported	Supported	Supported	Not supported
Solstice AdminSuite 2.3	Supported	Not supported	Not supported	Not supported
Solstice AdminSuite 3.0	Supported	Supported	Not supported	Not supported
Solaris Management Tools 1.0	Supported	Supported	Not supported	Not supported

TABLE 1–2 Matrix of Solaris Management Tool Support *(Continued)*

	Solaris 7	Solaris 8	Solaris 9	Solaris 10
Solaris Management Tools 2.0	Not supported	Supported (Solaris 8 01/01, 4/01, 7/01, 10/01, 2/02 releases only)	Not supported	Not supported
Solaris Management Tools 2.1	Not supported	Not supported	Supported	Supported

If you want to perform administration tasks on a system with a text-based terminal as the console, use Solaris Management Console commands instead. For more information, see Table 1–6.

Feature Descriptions for Solaris 10 Management Tools

This table describes the tools available in the Solaris 10 release.

TABLE 1–3 Feature Descriptions for Solaris 10 Management Tools

Feature or Tool	Supported in Solaris Management Console 2.1?
Computers and Networks tool	Yes
Diskless Client support	Yes, a diskless client command-line interface is available
Disks tool	Yes
Enhanced Disk tool (Solaris Volume Manager)	Yes
Job Scheduler tool	Yes
Log Viewer tool	Yes
Mail Alias support	Yes
Mounts and Shares tool	Yes
Name Service support	For users, groups, and network information only
Patch tool	Yes
Performance tool	Yes

TABLE 1–3 Feature Descriptions for Solaris 10 Management Tools *(Continued)*

Feature or Tool	Supported in Solaris Management Console 2.1?
Printer support	No, but Solaris Print Manager is available as a separate tool
Projects tool	Yes
role-based access control (RBAC) support	Yes
RBAC Tool	Yes
Serial Port tool	Yes
Software Package tool	No
System Information tool	Yes
User/Group tool	Yes

Feature Descriptions for Solaris 9 Management Tools

This table describes the tools available in the Solaris 9 releases.

TABLE 1–4 Feature Descriptions for Solaris 9 Management Tools

Feature or Tool	Supported in `admintool`?	Supported in Solaris Management Console 2.1?
Computers and Networks tool	No	Yes
Diskless Client support	No	Yes, a diskless client command-line interface is available
Disks tool	No	Yes
Enhanced Disk tool (Solaris Volume Manager)	No	Yes
Job Scheduler tool	No	Yes
Log Viewer tool	No	Yes
Mail Alias support	No	Yes
Mounts and Shares tool	No	Yes

TABLE 1–4 Feature Descriptions for Solaris 9 Management Tools *(Continued)*

Feature or Tool	Supported in `admintool`?	Supported in Solaris Management Console 2.1?
Name Service support	No	For users, groups, and network information only
Patch tool	No	Yes
Performance tool	No	Yes
Printer support	Yes	No, but Solaris Print Manager is available as a separate tool
Projects tool	No	Yes
RBAC support	No	Yes
RBAC tool	No	Yes
Serial Port tool	Yes	Yes
Software Package tool	Yes	No
System Information tool	No	Yes
User/Group tool	Yes	Yes

Feature Descriptions for Solaris 8 Management Tools

The following table lists the tools that are available in various Solaris 8 releases.

TABLE 1–5 Feature Descriptions for Solaris 8 Management Tools

Feature or Tool	Supported in `admintool`?	Supported in Solstice AdminSuite 3.0? (Solaris 8 and Solaris 8 6/00 and 10/00 only)	Supported in Solaris Management Console 1.0?	Supported in Solaris Management Console 2.0? (Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02 only)
Diskless Client support	No	No	No	No, but a diskless command- line interface is available as a separate tool

TABLE 1–5 Feature Descriptions for Solaris 8 Management Tools *(Continued)*

Feature or Tool	Supported in `admintool`?	Supported in Solstice AdminSuite 3.0? (Solaris 8 and Solaris 8 6/00 and 10/00 only)	Supported in Solaris Management Console 1.0?	Supported in Solaris Management Console 2.0? (Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02 only)
Disks tool	No	No	No	Yes
Job Scheduler tool	No	No	No	Yes
Log Viewer tool	No	Yes	No	Yes
Mail Alias support	No	Yes	No	Yes
Mounts and Shares tool	No	Yes	No	Yes
Name Service support	No	Yes	No	For users, groups, and network information only
Printer support	Yes	No, but Solaris Print Manager is available as a separate tool	Yes	No, but Solaris Print Manager is available as a separate tool
Software Package tool	Yes	No	Yes	No
RBAC support	No	Yes (rights support only)	No	Yes
RBAC tool	No	No, but RBAC command-line interface is available as a separate tool	No	Yes
Serial Port tool	Yes	Yes	Yes	Yes
User/Group tool	Yes	Yes	Yes	Yes

Availability of Solaris Management Commands

This series of tables lists commands that perform the same tasks as the Solaris management tools. For information on diskless client support, see Chapter 7.

Solaris 10 System Management Commands

This table describes the commands that provide the same functionality as the Solaris management tools. You must be superuser or assume an equivalent role to use these commands. Some of these commands are for the local system only. Others commands operate in a name service environment. See the appropriate man page and refer to the `-D` option.

TABLE 1–6 Descriptions for Solaris Management Commands

Command	Description	Man Page
`smc`	Starts the Solaris Management Console	`smc`(1M)
`smcron`	Manages `crontab` jobs	`smcron`(1M)
`smdiskless`	Manages diskless client support	`smdiskless`(1M)
`smexec`	Manages entries in the `exec_attr` database	`smexec`(1M)
`smgroup`	Manages group entries	`smgroup`(1M)
`smlog`	Manages and views WBEM log files	`smlog`(1M)
`smmultiuser`	Manages bulk operations on multiple user accounts	`smmultiuser`(1M)
`smosservice`	Adds Operating System (OS) services and diskless client support	`smosservice`(1M)
`smprofile`	Manages profiles in the `prof_attr` and `exec_attr` databases	`smprofile`(1M)
`smrole`	Manages roles and users in role accounts	`smrole`(1M)
`smserialport`	Manages serial ports	`smserialport`(1M)
`smuser`	Manages user entries	`smuser`(1M)

This table describes the commands you can use to manage RBAC from the command line. You must be superuser or assume an equivalent role to use these commands. These commands cannot be used to manage RBAC information in a name service environment.

TABLE 1–7 RBAC Command Descriptions

Command	Description	References
`auths`	Displays authorizations granted to a user	`auths`(1)
`profiles`	Displays execution profiles for a user	`profiles`(1)
`roleadd`	Adds a new role to the system	`roleadd`(1M)
`roles`	Displays roles granted to a user	`roles`(1)

This table describes the commands you can use to manage users, groups, and RBAC features from the command line. You must be superuser or assume an equivalent role to use these commands. These commands cannot be used to manage user and group information in a name service environment.

TABLE 1–8 Solaris User/Group Command Descriptions

Command	Description	References
`useradd`, `usermod`, `userdel`	Adds, modifies, or removes a user	`useradd`(1M), `usermod`(1M), `userdel`(1M)
`groupadd`, `groupmod`, `groupdel`	Adds, modifies, or removes a group	`groupadd`(1M), `groupmod`(1M), `groupdel`(1M)

Solaris 8 System Management Commands

All of the commands that are listed in Table 1–7 and Table 1–8 are available in the Solaris 8 release.

For More Information About Solaris Management Tools

This table identifies where to find more information about Solaris management tools.

TABLE 1–9 For More Information About Solaris Management Tools

Tool	Availability	For More Information
Solaris Management Console 2.1 suite of tools	Solaris 9 and 10 releases	This guide and the console online help
Solaris Management Console 2.0 suite of tools	Solaris 8 1/01, 4/01, 7/01, 10/01, and 2/02 releases	Solaris Management Console online help
`admintool`	Solaris 9 and previous Solaris releases	`admintool`
AdminSuite 3.0	Solaris 8, Solaris 8 6/00, and Solaris 8 10/00 releases	*Solaris Easy Access Server 3.0 Installation Guide*
Diskless Client command-line interface	Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02, and Solaris 9 and 10 releases	Chapter 7

CHAPTER 2

Working With the Solaris Management Console (Tasks)

This chapter describes the Solaris management tools that are used to perform system administration tasks. Topics include starting the Solaris Management Console (console), setting up role-based access control (RBAC) to use with the console, and working with the Solaris management tools in a name service environment.

For information on the procedures associated with performing system management tasks by using the Solaris Management Console, see these task maps:

- "Using the Solaris Management Tools With RBAC (Task Map)" on page 40
- "Using the Solaris Management Tools in a Name Service Environment (Task Map)" on page 46

For information on troubleshooting Solaris Management Console problems, see "Troubleshooting the Solaris Management Console" on page 53.

Solaris Management Console (Overview)

The following sections provide information about the Solaris Manager Console.

What Is the Solaris Management Console?

The Solaris Management Console is a container for GUI-based management tools that are stored in collections referred to as *toolboxes*. The console includes a default toolbox with many basic management tools, including tools for managing the following:

- Users
- Projects
- `cron` jobs for mounting and sharing file systems

- `cron` jobs for managing disks and serial ports

For a brief description of each Solaris management tool, see Table 2–1.

You can add tools to the existing toolbox, or you can create new toolboxes.

The Solaris Management Console has three primary components:

- **The Solaris Management Console client**

  Called the *console*, this component is the visible interface and contains the GUI tools used to perform management tasks.

- **The Solaris Management Console server**

  This component is located either on the same machine as the console or remotely. This component provides all the *back-end* functionality that allows management through the console.

- **The Solaris Management Console toolbox editor**

  This application, which looks similar to the console, is used to add or modify toolboxes, to add tools to a toolbox, or to extend the scope of a toolbox. For example, you could add a toolbox to manage a name service domain.

The default toolbox is visible when you start the console.

Solaris Management Console Tools

This table describes the tools included in the default Solaris Management Console toolbox. Cross-references to background information for each tool are provided.

TABLE 2–1 Solaris Management Console Tool Suite

Category	Tool	Description	For More Information
System Status	System Information	Monitors and manages system information such as date, time, and time zone	Chapter 12, "Displaying and Changing System Information (Tasks)," in *System Administration Guide: Advanced Administration*
	Log Viewer	Monitors and manages the Solaris Management Console tools log and system logs	Chapter 21, "Troubleshooting Software Problems (Overview)," in *System Administration Guide: Advanced Administration*

TABLE 2–1 Solaris Management Console Tool Suite *(Continued)*

Category	Tool	Description	For More Information
	Processes	Monitors and manages system processes	"Processes and System Performance" in *System Administration Guide: Advanced Administration*
	Performance	Monitors system performance	Chapter 18, "Managing System Performance (Overview)," in *System Administration Guide: Advanced Administration*
System Configuration	Users	Manages users, rights, roles, groups, and mailing lists	"What Are User Accounts and Groups?" on page 72 and "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*
	Projects	Creates and manages entries in the `/etc/project` database	Chapter 2, "Projects and Tasks (Overview)," in *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*
	Computers and Networks	Creates and monitors computer and network information	Solaris Management Console online help
	Patches	Manages patches	Chapter 18
Services	Scheduled Jobs	Creates and manages scheduled `cron` jobs	"Ways to Automatically Execute System Tasks" in *System Administration Guide: Advanced Administration*
Storage	Mounts and Shares	Mounts and shares file systems	Chapter 18, "Mounting and Unmounting File Systems (Tasks)," in *System Administration Guide: Devices and File Systems*
	Disks	Creates and manages disk partitions	Chapter 11, "Managing Disks (Overview)," in *System Administration Guide: Devices and File Systems*
	Enhanced Storage	Creates and manages volumes, hot spare pools, state database replicas, and disk sets	*Solaris Volume Manager Administration Guide*

TABLE 2–1 Solaris Management Console Tool Suite *(Continued)*

Category	Tool	Description	For More Information
Devices and Hardware	Serial Ports	Sets up terminals and modems	Chapter 8, "Managing Terminals and Modems (Overview)," in *System Administration Guide: Advanced Administration*

Context–sensitive help is available after you start a tool. For broader, more in-depth online information than the context help provides, see the expanded help topics. You can access these help topics from the console Help menu.

Why Use the Solaris Management Console?

The console provides a set of tools with many benefits for administrators. The console does the following:

- **Supports all experience levels**

  Inexperienced administrators can complete tasks by using the graphical user interface (GUI), which includes dialog boxes, wizards, and context help. Experienced administrators find that the console provides a convenient, secure alternative to using `vi` to manage hundreds of configuration parameters spread across dozens or hundreds of systems.

- **Controls user access to the system**

  Although any user can access the console by default, only superuser can make changes in the initial configuration. As described in "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*, it is possible to create special user accounts called *roles* can be created and assigned to users, typically administrators, who are permitted to make specific system changes.

  The key benefit of RBAC is that roles can be limited so that users have access to only those tasks that are necessary for doing their jobs. RBAC is *not* required for using the Solaris management tools. You can run all tools as superuser without making any changes.

- **Provides a command line interface**

  If preferred, administrators can operate the Solaris management tools through a command-line interface (CLI). Some commands are written specifically to mimic the GUI tool functions, such as the commands for managing users. These new commands are listed in Table 1–6, which includes the names and brief descriptions of each command. There is also a man page for each command.

  For Solaris management tools that have no special commands, such as the Mounts and Shares tool, use the standard UNIX commands.

For in-depth information about how RBAC works, its benefits, and how to apply those benefits to your site, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

To learn more about using RBAC with the Solaris management tools, see "Using the Solaris Management Tools With RBAC (Task Map)" on page 40.

Organization of the Solaris Management Console

In the following figure, the console is shown with the Users tool open.

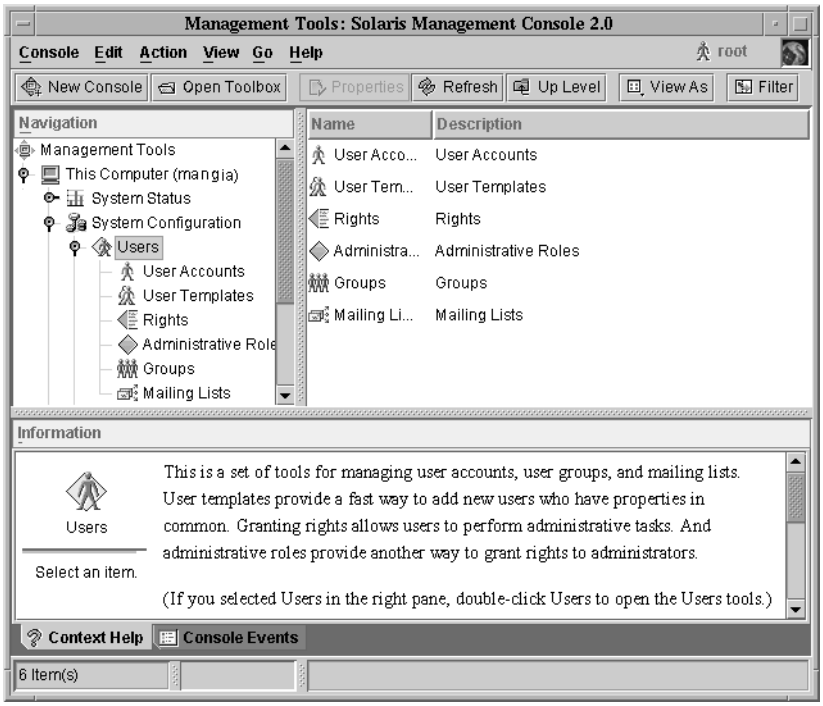

FIGURE 2–1 Solaris Management Console – Users Tool

The main part of the console consists of three panes:

- **Navigation pane** (at the left) – For accessing tools (or sets of tools), folders, or other toolboxes. Icons in the navigation pane are called *nodes* and are expandable if they are folders or toolboxes.
- **View pane** (at the right) – For viewing information related to the node selected in the navigation pane. The view pane shows either the contents of the selected folder, subordinate tools, or the data associated with the selected tool.
- **Information pane** (at the bottom) – For displaying context-sensitive help or console events.

Changing the Solaris Management Console Window

The layout of the console window is highly configurable. You can use the following features to change the console window layout:

- **View menu** – Use the Show option in the View menu to hide or display the optional bars and panes. The other options in the View menu control the display of nodes in the view pane.
- **Console menu** – Use the Preferences option to set the following: the initial toolbox, the orientation of panes, clicking or double-clicking for selection, text or icons in the tool bar, fonts, default tool loading, authentication prompts, and advanced logins.
- **Context Help or Console Events toggles** – Use the icons at the bottom of the information pane to toggle between the display of context-sensitive help and console events.

Solaris Management Console Documentation

The main source of documentation for using the console and its tools is the online help system. Two forms of online help are available: context-sensitive help and expanded help topics.

- **Context-sensitive help responds to your use of the console tools.**

  Clicking the cursor on tabs, entry fields, radio buttons, and so forth, causes the appropriate help to appear in the Information pane. You can close, or reopen the Information pane by clicking the question mark button on dialog boxes and wizards.
- **Expanded help topics are available from the Help menu or by clicking cross reference links in some context-sensitive help.**

  These topics appear in a separate viewer and contain more in-depth information than is provided by the context help. Topics include overviews of each tool, explanations of how each tool works, files used by a specific tool, and troubleshooting.

For a brief overview of each tool, refer to Table 2–1.

How Much Role-Based Access Control?

As described in "Why Use the Solaris Management Console?" on page 34, a major advantage of using the Solaris management tools is the ability to use Role-Based Access Control (RBAC). RBAC provides administrators with access to just the tools and commands they need to perform their jobs.

Depending on your security needs, you can use varying degrees of RBAC.

RBAC Approach	Description	For More Information
No RBAC	Allows you to perform all tasks as superuser. You can log in as yourself. When you select a Solaris management tool, you specify `root` as the user and the `root` password.	"How to Become Superuser (root) or Assume a Role" on page 39
`root` as a role	Eliminates anonymous `root` logins and prevents users from logging in as `root`. This approach requires users to log in as themselves before they assume the `root` role. Note that you can apply this approach whether or not you are using other roles.	"How to Plan Your RBAC Implementation" in *System Administration Guide: Security Services*
Single role only	Uses the Primary Administrator role, which is roughly equivalent to having `root` access only.	"Creating the Primary Administrator Role" on page 42
Suggested roles	Uses three roles that are easily configured: Primary Administrator, System Administrator, and Operator. These roles are appropriate for organizations with administrators at different levels of responsibility whose job capabilities roughly fit the suggested roles.	"Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*

RBAC Approach	Description	For More Information
Custom roles	You can add your own roles, depending on your organization's security needs.	"Managing RBAC" in *System Administration Guide: Security Services* and "How to Plan Your RBAC Implementation" in *System Administration Guide: Security Services*

Becoming Superuser (`root`) or Assuming a Role

Most administration tasks, such as adding users, file systems, or printers, require that you first log in as `root` (UID=0) or assume a role if you are using RBAC. The `root` account, also known as the *superuser* account, is used to make system changes and can override user file protection in emergency situations.

The superuser account and roles should be used only to perform administrative tasks to prevent indiscriminate changes to the system. The security problem associated with the superuser account is that a user has complete access to the system even when performing minor tasks.

In a non-RBAC environment, you can either log in to the system as superuser or use the `su` command to change to the superuser account. If RBAC is implemented, you can assume roles through the console or use `su` and specify a role.

When you use the console to perform administration tasks, you can do one of the following:

- Log in to the console as yourself and then supply the `root` user name and password
- Log in to the console as yourself and then assume a role

A major benefit of RBAC is that roles can be created to give limited access to specific functions only. If you are using RBAC, you can run restricted applications by assuming a role rather than by becoming superuser.

For step-by-step instructions on creating the Primary Administrator role, see "How to Create the First Role (Primary Administrator)" on page 43. For an overview on using RBAC, see Chapter 9, "Using Role-Based Access Control (Tasks)," in *System Administration Guide: Security Services*.

▼ How to Become Superuser (root) or Assume a Role

Become superuser or assume a role by using one of the following methods. Each method requires that you know either the superuser password or the role password.

Steps

1. **Become superuser. Select one of the following methods to become superuser:**

   - Log in as a user, start the Solaris Management Console, select a Solaris management tool, and then log in as `root`.

     This method enables to you perform any management task from the console.

     For information on starting the Solaris Management Console, see "How to Start the Solaris Management Console in a Name Service Environment" on page 51.

   - Log in as superuser on the system console.

     ```
     hostname console: root
     Password: root-password
     #
     ```

     The pound sign (#) is the Bourne shell prompt for the superuser account.

     This method provides complete access to all system commands and tools.

   - Log in as a user, and then change to the superuser account by using the `su` command at the command line.

     ```
     % su
     Password: root-password
     #
     ```

     This method provides complete access to all system commands and tools.

   - Log in remotely as superuser.

     This method is not enabled by default. You must modify the `/etc/default/login` file to remotely log in as superuser on the system console. For information on modifying this file, see Chapter 3, "Controlling Access to Systems (Tasks)," in *System Administration Guide: Security Services*.

     This method provides complete access to all system commands and tools.

2. **Assume a role. Select one of the following methods to assume a role:**

   - Log in as user, and then change to a role by using the `su` command at the command line.

     ```
     % su role
     Password: role-password
     $
     ```

     This method provides access to all the commands and tools that the role has access to.

   - Log in as a user, start the Solaris Management Console, select a Solaris management tool, and then assume a role.

For information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44.

This method provides access to the Solaris management tools that the role has access to.

Using the Solaris Management Tools With RBAC (Task Map)

This task map describes the tasks to do if you want to use the RBAC security features rather than the superuser account to perform administration tasks.

Note – The information in this chapter describes how to use the console with RBAC. RBAC overview and task information is included to show you how to initially set up RBAC with the console.

For detailed information on RBAC and how to use it with other applications, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

Task	Description	For Instructions
1. Start the console.	If your user account is already set up, start the console as yourself. Then, log in to the console as `root`. If you do not have a user account set up, become superuser first, and then start the console.	"How to Start the Console as Superuser or as a Role" on page 44
2. Add a user account for yourself.	Add a user account for yourself, if you do not have an account already.	Solaris Management Console online help "If You Are the First to Log in to the Console" on page 41
3. Create the Primary Administrator role	Create the Primary Administrator role. Then, add yourself to this role.	"How to Create the First Role (Primary Administrator)" on page 43

Task	Description	For Instructions
4. Assume the Primary Administrator role.	Assume the Primary Administrator role after you have created this role.	"How to Assume the Primary Administrator Role" on page 44
5. (Optional) Make `root` a role.	Make `root` a role and add yourself to the root role so that no other user can use the `su` command to become `root`.	"How to Plan Your RBAC Implementation" in *System Administration Guide: Security Services*
6. (Optional) Create other administrative roles.	Create other administrative roles and grant the appropriate rights to each role. Then, add the appropriate users to each role.	Chapter 9, "Using Role-Based Access Control (Tasks)," in *System Administration Guide: Security Services*

The following sections provide overview information and step-by-step instructions for using the Solaris Management Console and the RBAC security features.

If You Are the First to Log in to the Console

If you are the first administrator to log in to the console, start the console as a user (yourself). Then, log in as superuser. This method gives you complete access to all the console tools.

Here are the general steps, depending on whether you are using RBAC:

- *Without RBAC* – If you choose not to use RBAC, continue working as superuser. All other administrators will also need `root` access to perform their jobs.
- *With RBAC* – You'll need to do the following:
  - Set up your user account, if you do not already have an account.
  - Create the role called Primary Administrator.
  - Assign the Primary Administrator right to the role that you are creating.
  - Assign your user account to this role.

    For step-by-step instructions on creating the Primary Administrator role, see "How to Create the First Role (Primary Administrator)" on page 43.

    For an overview on using RBAC, see Chapter 9, "Using Role-Based Access Control (Tasks)," in *System Administration Guide: Security Services*.

Creating the Primary Administrator Role

An *administrator role* is a special user account. Users who assume a role are permitted to perform a predefined set of administrative tasks.

The Primary Administrator role is permitted to perform all administrative functions, similar to superuser.

If you are superuser, or a user assuming the Primary Administrator role, you can define which tasks other administrators are permitted to perform. With the help of the Add Administrative Role wizard, you can create a role, grant rights to the role, and then specify which users are permitted to assume that role. A *right* is a named collection of commands, or authorizations, for using specific applications. A right enables you to perform specific functions within an application. The use of rights can be granted or denied by an administrator.

You are prompted for the following information when you create the Primary Administrator role.

TABLE 2–2 Field Descriptions for Adding a Role by Using the Solaris Management Console

Field name	Description
Role name	Selects the name an administrator uses to log in to a specific role.
Full name	Provides a full, descriptive name of this role. (Optional)
Description	Provides further description of this role.
Role ID number	Selects the identification number assigned to this role. This number is the same as the set of identifiers for UIDs.
Role shell	Selects the shell that runs when a user logs in to a terminal or console window and assumes a role in that window.
Create a role mailing list	Creates a mailing list with the same name as the role, if checked. You can use this list to send email to everyone assigned to the role.
Role password and confirm Password	Sets and confirms the role password.
Available rights and granted Rights	Assigns rights to this role by choosing from the list of Available Rights and adding them to the list of Granted Rights.
Select a home directory	Selects the home directory server where this role's private files will be stored.
Assign users to this role	Adds specific users to the role so that they can assume the role to perform specific tasks.

For detailed information about role-based access control, and instructions on how to use roles to create a more secure environment, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

▼ How to Create the First Role (Primary Administrator)

This procedure describes how to create the Primary Administrator role and then assign it to your user account. This procedure assumes that your user account is already created.

Steps

1. **Start the console as yourself.**

   ```
   % /usr/sadm/bin/smc &
   ```

   For additional information on starting the console, see "How to Start the Console as Superuser or as a Role" on page 44.

   The console online help provides more information about creating a user account for yourself.

2. **Click on the This Computer icon in the Navigation pane.**
3. **Click on System Configuration->Users -> Administrative Roles.**
4. **Click Action->Add Administrative Role.**

   The Add Administrative Role wizard opens.

5. **Create the Primary Administrator role with the Administrative Role wizard by following these steps.**

   a. **Identify the role name, full role name, description, role ID number, role shell, and whether you want to create a role mailing list. Click Next.**

   b. **Set and confirm the role password. Click Next.**

   c. **Select the Primary Administrator right from the Available Rights column and add it to Granted Rights column. Click Next.**

   d. **Select the home directory for the role. Click Next.**

   e. **Assign yourself to the list of users who can assume the role. Click Next.**

   If necessary, see Table 2–2 for a description of the role fields.

6. **Click Finish.**

▼ How to Assume the Primary Administrator Role

After you have created the Primary Administrator role, log in to the console as yourself, and then assume the Primary Administrator role.

When you assume a role, you take on all the attributes of that role, including the rights. At the same time, you relinquish all of your own user properties.

Steps

1. **Start the console.**

   ```
   % /usr/sadm/bin/smc &
   ```

   For information on starting the console, see "How to Start the Console as Superuser or as a Role" on page 44.

2. **Log in with your user name and password.**

   A list shows which roles you are permitted to assume.

3. **Log in to the Primary Administrator role and provide the role password.**

Starting the Solaris Management Console

The following procedure describes how to start the console and gain access to the Solaris management tools.

For instructions on what to do if you are the first user to log in to the console, see "If You Are the First to Log in to the Console" on page 41.

▼ How to Start the Console as Superuser or as a Role

If you start the console as a user with your own user account, you have limited access to the Solaris management tools. For greater access, you can log in as yourself and then log in as one of the roles you are allowed to assume. If you are permitted to assume the role of Primary Administrator, you then have access to all the Solaris management tools. This role is equivalent to that of superuser.

Steps

1. **Verify that you are in a window environment, such as the CDE environment.**

2. **Start the console in one of the following ways:**

   - From the command line, type the following command:

```
% /usr/sadm/bin/smc &
```

It might take a minute or two for the console to come up the first time.

- Start the console from the Tools menu of the CDE front panel.
- Double-click the Solaris Management Console icon in CDE's Applications Manager or File Manager.

The Solaris Management Console window is displayed.

Note – Open a console in your window environment to display the Solaris Management Console startup messages. Do not attempt to start the Solaris Management Console server manually before starting the Solaris Management Console. The server starts automatically when you start the Solaris Management Console. For information on troubleshooting console problems, see "Troubleshooting the Solaris Management Console" on page 53.

---

3. **Double-click the This Computer icon under the Management Tools icon in the Navigation pane.**

   A list of categories is displayed.

4. **(Optional) Select the appropriate toolbox.**

   If you want to use a toolbox other than the default toolbox, select the appropriate toolbox from the Navigation pane. Or, select Open Toolbox from the console menu and load the toolbox you want.

   For information about using different toolboxes, see "How to Create a Toolbox for a Specific Environment" on page 49.

5. **Double-click the category icon to access a particular tool.**

   Use the online help to identify how to perform a specific task.

6. **Double-click the tool icon.**

   A pop-up Log-In window is displayed.

7. **Decide if you want to use the tool as superuser or as a role. If you are logging in a as superuser, enter the `root` password.**

8. **If you are logging in as yourself, backspace over the `root` user name. Then supply your user ID and user password.**

   A list of roles you can assume is displayed.

9. **Select the Primary Administrator role, or an equivalent role, and supply the role password.**

   For step-by-step instructions on creating the Primary Administrator role, see "How to Create the First Role (Primary Administrator)" on page 43.

   The main tool menu is displayed.

Using the Solaris Management Tools in a Name Service Environment (Task Map)

By default, the Solaris management tools are set up to operate in a local environment. For example, the Mounts and Shares tool enables you to mount and share directories on specific systems, but not in an NIS or NIS+ environment. However, you can manage information with the Users and Computers and Networks tools in a name service environment.

To work with a console tool in a name service environment, you need to create a name service toolbox, and then add the tool to that toolbox.

Task	Description	For Instructions
1. Verify prerequisites.	Verify you have completed the prerequisites before attempting to use the console in a name service environment.	"Prerequisites for Using the Solaris Management Console in a Name Service Environment" on page 48
2. Create a toolbox for the name service.	Use the New Toolbox wizard to create a toolbox for your name service tools.	"How to Create a Toolbox for a Specific Environment" on page 49
3. Add a tool to the name service toolbox.	Add the Users tool, or any other name service tool, to your name service toolbox.	"How to Add a Tool to a Toolbox" on page 50
4. Select the toolbox that was just created.	Select the toolbox you just created to manage name service information.	"How to Start the Solaris Management Console in a Name Service Environment" on page 51

RBAC Security Files

The RBAC security files that work with the Solaris Management Console are created when you upgrade to or install the Solaris 9 or Solaris 10 release. If you do not install the Solaris Management Console packages, the RBAC security files are installed without the necessary data for using RBAC. For information on the Solaris Management Console packages, see "Troubleshooting the Solaris Management Console" on page 53.

The RBAC security files in the Solaris 9 or Solaris 10 release are included in your name service so that you can use the Solaris Management Console tools in a name service environment.

The security files on a local server are populated into a name service environment as part of a standard upgrade by the `ypmake`, `nispopulate`, or equivalent LDAP commands. The following name services are supported:

- NIS
- NIS+
- LDAP
- files

Note – The `projects` database is not supported in the NIS+ environment.

The RBAC security files are created when you upgrade to or install the Solaris 9 or 10 release.

This table briefly describes the predefined security files that are installed on a Solaris 9 or 10 system.

TABLE 2–3 RBAC Security Files

Local File Name	Table or Map Name	Description
`/etc/user_attr`	`user_attr`	Associates users and roles with authorizations and rights profiles
`/etc/security/auth_attr`	`auth_attr`	Defines authorizations and their attributes and identifies associated help files
`/etc/security/prof_attr`	`prof_attr`	Defines rights profiles, lists the rights profiles assigned to the authorizations, and identifies associated help files
`/etc/security/exec_attr`	`exec_attr`	Defines the privileged operations assigned to a rights profile

For unusual upgrade cases, you might have to use the `smattrpop` command to populate RBAC security files in the following instances:

- When creating or modifying rights profiles
- When you need to include users and roles by customizing the `usr_attr` file

For more information, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

Prerequisites for Using the Solaris Management Console in a Name Service Environment

The following table identifies what you need to do before you can use the Solaris Management Console in a name service environment.

Prerequisite	For More Information
Install the Solaris 9 or 10 release.	*Solaris 10 Installation Guide: Basic Installations*
Set up your name service environment.	*System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
Select your management scope.	"Management Scope" on page 48
Make sure your `/etc/nsswitch.conf` file is configured so that you can access your name service data.	"`/etc/nsswitch.conf` File" on page 48

Management Scope

The Solaris Management Console uses the term *management scope* to refer to the name service environment that you want to use with the selected management tool. The management scope choices for the Users tool and the Computers and Networks tool are LDAP, NIS, NIS+, or files.

The management scope that you select during a console session should correspond to the primary name service identified in the `/etc/nsswitch.conf` file.

`/etc/nsswitch.conf` File

The `/etc/nsswitch.conf` file on each system specifies the policy for name service lookups (where data is read from) on that system.

Note – You must make sure that the name service accessed from the console, which you specify through the console Toolbox Editor, appears in the search path of the `/etc/nsswitch.conf` file. If the specified name service does not appear there, the tools might behave in unexpected ways, resulting in errors or warnings.

When you use the Solaris management tools in a name service environment, you might impact many users with a single operation. For example, if you delete a user in the NIS name service, that user is deleted on all systems that are using NIS.

If different systems in your network have different `/etc/nsswitch.conf` configurations, unexpected results might occur. So, all systems to be managed with the Solaris management tools should have a consistent name service configuration.

▼ How to Create a Toolbox for a Specific Environment

Applications for administering the Solaris Operating System are called tools. Those tools are stored in collections referred to as *toolboxes*. A toolbox can be located on a local server, where the console is located, or on a remote machine.

Use the Toolbox Editor to add a new toolbox, to add tools to an existing toolbox, or to change the scope of a toolbox. For example, use this tool to change the domain from local files to a name service.

Note – You can start the Toolbox Editor as a normal user. However, if you plan to make changes and save them to the default console toolbox, `/var/sadm/smc/toolboxes`, you must start the Toolbox Editor as `root`.

Steps

1. **Start the Toolbox Editor.**

   ```
   # /usr/sadm/bin/smc edit &
   ```

2. **Select Open from the Toolbox menu.**
3. **Select the This Computer icon in the Toolboxes: window.**
4. **Click Open.**

   The This Computer toolbox opens in the window.

5. **Select the This Computer icon again in the Navigation pane.**
6. **Select Add Folder from the Action menu.**
7. **Use the Folder wizard to add a new toolbox for your name service environment.**

   a. **Name and Description – Provide a name in the Full Name window. Click Next.**

      For example, provide “NIS tools” for the NIS environment.

   b. **Provide a description in the Description window. Click Next.**

      For example, “tools for NIS environment” is an appropriate example.

   c. **Icons – Use the default value for the Icons. Click Next.**

   d. **Management Scope – Select Override.**

e. **Select your name service under the Management Scope pull-down menu.**

f. **Add the name service master name in the Server field, if necessary.**

g. **Add the domain managed by the server in the Domain field.**

h. **Click Finish.**
   The new toolbox appears in the left Navigation pane.

8. **Select the new toolbox icon.**

9. **Select Save As from the Toolbox menu.**

10. **Enter the toolbox path name in the Local Toolbox Filename dialog box. Use the `.tbx` suffix.**

    `/var/sadm/smc/toolboxes/this_computer/`*toolbox-name*`.tbx`

11. **Click Save.**
    The new toolbox appears in the Navigation pane in the console window.

See Also After you have created a name service toolbox, you can put a name service tool into it. For more information, see "How to Add a Tool to a Toolbox" on page 50.

▼ How to Add a Tool to a Toolbox

In addition to the default tools that ship with the console, additional tools that can be launched from the console are being developed. As these tools become available, you can add one or more tools to an existing toolbox.

You can also create a new toolbox, for either local management or network management. Then, you can add tools to the new toolbox.

Steps

1. **Become superuser or assume an equivalent role.**
   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start the Toolbox Editor, if necessary.**

   ```
   # /usr/sadm/bin/smc edit &
   ```

3. **Select the toolbox.**
   If you want to work in a name service, select the toolbox you just created in the Toolbox Editor. For more information, see "How to Create a Toolbox for a Specific Environment" on page 49.

4. **Select Add Tool from the Action menu.**

5. **Use the Add Tool wizard to add the new tool.**

   a. **Server Selection – Add the name service master in the Server window. Click Next.**

   b. **Tools Selection – Select the tool you want to add from the Tools window. Click Next.**

      If this toolbox is a name service toolbox, choose a tool you want to work in a name service environment. For example, choose the Users tool.

   c. **Name and Description – Accept the default values. Click Next.**

   d. **Icons – Accept the default values, unless you have created custom icons. Click Next.**

   e. **Management Scope – Accept the default value "Inherit from Parent." Click Next.**

   f. **Tool Loading – Accept the default "Load tool when selected." Click Finish.**

6. **Select Save from the Toolbox menu to save the updated toolbox.**

   The Local Toolbox window is displayed.

▼ How to Start the Solaris Management Console in a Name Service Environment

After you have created a name service toolbox and added tools to it, you can start the Solaris Management Console and open that toolbox to manage a name service environment.

Before You Begin

Verify that the following prerequisites are met:

- Ensure that the system you are logged in to is configured to work in a name service environment.
- Verify that the `/etc/nsswitch.conf` file is configured to match your name service environment.

Steps

1. **Start the Solaris Management Console.**

   For more information, see "How to Start the Console as Superuser or as a Role" on page 44.

2. **Select the toolbox you created for the name service, which appears in the Navigation pane.**

   For information on creating a toolbox for a name service, see "How to Create a Toolbox for a Specific Environment" on page 49.

Adding Tools to the Solaris Management Console

This section describes how to add legacy tools or unbundled tools to the console. If you want to add authentication to these tools, see "Managing RBAC" in *System Administration Guide: Security Services*.

▼ How to Add a Legacy Tool to a Toolbox

A legacy tool is any application that was not designed specifically as a Solaris management tool. You can add three types of legacy tool applications to a console toolbox: X applications, command-line interface, and HTML. Each tool you add to a toolbox can then be launched from the Solaris Management Console.

Steps

1. **Become superuser or assume an equivalent role.**
2. **Start the Solaris Management Console Toolbox Editor, if necessary.**

   ```
   # /usr/sadm/bin/smc edit &
   ```

3. **Open the toolbox to which you want to add the legacy application.**
   The toolbox selected is opened in the Toolbox Editor.
4. **Select the node in the toolbox to which you want to add the legacy application.**
   A legacy application can be added to the top node of a toolbox or to another folder.
5. **Click Action->Add Legacy Application.**
   The first panel of the Legacy Application Wizard: General is displayed.
6. **Follow the instructions in the wizard.**
7. **Save the toolbox in the Toolbox Editor.**

▼ How to Install an Unbundled Tool

Follow this procedure if you want to add a new tool package that can be launched from the Solaris Management Console.

Steps

1. **Become superuser or assume an equivalent role.**
2. **Install the new tool package.**

   ```
   # pkgadd ABCDtool
   ```

3. **Restart the console so that it recognizes the new tool.**

   a. **Stop the console server.**

      ```
      # /etc/init.d/init.wbem stop
      ```

   b. **Start the console server.**

      ```
      # /etc/init.d/init.wbem start
      ```

4. **Start the console to verify that the new tool is displayed.**

   For more information, see "How to Start the Console as Superuser or as a Role" on page 44.

Troubleshooting the Solaris Management Console

Before using this troubleshooting procedure, make sure that the following packages are installed:

- `SUNWmc` – Solaris Management Console 2.1 (Server Components)
- `SUNWmcc` – Solaris Management Console 2.1 (Client Components)
- `SUNWmccom` – Solaris Management Console 2.1 (Common Components)
- `SUNWmcdev` – Solaris Management Console 2.1 (Development Kit)
- `SUNWmcex` – Solaris Management Console 2.1 (Examples)
- `SUNWwbmc` – Solaris Management Console 2.1 (WBEM Components)

These packages provide the basic Solaris Management Console launcher. You must install the `SUNWCprog` cluster to use the Solaris Management Console and all of its tools.

▼ How to Troubleshoot the Solaris Management Console

The client and the server are started automatically when you start the Solaris Management Console.

If the console is visible and you are having trouble running the tools, it might be that the server might not be running. Or, the server might be in a problem state that can be resolved by stopping and restarting it.

Steps 1. **Become superuser or assume an equivalent role.**

2. **Determine whether the console server is running.**

   ```
   # /etc/init.d/init.wbem status
   ```

   If the console server is running, you should see a message similar the following:

   ```
   SMC server version 2.1.0 running on port 898.
   ```

3. **If the console server is not running, start it.**

   ```
   # /etc/init.d/init.wbem start
   ```

   After a short time, you should see a message similar to the following:

   ```
   SMC server is ready.
   ```

4. **If the server is running and you are still having problems, stop the console server. Then, restart it.**

   a. **Stop the console server.**

      ```
      # /etc/init.d/init.wbem stop
      ```

      You should see a message similar to the following:

      ```
      Shutting down SMC server on port 898.
      ```

   b. **Start the console server.**

      ```
      # /etc/init.d/init.wbem start
      ```

CHAPTER 3

Working With the Sun Java Web Console (Tasks)

This chapter describes the Sun Java Web Console, which is used to administer web-based Sun system management applications that are installed and registered on your system. Topics in this chapter include the following:

For information on the procedures that are associated with using the Java Web Console, see "Administering the Java Web Console (Task Map)" on page 57.

Java Web Console (Overview)

The Java Web Console provides a common location for users to access web-based system management applications. You access the web console by logging in through a secure `https` port with one of several supported web browsers. The single entry point that the web console provides eliminates the need to learn URLs for multiple applications. In addition, the single entry point provides authentication and authorization for all applications that are registered with the web console.

All web console-based applications conform to the same user interface guidelines, which enhances ease of use. The web console also provides auditing and logging services for all registered users.

What Is the Java Web Console?

The Java Web Console is a web page where you can find the Sun system management web-based applications that are installed and registered on your system. Any compliant J2EE™ web application can register with the web console to make itself available to authenticated and authorized users. Registration is automatically a part of the installation process. Thus, registration requires no administrator intervention.

The Java Web Console provides the following:

- **A Single point of entry for login and the launching of system management applications**

  The Java Web Console is Sun's current direction for system management applications. No compatibility exists between the Java Web Console and the Solaris Management Console. The Java Web Console is a J2EE based web application, and Solaris Management Console is a Java application. However, you can run both consoles on the same system at the same time.

- **Single sign-on through a secure https port**

  Single sign-on in this context means that you do not have to authenticate yourself to each management application after you authenticate yourself to the web console.

- **Dynamically organized and aggregated applications**

  Applications are installed and displayed in the category of management tasks that is most applicable. Categories include the following:

  - Systems
  - Storage
  - Services
  - Desktop applications
  - Other

- **A Common look and feel**

  All console-based applications use the same components and behavior, thereby reducing the learning curve for administrators.

- **Standard, extensible authentication, authorization, and auditing mechanisms**

  The Java Web Console supports Pluggable Authentication Module (PAM), role-based access control (RBAC) roles, and Basic Security Module (BSM) auditing.

The Java Web Console includes the following two management commands:

- `smcwebserver` – This command starts and stops the console's web server.
- `smreg` – This command registers applications and controls configuration properties.

For more information, see the `smcwebserver`(1M) and `smreg`(1M) man pages.

Administering the Java Web Console (Task Map)

Task	Description	For Instructions
Start applications from the Java Web Console's launch page.	The Java Web Console's launch page lists all the registered system management applications that you have permission to use. You connect to a specific application by clicking its application name.	"How to Start Applications From the Java Web Console's Launch Page" on page 58
Change the Java Web Console's properties.	You should not have to change any of the web console's default properties. Properties that you might choose to change include the following: ■ Console session timeout ■ Logging level ■ Audit implementation	"How to Change the Java Web Console Properties" on page 61
Install the Java Web Console software.	You install the Java Web Console software by running the `setup` script. The web console packages are installed into a directory layout that is based on the J2EE web application structure.	"How to Install the Java Web Console Software" on page 63
Remove the Java Web Console software.	You can easily remove the web console software if you need to reinstall it.	"How to Remove the Java Web Console Software" on page 64

Getting Started With the Java Web Console

On the Java Web Console's launch page, a list of the registered system management applications that you have permission to use is displayed, as well as a brief description of each application. You connect to a specific application by clicking its

application name, which is a link to the actual application. By default, the selected application opens in the web console window. You can choose to open applications in separate browsers by clicking in the appropriate check box. However, the web console launch page remains available, so you can return to it and launch multiple applications under a single login.

The web console's user interface is a web page that contains links to all available Sun system management applications. To access this web page, type the URL in the web location field. You must specify the following information:

- SSL connection by using `https`
- Name and domain of the server that is hosting the console
- Port number 6789

Note – The first time you access the Java Web Console from a particular system, you must accept the server's certificate before the web console's launch page is displayed.

If RBAC is enabled on the system, you are prompted for a role password after you have successfully logged in. Following a successful login, the web console launch page is displayed.

▼ How to Start Applications From the Java Web Console's Launch Page

Steps

1. **Start a web browser that is compatible with the Java Web Console.**

   Compatible web browsers include the following:

   - Mozilla, Version 1.2 or later
   - Netscape, 6.2.x, and 7.x

2. **Type a URL in the web browser's location field.**

   For example, if the management server host is named `sailfish`, the URL is `https://sailfish:6789`. This URL takes you to the web console login page, where you can be authenticated and authorized.

3. **Accept the server's certificate before the web console's login page displays.**

   You only have to accept the server's certificate once, not each time you start an application.

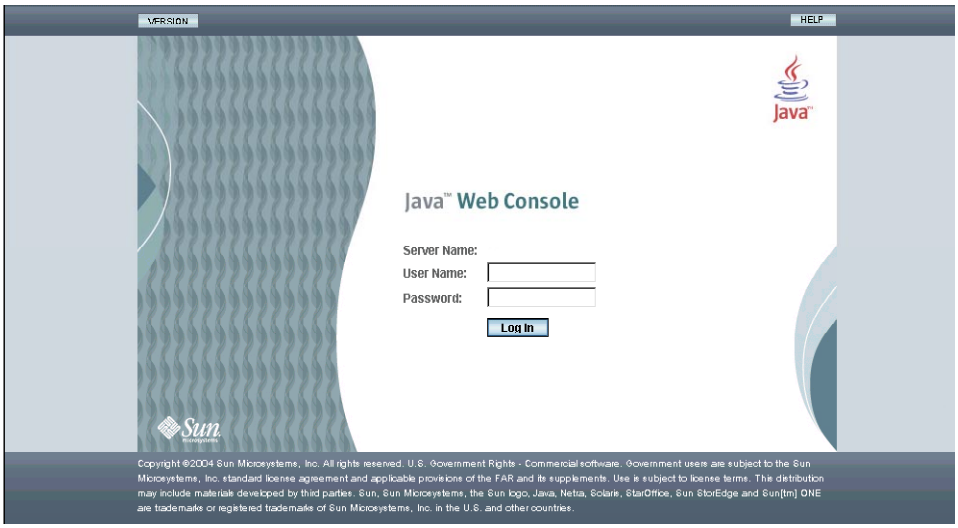

FIGURE 3–1 Java Web Console Login Page

After you are successfully authenticated, the launch page is displayed.

4. **Click the link for the application that you want to run. Click the check box if you want to run the application in a new window. Otherwise, the application will run in the default window, replacing the launch page.**

Tip – You can also launch an individual application directly and bypass the launch page by using the following syntax:

`https://`*hostname*`:6789/`*app-name*

where *app-name* is the application name that is used when the application is deployed. You can find the application name by reading the product documentation or by running the application from the web console's launch page. Then, note the URL that is displayed in the address location field.

Configuring the Java Web Console

The Java Web Console comes preconfigured to run without administrator intervention. However, you might choose to change some of the web console's default behavior. You can do so by reconfiguring properties that are in the web console's database. This task is similar in concept to editing a configuration file. However, in this case, you must use the `smreg` command to change these properties.

Properties control the behavior of the console. For example, if you want a longer timeout period, you would change the `session.timeout.value` property. The default values of most properties should not be modified unless there is a specific need that the default values do not provide, such as specifying your own login service. In general, the only property values you should change are the following:

- **Console session timeout**

  The web console's session timeout period is controlled by the `session.timeout.value` property. This property controls how long a web page can display with no activity before the session times out. After the timeout is reached, the user must log in again. The default value is 15 minutes. You can set a new value, in minutes, to conform to your own security policy. However, keep in mind that this property controls the timeout period for all registered applications.

- **Logging level**

  Administrators use logging properties to configure the logging service. `logging.default.level` is the configuration property that controls which messages are logged. The console log provides valuable information for troubleshooting problems. The following property values are available:

  - `all`
  - `info`
  - `off`
  - `severe`
  - `warning`

- **Auditing implementation**

  The web console supports three auditing implementations, `Solaris`, `Log`, and `None`. You can select an implementation by specifying the value of the `audit.default.type` configuration property. Only one auditing implementation is in effect at a time. The auditing implementation is used by all services and applications that generate audit events. The following four audit events are defined by the web console:

  - Login
  - Logout
  - Role assumption
  - Authorization

  The auditing implementations include the following:

  - `Solaris`

    This implementation is the default in this Solaris OS release. This implementation supports the BSM auditing mechanism. The auditing mechanism writes audit records into a system file in the `/var/audit` directory.

    You can display the records with the `praudit` command. For events to be captured, you must enable the BSM auditing mechanism on the system. In addition, the `/etc/security/audit_control` file must contain entries that indicate which events should be generated. You must set the `lo` event as the flag option to see login and logout events for each user. For more information,

see the `praudit`(1M) and `bsmconv`(1M) man pages and Part VII, "Solaris Auditing," in *System Administration Guide: Security Services*.

- `Log`

  You can configure this implementation to write to the system's `syslog` service. Audit messages are written to the console log if the logging service has been enabled at the `info` level. See Example 3–4 for more information.

- `None`

  No audit events are generated. Audit messages are written to the `Debug trace` log, if enabled.

Using the Console `Debug trace` Log

Properties in the `/etc/default/webconsole` file control console debug logging. Use the `debug.trace.level` property to turn on debug logging by setting the property to a value other than 0. Available choices include the following:

- **1** - Use this setting to record potentially severe errors.
- **2** - Use this setting to record important messages, as well as error messages of the 1 level.
- **3** - Use this setting to record all possible messages with full details.

By default, the `Debug trace` log is created in the `/var/log/webconsole` directory and is named `console_debug_log`. Historical logs, such as `console_debug_log.1` and `console_debug_log.2` might also exist in this directory. There can be up to 5 (default setting) historical logs stored in this directory before the earliest log is deleted and a new log is created.

EXAMPLE 3–1 Setting the Console `Debug trace` Log Level

Use the following command to set the `Debug trace` log level.

```
# smreg add -p -c debug.trace.level=level-number
```

EXAMPLE 3–2 Checking the Status of the `debug.trace.level` Property

To check the status of the `debug.trace.level` property, use the `smreg list` command.

```
# smreg list -p | grep "debug.trace.level"
```

▼ How to Change the Java Web Console Properties

Steps **1. Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Change the selected property value by using the `smreg` command.**

   # **/usr/sbin/smreg add -p -c** *name=value*

`-p`	Specifies that the object type is properties.
`-c`	Specifies that the property arguments are server configuration properties.
name=value	Specifies the property name and the new value for that property.

Example 3–3 Changing the Java Web Console's Session `Timeout` Property

This example shows how to set the session timeout value to 5 minutes.

```
# /usr/sbin/smreg add -p -c session.timeout.value=5
```

Example 3–4 Configuring the Java Web Console Logging Service

This example shows you how to set the logging level to the default, `off`.

```
# /usr/sbin/smreg add -p -c logging.default.level=off
```

Example 3–5 Choosing an Auditing Implementation for the Java Web Console

This example shows you how to set the auditing implementation to `None`.

```
# /usr/sbin/smreg add -p -c logging.default.level=None
```

Installing the Java Web Console Software

The Java Web Console is automatically installed as part of the Solaris 10 software installation. The following information is provided if you need to manually install or uninstall the web console.

▼ How to Install the Java Web Console Software

You install the Java Web Console software by running the `setup` script. The web console packages are installed into a directory layout that is based on the J2EE web application structure. For the Solaris software, the default installation is located at `/usr/share/webconsole`, which contains files for the console framework and services. The `console` subdirectory contains files that are relevant to the web console application, which is the user-visible part of the product. The `setup` command is located in the directory where the software was extracted.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Install the Java Web Console software.**

   # **/***default-installation-location***/setup [-h] [-n] [-u] [-f]**

   `-h` Specifies to print a usage statement.

   `-n` Specifies to not start the server at the end of installation.

   `-u` Specifies to uninstall the Java Web Console software.

   `-f` Specifies to uninstall the Tomcat and Java applications forcibly, if the applications were installed by using the `setup` command. Note that this option only applies when used with the `-u` option to uninstall the Java Web Console software.

Example 3–6 Installing the Java Web Console Software

This example shows you how to install the console software into the `/usr/share` directory.

```
# /usr/share/setup

Copyright 2002 Sun Microsystems, Inc. All rights reserved.

Installation of <SUNWjato> was successful.
Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.

.
.
.
    Registering com.sun.web.console_2.1.1.


    Registering com.sun.web.ui_2.1.1.
```

```
    Registering /usr/share/webconsole/lib/serviceapi.jar
    as com_sun_management_services_api.jar for scope ALL
.
.
.
Installation of <SUNWmdoc> was successful.

Installing man pages ...

Installation complete.

Starting Sun(TM) Web Console Version 2.1.1...
See /var/log/webconsole/console_debug_log for server logging information
#
```

▼ How to Remove the Java Web Console Software

Before You Begin

You must not be in any of the following directories or their subdirectories when you uninstall the Java Web Console software:

- `/usr/lib/webconsole`
- `/usr/share/webconsole`
- `/var/opt/webconsole`
- `/var/log/webconsole`

If you do not take this precaution, the software will not be completely removed.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Remove the Java Web Console software.**

   # **/***default-installation-location***/setup -u**

Example 3–7 Removing the Java Web Console Software

This example shows how to remove the Java Web Console software.

```
# /usr/lib/webconsole/setup -u

Shutting down Sun(TM) Web Console Version 2.1.1...
See /var/log/webconsole/console_debug_log for server logging information
Removing SUNWmdoc ...

Removal of <SUNWmdoc> was successful.
Removing SUNWmdemo ...
```

```
    Unregistering com.sun.web.admin.example_2.1.1.
.
.
.
     Removal of <SUNWjato> was successful.

Uninstallation complete.
#
```

Troubleshooting the Java Web Console Software

The following information is provided to help you troubleshoot any problems that you might encounter when installing or using the Java Web Console software.

▼ How to Register an Application With the Java Web Console

Applications typically are registered as part of the installation process, so you probably will not need to register an application yourself. The following instructions are provided if you need to register an application.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop the web server.**

   ```
   # smcwebserver stop
   ```

3. **Register an application.**

   ```
   # /usr/sbin/smreg add -a /opt/directory-name/app-name
   ```

   The `smreg` command manages the information in the Java Web Console's registration table. This script also performs some additional work to deploy the application.

4. **Restart the web server.**

   ```
   # /usr/sbin/smcwebserver restart
   ```

Example 3–8 Registering an Application

This example shows how to register an application that has been installed and unpacked in the `/usr/share/webconsole/example` directory.

```
# /usr/sbin/smreg add -a /usr/share/webconsole/example

Registering com.sun.web.admin.example_2.1.1.

# /usr/sbin/smcwebserver restart
```

▼ How to Unregister an Application From the Java Web Console

If you do not want a particular application to display in the web console's launch page, but you do not want to uninstall the software, you can use the `smreg` command to unregister the application.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Unregister an application.**

   ```
   # /usr/sbin/smreg remove -a app-name
   ```

Example 3–9 Unregistering an Application From the Java Web Console

This example shows how to unregister an application with the *app-name* `com.sun.web.admin.example_2.1.1`.

```
# /usr/sbin/smreg remove -a com.sun.web.admin.example_2.1.1

 Unregistering com.sun.web.admin.example_2.1.1.
```

Java Web Console Reference Information

This reference section includes the following topics:

Java Web Console Security Considerations

There are several security considerations to keep in mind when you use applications that are in the Java Web Console. These security considerations include the following:

- **Application and console access** – Whether you can see a particular application in the Java Web Console's launch page
- **Application permissions** – The levels of permissions that you must have to run parts or all of an application
- **Application access to remote systems** – How security credentials relate to remote systems

Access to Applications That Are Registered on a System

After you successfully log in to a system, you might not automatically have access to all of the applications that are registered on that system. Typically, applications are installed so that all users can see them. As an administrator, you can grant and restrict access to applications. To restrict access, specify the rights in the `authTypes` tag, which is in the application's `app.xml` file. You can find the applications `app.xml` file in the `installation-location/WEB-INF/` subdirectory. To control access to the web console itself, use the `authTypes` tag in the web console's `app.xml` file. Note that permissions to the web console are usually open so that any valid user can log in. For more information, see "Specifying Authorizations With the `authTypes` Tag" on page 68.

Application Privileges

If an application is displayed on the Java Web Console's launch page, you can run that application. However, an application might make additional authorization checks based upon the authenticated user or role identity. These checks are not controlled by the `authTypes` tag, but are explicitly coded into the application itself. For example, an application might grant read access to all authenticated users, but restrict update access to a few users or a few roles.

Application Access to Remote Systems

Having all the appropriate credentials does not guarantee that you can use an application to manage every system within the application's scope of operation. Each system that you administer by using the Java Web Console application has its own security domain. Having read-and-write permissions on the web console system does not guarantee that those credentials are automatically sufficient to administer any other remote system.

In general, access to remote systems depends on how the security is implemented in the web application. Typically, web applications make calls to *agents* that perform actions on behalf of the applications. These applications must be authenticated by the agents based on their web console credentials and the credentials by which they are known on the agent system. Depending upon how this agent authentication is done, an authorization check might also be made on the agent itself, based upon this authenticated identity.

For example, in web applications that use remote WBEM agents, authentication typically uses the user or role identity that initially authenticated to the Java Web Console. If this authentication fails on that agent system, access to that system will be denied in the web application. If authentication succeeds on that agent system, access might still be denied if the agent makes an access control check and denies access there. Most applications are written so that the authentication and authorization checks on the agent never fail if you have been successfully authenticated on the web console and assumed the correct role.

Specifying Authorizations With the `authTypes` Tag

While most system management web applications do not require any administrator intervention to use the `authTypes` tag, the system administrator might need to change the values of this tag. This tag contains a set of information that describes the level of authorization that is required for a user to view an application in the Java Web Console. The web console determines if a user is authorized to see a particular application, based on that application's specified authorization requirements. Each application can determine whether a user must have proper authorization to run the application. This determination might be made as part of the application installation process. Or, you might need to supply the information, depending on your own security requirements. The product documentation for the application should contain the information that is necessary to determine whether you need to specify a particular permission.

You can nest several other `authTypes` tags within the `authTypes` tag. The `authTypes` tag must contain at least one `authTypes` tag that provides the following necessary information:

- Type of authorization check to perform
- `Permission` subclass name
- Parameters that are required to instantiate the `Permission` subclass

In the following example, the `authTypes` tag has one attribute, `name`. The required `name` attribute is the name of the authorization service type of implementation. Different authorization types might require different values for the `classType` and `permissionParam` tags.

```
<authTypes>
    <authType name="SolarisRbac">
        <classType>com.sun.management.solaris.RbacPermission</classType>
        <permissionParam name="permission">solaris.admin.serialmgr.read</permissionParam>
    </authType>
</authTypes>
```

The following table shows the tags that can be nested within an `authTypes` tag

TABLE 3–1 Nested Tags for the `authTypes` Tag

Tag	Attribute	Description
`classType`		The `Permission` subclass name. This tag is a required tag.
`permissionParam`	`name`	The parameters that would be required to create an instance of the class specified by `classType`.

The `authTypes` tag and nested `authTypes` tags are required elements in the `app.xml` file. If you want to register an application that is available to anyone, specify the `authTypes` tag with no content, as shown in the following example.

```
<authTypes>
        <authType name="">
            <classType></classType>
            <permissionParam name=""></permissionParam>
        </authType>
</authTypes>
```

CHAPTER 4

Managing User Accounts and Groups (Overview)

This chapter provides guidelines and planning information for managing user accounts and groups. This chapter also includes information about customizing the user's work environment.

This is a list of the overview information in this chapter.

For step-by-step instructions on managing user accounts and groups, see Chapter 5.

What's New or Changed in Managing Users and Groups?

The Admintool software is no longer available in this release. The following table describes available tools for user account and group management.

TABLE 4–1 Tools for User Account and Group Management

Tool Name	Description	For More Information
Solaris Management Console	Graphical tool that is used to manage users, groups, roles, rights, mailing lists, disks, terminals, and modems.	"Setting Up User Accounts (Task Map)" on page 101

TABLE 4–1 Tools for User Account and Group Management *(Continued)*

Tool Name	Description	For More Information
`smuser`, `smrole`, `smgroup`	Commands that are used to manage users, groups and roles. The SMC services must be running to use these commands.	"Example—Adding a Group and User With the `smgroup` and `smuser` Commands" on page 107
`useradd`, `groupadd`, `roleadd`; `usermod`, `groupmod`, `rolemod`; `userdel`, `groupdel`, `roledel`	Commands that are used to manage users, groups, and roles.	"Example—Adding a Group and User With the `groupadd` and `useradd` Commands" on page 107

What Are User Accounts and Groups?

One basic system administration task is to set up a user account for each user at a site. A typical user account includes the information a user needs to log in and use a system, without having the system's root password. The components of user account information are described in "User Account Components" on page 72.

When you set up a user account, you can add the user to predefined groups of users. A typical use of groups is to set up group permissions on a file and directory, which allows access only to users who are part of that group.

For example, you might have a directory containing confidential files that only a few users should be able to access. You could set up a group called `topsecret` that includes the users working on the `topsecret` project. And, you could set up the `topsecret` files with read permission for the `topsecret` group. That way, only the users in the `topsecret` group would be able to read the files.

A special type of user account, called a *role*, is used to give selected users special privileges. For more information, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

User Account Components

The following sections describe the specific components of a user account.

User (Login) Names

User names, also called *login names*, let users access their own systems and remote systems that have the appropriate access privileges. You must choose a user name for each user account that you create.

Consider establishing a standard way of assigning user names so that they are easier for you to track. Also, names should be easy for users to remember. A simple scheme when selecting a user name is to use the first name initial and first seven letters of the user's last name. For example, Ziggy Ignatz becomes `zignatz`. If this scheme results in duplicate names, you can use the first initial, middle initial, and the first six characters of the user's last name. For example, Ziggy Top Ignatz becomes `ztignatz`.

If this scheme still results in duplicate names, consider using the following scheme to create a user name:

- The first initial, middle initial, first five characters of the user's last name
- The number 1, or 2, or 3, and so on, until you have a unique name.

Note – Each new user name must be distinct from any mail aliases that are known to the system or to an NIS or NIS+ domain. Otherwise, mail might be delivered to the alias rather than to the actual user.

For detailed guidelines on setting up user (login) names, see "Guidelines for Using User Names, User IDs, and Group IDs" on page 79.

User ID Numbers

Associated with each user name is a user identification number (UID). The UID number identifies the user name to any system on which the user attempts to log in. And, the UID number is used by systems to identify the owners of files and directories. If you create user accounts for a single individual on a number of different systems, always use the same user name and ID number. In that way, the user can easily move files between systems without ownership problems.

UID numbers must be a whole number that is less than or equal to 2147483647. UID numbers are required for both regular user accounts and special system accounts. The following table lists the UID numbers that are reserved for user accounts and system accounts.

TABLE 4–2 Reserved UID Numbers

UID Numbers	User or Login Accounts	Description
0 – 99	root, `daemon`, `bin`, `sys`, and so on	System accounts
100 – 2147483647	Regular users	General purpose accounts
60001 and 65534	`nobody` and `nobody4`	Anonymous users
60002	`noaccess`	Non trusted users

Do not assign UIDs 0 through 99, which are reserved for system use, to regular user accounts. By definition, `root` always has UID 0, `daemon` has UID 1, and pseudo-user `bin` has UID 2. In addition, you should give `uucp` logins and pseudo user logins, such as `who`, `tty`, and `ttytype`, low UIDs so that they fall at the beginning of the `passwd` file.

For additional guidelines on setting up UIDs, see "Guidelines for Using User Names, User IDs, and Group IDs" on page 79.

As with user (login) names, you should adopt a scheme to assign unique UID numbers. Some companies assign unique employee numbers. Then, administrators add a number to the employee number to create a unique UID number for each employee.

To minimize security risks, you should avoid reusing the UIDs from deleted accounts. If you must reuse a UID, "wipe the slate clean" so that the new user is not affected by attributes set for a former user. For example, a former user might have been denied access to a printer by being included in a printer deny list. However, that attribute might be inappropriate for the new user.

Using Large User IDs and Group IDs

UIDs and group IDs (GIDs) can be assigned up to the maximum value of a signed integer, or 2147483647.

However, UIDs and GIDs over 60000 do not have full functionality and are incompatible with many Solaris features. So, avoid using UIDs or GIDs over 60000.

The following table describes interoperability issues with Solaris products and previous Solaris releases.

TABLE 4–3 Interoperability Issues for UIDs or GIDs Over 60000

Category	Product or Command	Issue
NFS interoperability	SunOS 4.0 NFS software and compatible releases	NFS server and client code truncates large UIDs and GIDs to 16 bits. This situation can create security problems if systems running SunOS 4.0 and compatible releases are used in an environment where large UIDs and GIDs are being used. Systems running SunOS 4.0 and compatible releases require a patch to avoid this problem.
Name service interoperability	NIS name service and file-based name service	Users with UIDs greater than 60000 can log in or use the `su` command on systems running the Solaris 2.5 (and compatible releases). However, their UIDs and GIDs will be set to 60001 (`nobody`).

TABLE 4–3 Interoperability Issues for UIDs or GIDs Over 60000 *(Continued)*

Category	Product or Command	Issue
	NIS+ name service	Users with UIDs greater than 60000 are denied access on systems running Solaris 2.5 (and compatible releases) and the NIS+ name service.

TABLE 4–4 Large UID or GID Limitation Summary

UID or GID	Limitations
60003 or greater	■ Users who log in to systems running Solaris 2.5 (and compatible releases) and the NIS or files name service get a UID and GID of `nobody`.
65535 or greater	■ Systems running Solaris 2.5 (and compatible releases) with the NFS version 2 software truncate UIDs to 16 bits, creating possible security problems. ■ Users who use the `cpio` command with the default archive format to copy a file see an error message for each file. And, the UIDs and GIDs are set to `nobody` in the archive. ■ x86 based systems: Users that run SVR3-compatible applications will probably see `EOVERFLOW` return codes from system calls. ■ x86 based systems: If users attempt to create a file or directory on a mounted System V file system, the System V file system returns an `EOVERFLOW` error.
100000 or greater	■ The `ps -l` command displays a maximum five-digit UID. So, the printed column won't be aligned when it includes a UID or GID larger than 99999.
262144 or greater	■ Users who use the `cpio` command with the `-H odc` format or the `pax -x cpio` command to copy files see an error message returned for each file. And, the UIDs and GIDs are set to `nobody` in the archive.
1000000 or greater	■ Users who use the `ar` command have their UIDs and GIDs set to `nobody` in the archive.
2097152 or greater	■ Users who use the `tar` command, the `cpio -H ustar` command, or the `pax -x tar` command have their UIDs and GIDs set to `nobody`.

Groups

A *group* is a collection of users who can share files and other system resources. For example, users who working on the same project could be formed into a group. A group is traditionally known as a UNIX group.

Each group must have a name, a group identification (GID) number, and a list of user names that belong to the group. A GID number identifies the group internally to the system. The two types of groups that a user can belong to are as follows:

- **Primary group –** Specifies a group that the operating system assigns to files that are created by the user. Each user must belong to a primary group.
- **Secondary groups –** Specifies one or more groups to which a user also belongs. Users can belong to up to 15 secondary groups.

For detailed guidelines on setting up group names, see "Guidelines for Using User Names, User IDs, and Group IDs" on page 79.

Sometimes, a user's secondary group is not important. For example, ownership of files reflect the primary group, not any secondary groups. Other applications, however, might rely on a user's secondary group memberships. For example, a user has to be a member of the `sysadmin` group (group 14) to use the Admintool software in previous Solaris releases. However, it doesn't matter if group 14 is his or her current primary group.

The `groups` command lists the groups that a user belongs to. A user can have only one primary group at a time. However, a user can temporarily change the user's primary group, with the `newgrp` command, to any other group in which the user is a member.

When adding a user account, you must assign a primary group for a user or accept the default group, `staff` (group 10). The primary group should already exist. If the primary group does not exist, specify the group by a GID number. User names are not added to primary groups. If user names were added to primary groups, the list might become too long. Before you can assign users to a new secondary group, you must create the group and assign it a GID number.

Groups can be local to a system or managed through a name service. To simplify group administration, you should use a name service such as NIS or a directory service such as LDAP. These services enable you to centrally manage group memberships.

Passwords

You can specify a password for a user when you add the user. Or, you can force the user to specify a password when the user first logs in. User passwords must comply with the following syntax:

- Password length must at least match the value identified by the `PASSLENGTH` variable in the `/etc/default/passwd` file. By default, `PASSLENGTH` is set to 6.
- The first 6 characters of the password must contain at least two alphabetic characters and have at least one numeric or special character.
- You can increase the maximum password length to more than eight characters by configuring the `/etc/policy.conf` file with an algorithm that supports greater than eight characters.

Although user names are publicly known, passwords must be kept secret and known only to users. Each user account should be assigned a password. The password can be a combination of six to eight letters, numbers, or special characters.

To make your computer systems more secure, users should change their passwords periodically. For a high level of security, you should require users to change their passwords every six weeks. Once every three months is adequate for lower levels of security. System administration logins (such as root and sys) should be changed monthly, or whenever a person who knows the root password leaves the company or is reassigned.

Many breaches of computer security involve guessing a legitimate user's password. You should make sure that users avoid using proper nouns, names, login names, and other passwords that a person might guess just by knowing something about the user.

Good choices for passwords include the following:

- Phrases (`beammeup`)
- Nonsense words made up of the first letters of every word in a phrase. For example, `swotrb` for SomeWhere Over The RainBow.
- Words with numbers or symbols substituted for letters. For example, `sn00py` for snoopy.

Do not use these choices for passwords:

- Your name (spelled forwards, backwards, or jumbled)
- Names of family members or pets
- Car license numbers
- Telephone numbers
- Social Security numbers
- Employee numbers
- Words related to a hobby or interest
- Seasonal themes, such as Santa in December
- Any word in the dictionary

Home Directories

The home directory is the portion of a file system allocated to a user for storing private files. The amount of space you allocate for a home directory depends on the kinds of files the user creates, their size, and the number of files that are created.

A home directory can be located either on the user's local system or on a remote file server. In either case, by convention the home directory should be created as `/export/home/`*username*. For a large site, you should store home directories on a server. Use a separate file system for each `/export/home`*n* directory to facilitate backing up and restoring home directories. For example, `/export/home1`, `/export/home2`.

Regardless of where their home directory is located, users usually access their home directories through a mount point named `/home/`*username*. When AutoFS is used to mount home directories, you are not permitted to create any directories under the `/home` mount point on any system. The system recognizes the special status of `/home` when AutoFS is active. For more information about automounting home directories, see "Task Overview for Autofs Administration" in *System Administration Guide: Network Services*.

To use the home directory anywhere on the network, you should always refer to the home directory as `$HOME`, not as `/export/home/`*username*. The latter is machine-specific. In addition, any symbolic links created in a user's home directory should use relative paths (for example, `../../../x/y/x`) so that the links are valid no matter where the home directory is mounted.

Name Services

If you are managing user accounts for a large site, you might want to consider using a name or directory service such as LDAP, NIS, or NIS+. A name or directory service enables you to store user account information in a centralized manner instead of storing user account information in every system's `/etc` files. When you use a name or directory service for user accounts, users can move from system to system using the same user account without having site-wide user account information duplicated on every system. Using a name or directory service also promotes centralized and consistent user account information.

User's Work Environment

Besides having a home directory to create and store files, users need an environment that gives them access to the tools and resources they need to do their work. When a user logs in to a system, the user's work environment is determined by initialization files. These files are defined by the user's startup shell, such as the C, Korn, or Bourne shell.

A good strategy for managing the user's work environment is to provide customized user initialization files, such as `.login`, `.cshrc`, `.profile`, in the user's home directory.

Note – Do not use system initialization files, such as `/etc/profile` or `/etc/.login`, to manage a user's work environment. These files reside locally on systems and are not centrally administered. For example, if AutoFS is used to mount the user's home directory from any system on the network, you would have to modify the system initialization files on each system to ensure a consistent environment whenever a user moved from system to system.

For detailed information about customizing user initialization files for users, see "Customizing a User's Work Environment" on page 90.

Another way to customize user accounts is through role-based access control (RBAC). See "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services* for more information.

Guidelines for Using User Names, User IDs, and Group IDs

User names, UIDs, and GIDs should be unique within your organization, which might span multiple domains.

Keep the following guidelines in mind when creating user or role names, UIDs, and GIDs:

- **User names –** They should contain from two to eight letters and numerals. The first character should be a letter. At least one character should be a lowercase letter.

  Note – Even though user names can include a period (.), underscore (_), or hyphen (-), using these characters is not recommended because they can cause problems with some software products.

- **System accounts –** Do not use any of the user names, UIDs, or GIDs that are contained in the default `/etc/passwd` and `/etc/group` files. UIDs and GIDs 0-99 are reserved for system use and should not be used by anyone. This restriction includes numbers not currently in use.

  For example, `gdm` is the reserved user name and group name for the GNOME Display Manager daemon and should not be used for another user. For a complete listing of the default `/etc/passwd` and `/etc/group` entries, see Table 4–6 and Table 4–9.

  The `nobody` and `nobody4` accounts should never be used for running processes. These two accounts are reserved for use by NFS. Use of these accounts for running processes could lead to unexpected security risks. Processes that need to run as a non-root user should use the `daemon` or `noaccess` accounts.
- **System account configuration** – The configuration of the default system accounts should never be changed. This includes changing the login shell of a system account that is currently locked. The only exception to this rule is the setting of a password and password aging parameters for the root account.

Where User Account and Group Information Is Stored

Depending on your site policy, user account and group information can be stored in your local system's `/etc` files or in a name or directory service as follows:

- The NIS+ name service information is stored in tables.
- The NIS name service information is stored in maps.
- The LDAP directory service information is stored in indexed database files.

Note – To avoid confusion, the location of the user account and group information is generically referred to as a *file* rather than as a *database*, *table*, or *map*.

Most user account information is stored in the `passwd` file. Password information is stored as follows:

- In the `passwd` file when you are using NIS or NIS+
- In the `/etc/shadow` file when you are using `/etc` files
- In the `people` container when you are using LDAP

Password aging is available when you are using NIS+ or LDAP, but not NIS.

Group information is stored in the `group` file for NIS, NIS+ and files. For LDAP, group information is stored in the `group` container.

Fields in the `passwd` File

The fields in the `passwd` file are separated by colons and contain the following information:

username:*password*:*uid*:*gid*:*comment*:*home-directory*:*login-shell*

For example:

```
kryten:x:101:100:Kryten Series 4000 Mechanoid:/export/home/kryten:/bin/csh
```

The following table describes the `passwd` file fields.

TABLE 4–5 Fields in the `passwd` File

Field Name	Description
username	Contains the user or login name. User names should be unique and consist of 1-8 letters (A-Z, a-z) and numerals (0-9). The first character must be a letter, and at least one character must be a lowercase letter.
password	Contains an `x`, a placeholder for the encrypted password. The encrypted password is stored in the `shadow` file.
uid	Contains a user identification (UID) number that identifies the user to the system. UID numbers for regular users should range from 100 to 60000. All UID numbers should be unique.
gid	Contains a group identification (GID) number that identifies the user's primary group. Each GID number must be a whole number between 0 and 60002. The numbers 60001 and 60002 are assigned to `nobody` and `noaccess`. The number 65534 is assigned to `nobody4`.
comment	Usually contains the full name of the user. This field is informational only. It is sometimes called the GECOS field because it was originally used to hold the login information needed to submit batch jobs to a mainframe running GECOS (General Electric Computer Operating System) from UNIX systems at Bell Labs.
home-directory	Contains the user's home directory path name.
login-shell	Contains the user's default login shell, such as `/bin/sh`, `/bin/csh`, or `/bin/ksh`. Table 4–20 describes basic shell features.

Default `passwd` File

The default Solaris `passwd` file contains entries for standard daemons. Daemons are processes that are usually started at boot time to perform some system-wide task, such as printing, network administration, or port monitoring.

```
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/:
webservd:x:80:80:WebServer Reserved UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
```

```
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
```

TABLE 4–6 Default `passwd` File Entries

User Name	User ID	Description
`root`	`0`	Superuser account
`daemon`	`1`	Umbrella system daemon associated with routine system tasks
`bin`	`2`	Administrative daemon associated with running system binaries to perform some routine system task
`sys`	`3`	Administrative daemon associated with system logging or updating files in temporary directories
`adm`	`4`	Administrative daemon associated with system logging
`lp`	`71`	Line printer daemon
`uucp`	`5`	Daemon associated with `uucp` functions
`nuucp`	`6`	Another daemon associated with `uucp` functions
`smmsp`	`25`	Sendmail message submission program daemon
`webservd`	`80`	Account reserved for WebServer access
`gdm`	50	GNOME Display Manager daemon
`listen`	`37`	Network listener daemon
`nobody`	`60001`	Account reserved for anonymous NFS access.
`noaccess`	`60002`	Assigned to a user or a process that needs access to a system through some application but without actually logging in.
`nobody4`	`65534`	SunOS 4.0 or 4.1 version of the `nobody` user account

Fields in the `shadow` File

The fields in the `shadow` file are separated by colons and contain the following information:

username : *password* : *lastchg* : *min* : *max* : *warn* : *inactive* : *expire*

For example:

```
rimmer:86Kg/MNT/dGu.:8882:0::5:20:8978
```

The following table describes the `shadow` file fields.

TABLE 4–7 Fields in the `shadow` File

Field Name	Description
username	Contains the user name (also called the login name).
password	Might contain the one of following entries: ■ A 13-character encrypted user password ■ The string `*LK*`, which indicates an inaccessible account ■ The string `NP`, which indicates no password for the account
lastchg	Indicates the number of days between January 1, 1970, and the last password modification date.
min	Contains the minimum number of days required between password changes.
max	Contains the maximum number of days the password is valid before the user is prompted to specify a new password.
warn	Indicates the number of days before the password expires that the user is warned.
inactive	Contains the number of days a user account can be inactive before being locked.
expire	Contains the absolute date when the user account expires. Past this date, the user cannot log in to the system.

Fields in the `group` File

The fields in the `group` file are separated by colons and contain the following information:

group-name : *group-password* : *gid* : *user-list*

For example:

```
bin::2:root,bin,daemon
```

The following table describes the `group` file fields.

TABLE 4–8 Fields in the `group` File

Field Name	Description
group-name	Contains the name assigned to the group. For example, members of the chemistry department in a university might be called `chem`. Group names can have a maximum of eight characters.

TABLE 4–8 Fields in the `group` File *(Continued)*

Field Name	Description
group-password	Usually contains an asterisk or is empty. The *group-password* field is a relic of earlier versions of UNIX. If a group has a password, the `newgrp` command prompts users to enter the password. However, no utility exists to set the password.
gid	Contains the group's GID number. This number must be unique on the local system, and should be unique across the entire organization. Each GID number must be a whole number between 0 and 60002. Numbers under 100 are reserved for system default group accounts. User defined groups can range from 100 to 60000. The numbers 60001 and 60002 are reserved and assigned to `nobody` and `noaccess`, respectively.
user-list	Contains a comma-separated list of user names, representing the user's secondary group memberships. Each user can belong to a maximum of 15 secondary groups.

Default `group` file

The default Solaris `group` file contains the following system groups that support some system-wide task, such as printing, network administration, or electronic mail. Many of these groups having corresponding entries in the `passwd` file.

```
root::0:
other::1:
bin::2:root,daemon
sys::3:root,bin,adm
adm::4:root,daemon
uucp::5:root
mail::6:root
tty::7:root,adm
lp::8:root,adm
nuucp::9:root
staff::10:
daemon::12:root
smmsp::25:
sysadmin::14:
gdm::50:
webservd::80:
nobody::60001:
noaccess::60002:
nogroup::65534:
```

TABLE 4–9 Default `group` File Entries

Group Name	Group ID	Description
`root`	`0`	Superuser group

TABLE 4–9 Default `group` File Entries *(Continued)*

Group Name	Group ID	Description
`other`	`1`	Optional group
`bin`	`2`	Administrative group associated with running system binaries
`sys`	`3`	Administrative group associated with system logging or temporary directories
`adm`	`4`	Administrative group associated with system logging
`uucp`	`5`	Group associated with `uucp` functions
`mail`	`6`	Electronic mail group
`tty`	`7`	Group associated with tty devices
`lp`	`8`	Line printer group
`nuucp`	`9`	Group associated with `uucp` functions
`staff`	`10`	General administrative group.
`daemon`	`12`	Group associated with routine system tasks
`sysadmin`	`14`	Administrative group associated with legacy Admintool and Solstice AdminSuite tools
`smmsp`	25	Daemon for Sendmail message submission program
`webservd`	`80`	Group reserved for WebServer access
`gdm`	50	Group reserved for the GNOME Display Manager daemon
`nobody`	`60001`	Group assigned for anonymous NFS access
`noaccess`	`60002`	Group assigned to a user or a process that needs access to a system through some application but without actually logging in
`nogroup`	`65534`	Group assigned to a user who is not a member of a known group

Tools for Managing User Accounts and Groups

The following table lists the recommended tools for managing users and groups. These tools are included in the Solaris Management Console suite of tools. For information about starting and using the Solaris Management Console, see Chapter 2.

TABLE 4–10 Tools for Managing Users and Groups

Solaris Management Tool	Purpose
Users	Manage users accounts
User Templates	Create a set of attributes for a specific kind of user like students, engineers, or instructors
Rights	Manage RBAC rights
Administrative Roles	Manage RBAC administrative roles
Groups	Manage group information
Projects	Manage project information
Mailing Lists	Manage mailing lists

Use the Solaris Management Console online help for information on performing these tasks.

For information on the Solaris commands that can be used to manage user accounts and groups, see Table 1–6. These commands provide the same functionality as the Solaris management tools, including authentication and name service support.

Tasks for Solaris User and Group Management Tools

The Solaris user management tools enable you to manage user accounts and groups on a local system or in a name service environment.

This table describes the tasks you can do with the Users tool's User Accounts feature.

TABLE 4–11 Task Descriptions for User Accounts Tool

Task	Description
Add a user	Adds a user to the local system or name service.
Create a user template	Creates a template of predefined user attributes for creating users of the same group, such as students, contractors, or engineers.
Add a user with a user template	Adds a user with a template so that user attributes are predefined.
Clone a user template	Clones a user template if you would like to use a similar set of predefined user attributes. Then, change only some of the attributes as needed.
Set up user properties	Sets up user properties in advance of adding users. Properties include specifying whether a user template is used when adding a user, and whether the home directory or mail box is deleted by default when removing a user.
Add multiple users	Adds multiple users to the local system or name service by specifying a text file, typing each name, or automatically generating a series of user names.
View or change user properties	Displays or changes user properties such as login shell, password, or password options.
Assign rights to users	Assigns RBAC rights to users that will allow them to perform specific administration tasks.
Remove a user	Removes the user from the local system or the name service. Optionally, you can also specify whether the user's home directory or mailbox is removed. The user is also removed from any groups or roles.

For information about adding a user to the local system or name service, see "What Are User Accounts and Groups?" on page 72 and "User Account Components" on page 72.

TABLE 4–12 Task Descriptions for Rights Tool

Task	Description
Grant a right	Grants a user a right to run a specific command or application that was previously only available to an administrator.
View or change existing rights properties	Displays or changes existing rights.

TABLE 4–12 Task Descriptions for Rights Tool *(Continued)*

Task	Description
Add an authorization	Adds an authorization, which is a discrete right granted to a role or a user.
View or change an authorization	Displays or changes existing authorizations.

For more information on granting rights to users, see "Contents of Rights Profiles" in *System Administration Guide: Security Services*.

TABLE 4–13 Task Descriptions for Administrative Roles Tool

Task	Description
Add an administrative role	Adds a role that someone would use to perform a specific administrative task.
Assign rights to an administrative role	Assigns specific rights to a role that enable someone to perform a task.
Change an administrative role	Adds or removes rights from a role.

For more information on using administrative roles, see "How to Plan Your RBAC Implementation" in *System Administration Guide: Security Services*.

TABLE 4–14 Task Descriptions for Groups Tool

Task	Description
Add a group	Adds a group to the local system or name service so that the group name is available before you add the user.
Add a user to a group	Adds a user to a group if the user needs access to group-owned files.
Remove a user from a group	Removes a user from a group if the user no longer requires group file access.

For information on adding users to groups, see "Groups" on page 75.

TABLE 4–15 Task Descriptions for Mailing Lists Tool

Task	Description
Create a mailing list	Creates a mailing list, which is a list of user names for sending email messages.
Change a mailing list name	Changes the mailing list after it is created.

TABLE 4–15 Task Descriptions for Mailing Lists Tool *(Continued)*

Task	Description
Remove a mailing list	Removes a mailing list if it is no longer used.

For information on creating mailing lists, see the Solaris Management Console's online help.

TABLE 4–16 Task Descriptions for Projects Tool

Task	Description
Create or clone a project	Creates a new project or clones an existing project if the existing project has attributes similar to what you need for the new project.
Modify or view project attributes	Displays or changes existing project attributes.
Delete a project	Removes a project if the project is no longer used.

Managing Users and Resources With Projects

Starting in the Solaris 9 release, users and groups can be members of a *project*, an identifier that indicates a workload component that can be used as the basis of system usage or resource allocation chargeback. Projects are part of the Solaris resource management feature that is used to manage system resources.

Users need to be a member of a project to successfully log in to a system running the Solaris 9 release. By default, users are a member of the `group.staff` project when the Solaris 9 release is installed and no other project information is configured.

User project information is stored in the `/etc/project` file, which can be stored on the local system (files), the NIS name service, or the LDAP directory service. You can use the Solaris Management Console to manage project information.

The `/etc/project` file must exist for users to log in successfully, but requires no administration if you are not using projects.

For more information on using or setting up projects, see Chapter 2, "Projects and Tasks (Overview)," in *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*.

Customizing a User's Work Environment

Part of setting up a user's home directory is providing user initialization files for the user's login shell. A *user initialization file* is a shell script that sets up a work environment for a user after the user logs in to a system. Basically, you can perform any task in a user initialization file that you can do in a shell script. However, a user initialization file's primary job is to define the characteristics of a user's work environment, such as a user's search path, environment variables, and windowing environment. Each login shell has its own user initialization file or files, which are listed in the following table.

TABLE 4–17 User Initialization Files for Bourne, C, and Korn Shells

Shell	User Initialization File	Purpose
Bourne	`$HOME/.profile`	Defines the user's environment at login
C	`$HOME/.cshrc`	Defines the user's environment for all C shells and is invoked after login shell
	`$HOME/.login`	Defines the user's environment at login
Korn	`$HOME/.profile`	Defines the user's environment at login
	`$HOME/$ENV`	Defines user's environment at login in the file and is specified by the Korn shell's `ENV` environment variable

The Solaris environment provides default user initialization files for each shell in the `/etc/skel` directory on each system, as shown in the following table.

TABLE 4–18 Default User Initialization Files

Shell	Default File
C	`/etc/skel/local.login`
	`/etc/skel/local.cshrc`
Bourne or Korn	`/etc/skel/local.profile`

You can use these files as a starting point and modify them to create a standard set of files that provide the work environment common to all users. Or, you can modify these files to provide the working environment for different types of users. Although you cannot create customized user initialization files with the Users tool, you can populate a user's home directory with user initialization files located in a specified "skeleton" directory. You can do this by creating a user template with the User Templates tool and specifying a skeleton directory from which to copy user initialization files.

For step-by-step instructions on how to create sets of user initialization files for different types of users, see "How to Customize User Initialization Files" on page 103.

When you use the Users tool to create a new user account and select the create home directory option, the following files are created, depending on which login shell is selected:

TABLE 4–19 Files Created by Users Tool When Adding a User

Shell	Files Created
C	The `/etc/skel/local.cshrc` and the `/etc/skel/local.login` files are copied into the user's home directory and are renamed `.cshrc` and `.login`, respectively.
Bourne and Korn	The `/etc/skel/local.profile` file is copied into the user's home directory and renamed `.profile`.

If you use the `useradd` command to add a new user account and specify the `/etc/skel` directory by using the `-k` and `-m` options, all three `/etc/skel/local*` files and the `/etc/skel/.profile` file are copied into the user's home directory. At this point, you need to rename them to whatever is appropriate for the user's login shell.

Using Site Initialization Files

The user initialization files can be customized by both the administrator and the user. This important feature can be accomplished with centrally located and globally distributed user initialization files, called *site initialization files*. Site initialization files enable you to continually introduce new functionality to the user's work environment, while enabling the user to customize the user's initialization file.

When you reference a site initialization file in a user initialization file, all updates to the site initialization file are automatically reflected when the user logs in to the system or when a user starts a new shell. Site initialization files are designed for you to distribute site-wide changes to users' work environments that you did not anticipate when you added the users.

Any customization that can be done in a user initialization file can be done in a site initialization file. These files typically reside on a server, or set of servers, and appear as the first statement in a user initialization file. Also, each site initialization file must be the same type of shell script as the user initialization file that references it.

To reference a site initialization file in a C-shell user initialization file, place a line similar to the following at the beginning of the user initialization file:

`source /net/`*machine-name/export/site-files/site-init-file*

To reference a site initialization file in a Bourne-shell or Korn-shell user initialization file, place a line similar to the following at the beginning of the user initialization file:

`. /net/`*machine-name/export/site-files/site-init-file*

Avoiding Local System References

You should not add specific references to the local system in the user initialization file. You want the instructions in a user initialization file to be valid regardless of which system the user logs into. For example:

- To make a user's home directory available anywhere on the network, always refer to the home directory with the variable `$HOME`. For example, use `$HOME/bin` instead of `/export/home/`*username*`/bin`. The `$HOME` variable works when the user logs in to another system and the home directories are automounted.
- To access files on a local disk, use global path names, such as `/net/`*system-name/directory-name*. Any directory referenced by `/net/`*system-name* can be mounted automatically on any system on which the user logs in, assuming the system is running AutoFS.

Shell Features

The following table lists basic shell features that each shell provides, which can help you determine what you can and can't do when creating user initialization files for each shell.

TABLE 4–20 Basic Features of Bourne, C, and Korn Shells

Feature	Bourne	C	Korn
Known as the standard shell in UNIX	Yes	No	No
Compatible syntax with Bourne shell	-	No	Yes

TABLE 4–20 Basic Features of Bourne, C, and Korn Shells *(Continued)*

Feature	Bourne	C	Korn
Job control	Yes	Yes	Yes
History list	No	Yes	Yes
Command-line editing	No	Yes	Yes
Aliases	No	Yes	Yes
Single-character abbreviation for login directory	No	Yes	Yes
Protection from overwriting (`noclobber`)	No	Yes	Yes
Setting to ignore Control-D (`ignoreeof`)	No	Yes	Yes
Enhanced `cd` command	No	Yes	Yes
Initialization file separate from `.profile`	No	Yes	Yes
Logout file	No	Yes	No

Shell Environment

A shell maintains an environment that includes a set of variables defined by the `login` program, the system initialization file, and the user initialization files. In addition, some variables are defined by default. A shell can have two types of variables:

- **Environment variables –** Variables that are exported to all processes spawned by the shell. Their settings can be seen with the `env` command. A subset of environment variables, such as `PATH`, affects the behavior of the shell itself.
- **Shell (local) variables –** Variables that affect only the current shell. In the C shell, a set of these shell variables have a special relationship to a corresponding set of environment variables. These shell variables are `user`, `term`, `home`, and `path`. The value of the environment variable counterpart is initially used to set the shell variable.

In the C shell, you use the lowercase names with the `set` command to set shell variables. You use uppercase names with the `setenv` command to set environment variables. If you set a shell variable, the shell sets the corresponding environment variable and vice versa. For example, if you update the `path` shell variable with a new path, the shell also updates the `PATH` environment variable with the new path.

In the Bourne and Korn shells, you can use the uppercase variable name equal to some value to set both shell and environment variables. You also have to use the `export` command to activate the variables for any subsequently executed commands.

For all shells, you generally refer to shell and environment variables by their uppercase names.

In a user initialization file, you can customize a user's shell environment by changing the values of the predefined variables or by specifying additional variables. The following table shows how to set environment variables in a user initialization file.

TABLE 4–21 Setting Environment Variables in a User Initialization File

Shell Type	Line to Add to the User Initialization File
C shell	`setenv` *VARIABLE value* Example: `setenv MAIL /var/mail/ripley`
Bourne or Korn shell	*VARIABLE=value*`; export` *VARIABLE* Example: `MAIL=/var/mail/ripley;export MAIL`

The following table describes environment variables and shell variables that you might want to customize in a user initialization file. For more information about variables that are used by the different shells, see the `sh`(1), `ksh`(1), or `csh`(1) man pages.

TABLE 4–22 Shell and Environment Variable Descriptions

Variable	Description
`CDPATH`, or `cdpath` in the C shell	Sets a variable used by the `cd` command. If the target directory of the `cd` command is specified as a relative path name, the `cd` command first looks for the target directory in the current directory (“.”). If the target is not found, the path names listed in the `CDPATH` variable are searched consecutively until the target directory is found and the directory change is completed. If the target directory is not found, the current working directory is left unmodified. For example, the `CDPATH` variable is set to `/home/jean`, and two directories exist under `/home/jean`, `bin`, and `rje`. If you are in the `/home/jean/bin` directory and type `cd rje`, you change directories to `/home/jean/rje`, even though you do not specify a full path.
`history`	Sets the history for the C shell.
`HOME`, or `home` in the C shell	Sets the path to the user's home directory.
`LANG`	Sets the locale.

TABLE 4–22 Shell and Environment Variable Descriptions *(Continued)*

Variable	Description
LOGNAME	Defines the name of the user currently logged in. The default value of LOGNAME is set automatically by the login program to the user name specified in the passwd file. You should only need to refer to, not reset, this variable.
LPDEST	Sets the user's default printer.
MAIL	Sets the path to the user's mailbox.
MANPATH	Sets the hierarchies of man pages that are available.
PATH, or path in the C shell	Specifies, in order, the directories that the shell searches to find the program to run when the user types a command. If the directory is not in the search path, users must type the complete path name of a command. As part of the login process, the default PATH is automatically defined and set as specified in .profile (Bourne or Korn shell) or .cshrc (C shell). The order of the search path is important. When identical commands exist in different locations, the first command found with that name is used. For example, suppose that PATH is defined in Bourne and Korn shell syntax as PATH=/bin:/usr/bin:/usr/sbin:$HOME/bin and a file named sample resides in both /usr/bin and /home/jean/bin. If the user types the command sample without specifying its full path name, the version found in /usr/bin is used.
prompt	Defines the shell prompt for the C shell.
PS1	Defines the shell prompt for the Bourne or Korn shell.
SHELL, or shell in the C shell	Sets the default shell used by make, vi, and other tools.
TERMINFO	Specifies the path name for an unsupported terminal that has been added to the terminfo file. Use the TERMINFO variable in either the /etc/profile or /etc/.login file. When the TERMINFO environment variable is set, the system first checks the TERMINFO path defined by the user. If the system does not find a definition for a terminal in the TERMINFO directory defined by the user, it searches the default directory, /usr/share/lib/terminfo, for a definition. If the system does not find a definition in either location, the terminal is identified as "dumb."
TERM, or term in the C shell	Defines the terminal. This variable should be reset in either the /etc/profile or /etc/.login file. When the user invokes an editor, the system looks for a file with the same name that is defined in this environment variable. The system searches the directory referenced by TERMINFO to determine the terminal characteristics.

TABLE 4–22 Shell and Environment Variable Descriptions *(Continued)*

Variable	Description
`TZ`	Sets the time zone. The time zone is used to display dates, for example, in the `ls -l` command. If `TZ` is not set in the user's environment, the system setting is used. Otherwise, Greenwich Mean Time is used.

The `PATH` Variable

When the user executes a command by using the full path, the shell uses that path to find the command. However, when users specify only a command name, the shell searches the directories for the command in the order specified by the `PATH` variable. If the command is found in one of the directories, the shell executes the command.

A default path is set by the system. However, most users modify it to add other command directories. Many user problems related to setting up the environment and accessing the correct version of a command or a tool can be traced to incorrectly defined paths.

Setting Path Guidelines

Here are some guidelines for setting up efficient `PATH` variables:

- If security is not a concern, put the current working directory (.) first in the path. However, including the current working directory in the path poses a security risk that you might want to avoid, especially for superuser.
- Keep the search path as short as possible. The shell searches each directory in the path. If a command is not found, long searches can slow down system performance.
- The search path is read from left to right, so you should put directories for commonly used commands at the beginning of the path.
- Make sure that directories are not duplicated in the path.
- Avoid searching large directories, if possible. Put large directories at the end of the path.
- Put local directories before NFS mounted directories to lessen the chance of "hanging" when the NFS server does not respond. This strategy also reduces unnecessary network traffic.

Examples—Setting a User's Default Path

The following examples show how to set a user's default path to include the home directory and other NFS mounted directories. The current working directory is specified first in the path. In a C-shell user initialization file, you would add the following:

```
set path=(. /usr/bin $HOME/bin /net/glrr/files1/bin)
```

In a Bourne-shell or Korn-shell user initialization file, you would add the following:

```
PATH=.:/usr/bin:/$HOME/bin:/net/glrr/files1/bin
export PATH
```

Locale Variables

The LANG and LC environment variables specify the locale-specific conversions and conventions for the shell. These conversions and conventions include time zones, collation orders, and formats of dates, time, currency, and numbers. In addition, you can use the stty command in a user initialization file to indicate whether the terminal session will support multibyte characters.

The LANG variable sets all possible conversions and conventions for the given locale. You can set various aspects of localization separately through these LC variables: LC_COLLATE, LC_CTYPE, LC_MESSAGES, LC_NUMERIC, LC_MONETARY, and LC_TIME.

The following table describes some of the values for the LANG and LC environment variables.

TABLE 4–23 Values for LANG and LC Variables

Value	Locale
de_DE.ISO8859-1	German
en_US.UTF-8	American English (UTF-8)
es_ES.ISO8859-1	Spanish
fr_FR.ISO8859-1	French
it_IT.ISO8859-1	Italian
ja_JP.eucJP	Japanese (EUC)
ko_KR.EUC	Korean (EUC)
sv_SE.ISO8859-1	Swedish
zh_CN.EUC	Simplified Chinese (EUC)
zh_TW.EUC	Traditional Chinese (EUC)

For more information on supported locales, see the *International Language Environments Guide*.

Examples—Setting the Locale Using the LANG Variables

The following examples show how to set the locale by using the LANG environment variables. In a C-shell user initialization file, you would add the following:

```
setenv LANG de_DE.ISO8859-1
```

In a Bourne-shell or Korn-shell user initialization file, you would add the following:

```
LANG=de_DE.ISO8859-1; export LANG
```

Default File Permissions (umask)

When you create a file or directory, the default file permissions assigned to the file or directory are controlled by the *user mask*. The user mask is set by the umask command in a user initialization file. You can display the current value of the user mask by typing umask and pressing Return.

The user mask contains the following octal values:

- The first digit sets permissions for the user
- The second digit sets permissions for group
- The third digit sets permissions for other, also referred to as "world"

Note that if the first digit is zero, it is not displayed. For example, if the user mask is set to 022, 22 is displayed.

To determine the umask value you want to set, subtract the value of the permissions you want from 666 (for a file) or 777 (for a directory). The remainder is the value to use with the umask command. For example, suppose you want to change the default mode for files to 644 (rw-r--r--). The difference between 666 and 644 is 022, which is the value you would use as an argument to the umask command.

You can also determine the umask value you want to set by using the following table. This table shows the file and directory permissions that are created for each of the octal values of umask.

TABLE 4–24 Permissions for umask Values

umask Octal Value	File Permissions	Directory Permissions
0	rw-	rwx
1	rw-	rw-
2	r--	r-x
3	r--	r--

TABLE 4–24 Permissions for `umask` Values *(Continued)*

`umask` Octal Value	File Permissions	Directory Permissions
4	-w-	-wx
5	-w-	-w-
6	--x	--x
7	--- (none)	--- (none)

The following line in a user initialization file sets the default file permissions to `rw-rw-rw-`.

```
umask 000
```

Examples of User and Site Initialization Files

The following sections provide examples of user and site initialization files that you can use to start customizing your own initialization files. These examples use system names and paths that you need to change for your particular site.

Example—The `.profile` File

```
(Line 1) PATH=$PATH:$HOME/bin:/usr/local/bin:/usr/ccs/bin:.
(Line 2) MAIL=/var/mail/$LOGNAME
(Line 3) NNTPSERVER=server1
(Line 4) MANPATH=/usr/share/man:/usr/local/man
(Line 5) PRINTER=printer1
(Line 6) umask 022
(Line 7) export PATH MAIL NNTPSERVER MANPATH PRINTER
```

1. Defines the user's shell search path
2. Defines the path to the user's mail file
3. Defines the user's Usenet news server
4. Defines the user's search path for man pages
5. Defines the user's default printer
6. Sets the user's default file creation permissions
7. Sets the listed environment variables

Example—The `.cshrc` File

```
(Line 1) set path=($PATH $HOME/bin /usr/local/bin /usr/ccs/bin)
(Line 2) setenv MAIL /var/mail/$LOGNAME
(Line 3) setenv NNTPSERVER server1
(Line 4) setenv PRINTER printer1
(Line 5) alias h history
```

(Line 6) `umask 022`
(Line 7) `source /net/server2/site-init-files/site.login`

1. Defines the user's shell search path.
2. Defines the path to the user's mail file.
3. Defines the user's Usenet news server.
4. Defines the user's default printer.
5. Creates an alias for the `history` command. The user needs to type only `h` to run the `history` command.
6. Sets the user's default file creation permissions.
7. Sources the site initialization file.

Example—Site Initialization File

The following shows an example site initialization file in which a user can choose a particular version of an application.

```
# @(#)site.login
main:
echo "Application Environment Selection"
echo ""
echo "1. Application, Version 1"
echo "2. Application, Version 2"
echo ""
echo -n "Type 1 or 2 and press Return to set your
application environment: "

set choice = $<

if ( $choice !~ [1-2] ) then
goto main
endif

switch ($choice)

case "1":
setenv APPHOME /opt/app-v.1
breaksw

case "2":
setenv APPHOME /opt/app-v.2
endsw
```

This site initialization file could be referenced in a user's `.cshrc` file (C shell users only) with the following line:

```
source /net/server2/site-init-files/site.login
```

In this line, the site initialization file is named `site.login` and is located on a server named `server2`. This line also assumes that the automounter is running on the user's system.

CHAPTER 5

Managing User Accounts and Groups (Tasks)

This chapter describes how to set up and maintain user accounts and groups.

For information on the procedures associated with setting up and maintaining user accounts and groups, see the following:

- "Setting Up User Accounts (Task Map)" on page 101
- "Maintaining User Accounts (Task Map)" on page 111

For background information about managing user accounts and groups, see Chapter 4.

Setting Up User Accounts (Task Map)

Task	Description	For Instructions
Gather user information.	Use a standard form to gather user information to help you keep user information organized.	"How to Gather User Information" on page 102
Customize user initialization files.	You can set up user initialization files (`.cshrc`, `.profile`, `.login`), so that you can provide new users with consistent environments.	"How to Customize User Initialization Files" on page 103

Task	Description	For Instructions
Add a group.	You can add a group with the following tools: Solaris Management Console's Groups tool Solaris command-line interface tools	"How to Add a Group With the Solaris Management Console's Groups Tool" on page 105 "How to Add Groups and Users With Command-Line Tools" on page 107
Add a user.	You can add a user with the following tools: Solaris Management Console's Users tool Solaris command-line interface tools	"How to Add a User With the Solaris Management Console's Users Tool" on page 106 "How to Add Groups and Users With Command-Line Tools" on page 107
Set up a user template.	You can create a user template so that you don't have to manually add all similar user properties.	See Solaris Management Console online help
Add rights or a role to a user.	You can add rights or a role to a user so that the user can perform a specific command or task.	See Solaris Management Console online help
Share the user's home directory.	You must share the user's home directory so that the directory can be remotely mounted from the user's system.	"How to Share a User's Home Directory" on page 108
Mount the user's home directory.	You must mount the user's home directory on the user's system.	"How to Mount a User's Home Directory" on page 110

How to Gather User Information

You can create a form such as the following to gather information about users before adding their accounts.

Item	Description
User Name:	
Role Name:	

Item	Description
Profiles or Authorizations:	
UID:	
Primary Group:	
Secondary Groups:	
Comment:	
Default Shell:	
Password Status and Aging:	
Home Directory Path Name:	
Mounting Method:	
Permissions on Home Directory:	
Mail Server:	
Department Name:	
Department Administrator:	
Manager:	
Employee Name:	
Employee Title:	
Employee Status:	
Employee Number:	
Start Date:	
Add to These Mail Aliases:	
Desktop System Name:	

▼ How to Customize User Initialization Files

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Create a skeleton directory for each type of user.**

   # **mkdir /***shared-dir***/skel/***user-type*

shared-dir The name of a directory that is available to other systems on the network.

user-type The name of a directory to store initialization files for a type of user.

3. **Copy the default user initialization files into the directories that you created for different types of users.**

```
# cp /etc/skel/local.cshrc /shared-dir/skel/user-type/.cshrc
# cp /etc/skel/local.login /shared-dir/skel/user-type/.login
# cp /etc/skel/local.profile /shared-dir/skel/user-type/.profile
```

Note – If the account has profiles assigned to it, then the user has to launch a special version of the shell called a profile shell to use commands (with any security attributes) that are assigned to the profile. There are three *profile shells* corresponding to the types of shells: `pfsh` (Bourne shell), `pfcsh` (C shell), and `pfksh` (Korn shell). For information about profile shells, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

---

4. **Edit the user initialization files for each user type and customize them based on your site's needs.**

   For a detailed description on the ways to customize the user initialization files, see "Customizing a User's Work Environment" on page 90.

5. **Set the permissions for the user initialization files.**

```
# chmod 744 /shared-dir/skel/user-type/.*
```

6. **Verify that the permissions for the user initialization files are correct.**

```
# ls -la /shared-dir/skel/*
```

Example 5–1 Customizing User Initialization Files

The following example shows how to customize the C-shell user initialization file in the `/export/skel/enduser` directory designated for a particular type of user. For an example of a `.cshrc` file, see "Example—The `.cshrc` File" on page 99.

```
# mkdir /export/skel/enduser
# cp /etc/skel/local.cshrc /export/skel/enduser/.cshrc

(Edit .cshrc file)
# chmod 744 /export/skel/enduser/.*
```

▼ How to Add a Group With the Solaris Management Console's Groups Tool

You can add existing users to the group when you add the group. Or, you can just add the group and then add the user to the group when you add the user.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start the Solaris Management Console.**

   ```
   # /usr/sadm/bin/smc &
   ```

   For more information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44 or "How to Start the Solaris Management Console in a Name Service Environment" on page 51.

3. **Click the This Computer icon under the Management Tools icon in the Navigation pane.**

   A list of categories is displayed.

4. **(Optional) Select the appropriate toolbox for your name service environment.**

5. **Click the System Configuration icon.**

6. **Click the User Accounts icon.**

7. **Provide the superuser password or the role password.**

8. **Click the Groups icon.**

   Use the Context help to add a group to the system.

9. **Select Add Group from the Action menu.**

10. **Identify the group name at the Group Name prompt under Group Identification.**

    For example, `mechanoids`.

11. **Identify the group number at the Group ID number prompt.**

    For example, GID 101.

12. **Click OK.**

▼ How to Add a User With the Solaris Management Console's Users Tool

Use the following procedure to add a user with the Solaris Management Console's Users tool.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start the Solaris Management Console.**

   ```
   # /usr/sadm/bin/smc &
   ```

   For more information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44 or "How to Start the Solaris Management Console in a Name Service Environment" on page 51.

3. **Click the This Computer icon under the Management Tools icon in the Navigation pane.**

   A list of categories is displayed.

4. **(Optional) Select the appropriate toolbox for your name service environment.**

5. **Click the System Configuration icon.**

6. **Click the User Accounts icon.**

7. **Provide the superuser password or the role password.**

8. **Click the Users icon.**

   Use the Context help to add a user to the system.

9. **Select Add User⇒With Wizard from the Action menu.**

   Click Next between the steps below.

   a. **Identify the user name or login name at the User Name prompt.**

      For example, `kryten`

   b. **(Optional) Identify the user's full name at the Full Name prompt.**

      For example, `kryten series 3000`.

   c. **(Optional) Provide a further description of this user at the Description prompt.**

   d. **Provide the user ID at the User ID Number prompt.**

      For example, `1001`.

e. **Select the User Must Use This Password At First Login option.**

   Provide a password for the user at the Password prompt and then confirm the password at the Confirm Password prompt.

f. **Select the user's primary group.**

   For example, `mechanoids`.

g. **Create the user's home directory by accepting the defaults at the Server and Path prompts.**

h. **Specify the mail server.**

i. **Review the information you provided and go back to correct the information, if necessary. Otherwise, click Finish.**

How to Add Groups and Users With Command-Line Tools

This section provides examples of adding users and groups with command-line tools.

Example—Adding a Group and User With the `groupadd` and `useradd` Commands

The following example shows how to use the `groupadd` and `useradd` commands to add the group `scutters` and the user `scutter1` to files on the local system. These commands cannot be used to manage users in a name service environment.

```
# groupadd -g 102 scutters
# useradd -u 1003 -g 102 -d /export/home/scutter1 -s /bin/csh \
-c "Scutter 1" -m -k /etc/skel scutter1
64 blocks
```

For more information, see the `groupadd`(1M) and `useradd`(1M) man pages.

Example—Adding a Group and User With the `smgroup` and `smuser` Commands

The following example shows how to use the `smgroup` and `smuser` commands to add the group `gelfs` and the user `camille` to the NIS domain `solar.com` on the host `starlite`.

```
# /usr/sadm/bin/smgroup add -D nis:/starlitesolar.com -- -g 103 -n gelfs
# /usr/sadm/bin/smuser add -D nis:/starlite/solar.com -- -u 1004
-n camille -c "Camille G." -d /export/home/camille -s /bin/csh -g gelfs
```

For more information, see the `smgroup`(1M) and `smuser`(1M) man pages.

Setting Up Home Directories With the Solaris Management Console

Keep the following in mind when using the Solaris Management Console tools to manage user home directories:

- If you use the Users tool's Add User Wizard to add a user account and you specify the user's home directory as `/export/home/`*username*, the home directory is automatically set up to be automounted. Also, the following entry is added to the `passwd` file:

  `/home/`*username*

- There is only way you can use Users tool to set up a user account that does not automount the home directory. First, set up a user account template that disables this feature. Then, add users with this template. You cannot disable this feature with the Add User Wizard.
- You can use the `smuser add` command with the `-x autohome=N` option to add a user without automounting the user's home directory. However, there is no option to the `smuser delete` command to remove the home directory after the user is added. You would have to remove the user and the user's home directory with the Users tool.

▼ How to Share a User's Home Directory

Use the following procedure to share a user's home directory.

Steps

1. **Become superuser or assume an equivalent role on the system that contains the home directory.**
2. **Verify that the `mountd` daemon is running.**

   In this release, `mountd` is now started as part of the NFS server service. To see if the `mountd` daemon is running, type the following command:

   ```
   # svcs network/nfs/server
   STATE          STIME    FMRI
   online         Aug_26   svc:/network/nfs/server:default
   ```

3. **If the `mountd` daemon is not running, start it.**

   ```
   # svcadm network/nfs/server
   ```

4. **List the file systems that are shared on the system.**

   ```
   # share
   ```

5. **Select one of the following based on whether the file system that contains the user's home directory is already shared.**

   a. **If the user's home directory is already shared, go to the step 8.**

   b. **If the user's home directory is not shared, go to Step 6.**

6. **Edit the `/etc/dfs/dfstab` file and add the following line:**

   ```
   share -F nfs /file-system
   ```

   /file-system is the file system that contains the user's home directory that you need to share. By convention, the file system is `/export/home`.

7. **Share the file systems listed in the `/etc/dfs/dfstab` file.**

   ```
   # shareall -F nfs
   ```

   This command executes all the `share` commands in the `/etc/dfs/dfstab` file so that you do not have to wait to reboot the system.

8. **Verify that a user's home directory is shared.**

   ```
   # share
   ```

Example 5–2 Sharing a User's Home Directory

The following example shows how to share the `/export/home` directory.

```
# svcs network/nfs/server
# svcadm network/nfs/server
# share
# vi /etc/dfs/dfstab

(The line share -F nfs /export/home is added.)
# shareall -F nfs
# share
-               /usr/dist               ro   ""
-               /export/home/user-name  rw   ""
```

See Also If the user's home directory is not located on the user's system, you have to mount the user's home directory from the system where it is located. For detailed instructions, see "How to Mount a User's Home Directory" on page 110.

▼ How to Mount a User's Home Directory

For information on automounting a home directory, see "Task Overview for Autofs Administration" in *System Administration Guide: Network Services*.

Steps

1. **Make sure that the user's home directory is shared.**

   For more information, see "How to Share a User's Home Directory" on page 108.

2. **Log in as superuser on the user's system.**

3. **Edit the `/etc/vfstab` file and create an entry for the user's home directory.**

   system-name`:/export/home/`*user-name* `- /export/home/`*username* `nfs - yes rw`

system-name	The name of the system where the home directory is located.
`/export/home/`*username*	The name of the user's home directory that will be shared. By convention, `/export/home/`*username* contains user home directories. However, you can use a different file system.
`-`	Required placeholders in the entry.
`/export/home/`*username*	The name of the directory where the user's home directory will be mounted.

   For more information about adding an entry to the `/etc/vfstab` file, see "Mounting File Systems" in *System Administration Guide: Devices and File Systems*.

4. **Create the mount point for the user's home directory.**

   `# mkdir -p /export/home/`*username*

5. **Mount the user's home directory.**

   ```
   # mountall
   ```

   All entries in the current `vfstab` file (whose `mount at boot` fields are set to `yes`) are mounted.

6. **Verify that the home directory is mounted.**

   `# mount | grep` *username*

Example 5–3 Mounting a User's Home Directory

The following example shows how to mount user `ripley`'s home directory.

```
# vi /etc/vfstab

(The line  venus:/export/home/ripley - /export/home/ripley
nfs - yes rw is added.)
```

```
# mkdir -p /export/home/ripley
# mountall
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=...
/devices on /devices read/write/setuid/dev=46c0000 on Thu Jan  8 09:38:19 2004
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=...
/proc on /proc read/write/setuid/dev=4700000 on Thu Jan  8 09:38:27 2004
/etc/mnttab on mnttab read/write/setuid/dev=47c0000 on Thu Jan  8 09:38:27 2004
/dev/fd on fd read/write/setuid/dev=4800000 on Thu Jan  8 09:38:30 2004
/var/run on swap read/write/setuid/xattr/dev=1 on Thu Jan  8 09:38:30 2004
/tmp on swap read/write/setuid/xattr/dev=2 on Thu Jan  8 09:38:30 2004
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr/onerror=...
/export/home/ripley on venus:/export/home/ripley remote/read/write/setuid/xattr/dev=...
```

Maintaining User Accounts (Task Map)

Task	Description	Instructions
Modify a group.	You can modify a group's name or the users in a group by using the Groups tool.	"How to Modify a Group" on page 113
Delete a group.	You can delete a group if it is no longer needed.	"How to Delete a Group" on page 113
Modify a user account.	*Disable a user account* You can temporarily disable a user account if it will be needed in the future. *Change a user's password* You might need to change a user's password if the user forgets it. *Set password aging* You can force users to change their passwords periodically with User Account tool's Password Options menu.	"How to Disable a User Account" on page 115 "How to Change a User's Password" on page 116 "How to Set Password Aging on a User Account" on page 116
Delete a user account.	You can delete a user account if it is no longer needed.	"How to Delete a User Account" on page 117

Modifying User Accounts

Unless you define a user name or UID number that conflicts with an existing one, you should never need to modify a user account's user name or UID number. Use the following steps if two user accounts have duplicate user names or UID numbers:

- If two user accounts have duplicate UID numbers, use the Users tool to remove one account and add it again with a different UID number. You cannot use the Users tool to modify a UID number of an existing user account.
- If two user accounts have duplicate user names, use the Users tool to modify one of the accounts and change the user name.

If you do use the Users tool to change a user name, the home directory's ownership is changed, if a home directory exists for the user.

One part of a user account that you can change is a user's group memberships. Select the Properties option from Users tool's Action menu to add or delete a user's secondary groups. Alternatively, you can use the Groups tool to directly modify a group's member list.

You can also modify the following parts of a user account:

- Description (comment)
- Login shell
- Passwords and password options
- Home directory and home directory access
- Rights and roles

Disabling User Accounts

Occasionally, you might need to temporarily or permanently disable a user account. Disabling or locking a user account means that an invalid password, `*LK*`, is assigned to the user account, preventing future logins.

The easiest way to disable a user account is to lock the password for an account with Users tool.

You can also enter an expiration date in the account availability section of the User Properties screen. An expiration date enables you to set a limit on how long the account is active.

Other ways to disable a user account: set up password aging or change the user's password.

Deleting User Accounts

When you delete a user account with the Users tool, the software deletes the entries in the `passwd` and `group` files. In addition, the files in the user's home directory and mail directory are deleted also.

▼ How to Modify a Group

Use the following procedure to modify a group.

Steps

1. **Become superuser or assume an equivalent role.**
   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.
2. **Start the Solaris Management Console.**
   ```
   # /usr/sadm/bin/smc &
   ```
   For more information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44 or "How to Start the Solaris Management Console in a Name Service Environment" on page 51.
3. **Click the This Computer icon under the Management Tools icon in the Navigation pane.**
   A list of categories is displayed.
4. **(Optional) Select the appropriate toolbox for your name service environment.**
5. **Click the System Configuration icon.**
6. **Click the User Accounts icon.**
7. **Provide the superuser password or the role password.**
8. **Click the Groups icon.**
9. **Select the group to modify.**
   For example, select `scutters`.
10. **Modify the selected group in the Group Name: text box.**
    For example, change `scutters` to `scutter`.
    All the users that were in the `scutters` group are now in the `scutter` group.
11. **Click OK.**

▼ How to Delete a Group

Use the following procedure to delete a group.

Steps

1. **Become superuser or assume an equivalent role.**
   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide:*

Security Services.

2. **Start the Solaris Management Console.**

   ```
   # /usr/sadm/bin/smc &
   ```

   For more information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44 or "How to Start the Solaris Management Console in a Name Service Environment" on page 51.

3. **Click the This Computer icon under the Management Tools icon in the Navigation pane.**

   A list of categories is displayed.

4. **(Optional) Select the appropriate toolbox for your name service environment.**

5. **Click the System Configuration icon.**

6. **Click the User Accounts icon.**

7. **Provide the superuser password or the role password.**

8. **Click the Groups icon.**

9. **Select the group to delete.**

   For example, select `scutter`.

10. **Click OK in the popup window.**

    The group is removed from all the users who were a member of this group.

Administering Passwords

You can use the Users tool for password administration. This tool includes the following capabilities:

- Specifying a normal password for a user account
- Enabling users to create their own passwords during their first login
- Disabling or locking a user account
- Specifying expiration dates and password aging information

Note – Password aging is not supported by the NIS name service.

Using Password Aging

If you are using NIS+ or the `/etc` files to store user account information, you can set up password aging on a user's password. Starting in the Solaris 9 12/02 release, password aging is also supported in the LDAP directory service.

Password aging enables you to force users to change their passwords periodically or to prevent a user from changing a password before a specified interval. If you want to prevent an intruder from gaining undetected access to the system by using an old and inactive account, you can also set a password expiration date when the account becomes disabled. You can set password aging attributes with the `passwd` command or the Solaris Management Console's Users tool.

For information about starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44.

▼ How to Disable a User Account

Use the following procedure if you need to disable a user account.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start the Solaris Management Console.**

   ```
   # /usr/sadm/bin/smc &
   ```

   For more information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44 or "How to Start the Solaris Management Console in a Name Service Environment" on page 51.

3. **Click the This Computer icon under the Management Tools icon in the Navigation pane.**

   A list of categories is displayed.

4. **(Optional) Select the appropriate toolbox for your name service environment.**

5. **Click the System Configuration icon.**

6. **Click the User Accounts icon.**

7. **Provide the superuser password or the role password.**

8. **Click the Users icon.**

9. **Double–click the user.**

   For example, select `scutter2`.

10. **Select the Account is Locked option in the Account Availability section of the General tab features.**

11. **Click OK.**

▼ How to Change a User's Password

Use the following procedure when a user forgets her password.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start the Solaris Management Console.**

   ```
   # /usr/sadm/bin/smc &
   ```

   For more information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44 or "How to Start the Solaris Management Console in a Name Service Environment" on page 51.

3. **Click the This Computer icon under the Management Tools icon in the Navigation pane.**

   A list of categories is displayed.

4. **(Optional) Select the appropriate toolbox for your name service environment.**

5. **Click the System Configuration icon.**

6. **Click the User Accounts icon.**

7. **Provide the superuser password or the role password.**

8. **Click the Users icon.**

9. **Double–click the user who needs a new password.**

   For example, select `scutter1`.

10. **Select the Password Tab.**

11. **Select the User Must Use This Password at Next Login option.**

12. **Enter the user's new password.**

13. **Click OK.**

▼ How to Set Password Aging on a User Account

Use the following procedure to set password aging on a user account.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide:*

Security Services.

2. **Start the Solaris Management Console.**

   ```
   # /usr/sadm/bin/smc &
   ```

   For more information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44 or "How to Start the Solaris Management Console in a Name Service Environment" on page 51.

3. **Click the This Computer icon under the Management Tools icon in the Navigation pane.**

   A list of categories is displayed.

4. **(Optional) Select the appropriate toolbox for your name service environment.**

5. **Click the System Configuration icon.**

6. **Click the User Accounts icon.**

7. **Provide the superuser password or the role password.**

8. **Click the Users icon.**

9. **Double–click the user.**

   For example, select `scutter2`.

10. **Select the Password Options Tab.**

11. **Select the appropriate Password Options in Days option.**

    For example, select Users Must Change Within to set a date when the user must change his or her password.

12. **Click OK.**

▼ How to Delete a User Account

Use the following procedure to remove a user account.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start the Solaris Management Console.**

   ```
   # /usr/sadm/bin/smc &
   ```

   For more information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 44 or "How to Start the Solaris

Management Console in a Name Service Environment" on page 51.

3. **Click the This Computer icon under the Management Tools icon in the Navigation pane.**
   A list of categories is displayed.
4. **(Optional) Select the appropriate toolbox for your name service environment.**
5. **Click the System Configuration icon.**
6. **Click the User Accounts icon.**
7. **Provide the superuser password or the role password.**
8. **Click the Users icon.**
9. **Double–click the user account to be removed.**
   For example, select `scutter4`.
10. **Click Delete in the popup window if you are sure you want to remove the user account.**
    You are prompted to remove the user's home directory and mailbox contents.

CHAPTER 6

Managing Client-Server Support (Overview)

This chapter describes the management of server and client support on a network. Overview information is provided about each system configuration (referred to as a *system type*) that is supported in the Solaris Operating System. This chapter also includes guidelines for selecting the appropriate system type to meet your needs.

This is a list of the overview information in this chapter.

For step-by-step instructions about how to manage diskless client support, see Chapter 7.

Where to Find Client-Server Tasks

Use this table to find step-by-step instructions for setting up server and client support.

Client-Server Services	For More Information
Install or JumpStart clients	*Solaris 10 Installation Guide: Network-Based Installations*
Diskless client systems in the Solaris OS	"Diskless Client Management Overview" on page 124 and Chapter 7

Client-Server Services	For More Information
Diskless client systems in previous Solaris releases	*Solstice AdminSuite 2.3 Administration Guide*

What Are Servers, Clients, and Appliances?

Systems on the network can usually be described as one of the system types in this table.

System Type	Description
Server	A system that provides services to other systems in its network. There are file servers, boot servers, web servers, database servers, license servers, print servers, installation servers, appliance servers, and even servers for particular applications. This chapter uses the term *server* to mean a system that provides boot services and file systems for other systems on the network.
Client	A system that uses remote services from a server. Some clients have limited disk storage capacity, or perhaps none at all. Such clients must rely on remote file systems from a server to function. Diskless systems and appliance systems are examples of this type of client. Other clients might use remote services (such as installation software) from a server. However, they don't rely on a server to function. A stand-alone system is a good example of this type of client. A stand-alone system has its own hard disk that contains the root (/), `/usr`, and `/export/home` file systems and swap space.
Appliance	A network appliance such as the Sun Ray appliance provides access to applications and the Solaris OS. An appliance gives you centralized server administration, and no client administration or upgrades. Sun Ray appliances also provide *hot desking*. Hot desking enables you to instantly access your computing session from any appliance in the server group, exactly where you left off. For more information, see http://www.sun.com/products/sunray.

What Does Client Support Mean?

Support for a client means providing software and services to help the client function. Support can include the following:

- Making a system known to the network (host name and Ethernet address information)
- Providing installation services to remotely boot and install a system
- Providing Solaris OS services and application services to a system with limited disk space or no disk space

Overview of System Types

System types are sometimes defined by how they access the root (/) and `/usr` file systems, including the swap area. For example, stand-alone systems and server systems mount these file systems from a local disk. Other clients mount the file systems remotely, relying on servers to provide these services. This table lists some of the characteristics of each system type.

TABLE 6–1 Characteristics of System Types

System Type	Local File Systems	Local Swap Space?	Remote File Systems	Network Use	Relative Performance
Server	root (/)	Yes	None	High	High
	`/usr`				
	`/home`				
	`/opt`				
	`/export/home`				
	`/export/root`				
Stand-alone system	root (/)	Yes	None	Low	High
	`/usr`				
	`/export/home`				

TABLE 6–1 Characteristics of System Types *(Continued)*

System Type	Local File Systems	Local Swap Space?	Remote File Systems	Network Use	Relative Performance
Diskless client	None	No	root (/) swap /usr /home	High	Low
Appliance	None	None	None	High	High

Servers

A server system contains the following file systems:

- The `root` (`/`) and `/usr` file systems, plus swap space
- The `/export` and `/export/home` file systems, which support client systems and provide home directories for users
- The `/opt` directory or file system for storing application software

Servers can also contain the following software to support other systems:

- Solaris OS services for diskless systems that are running a different release
- Clients that use a different platform than the server
- Solaris CD image software and boot software for networked systems to perform remote installations
- JumpStart™ directory for networked systems to perform custom JumpStart installations

Stand-Alone Systems

A *networked stand-alone system* can share information with other systems in the network. However, can continue to function if detached from the network.

A stand-alone system can function autonomously because it has its own hard disk that contains the root (`/`), `/usr`, and `/export/home` file systems and swap space. Thus, the stand-alone system has local access to OS software, executables, virtual memory space, and user-created files.

Note – A stand-alone system requires sufficient disk space to hold its necessary file systems.

A *non-networked stand-alone system* is a stand-alone system with all the characteristics just listed, except it is not connected to a network.

Diskless Clients

A *diskless client* has no disk and depends on a server for all its software and storage needs. A diskless client remotely mounts its root (/), /usr, and /home file systems from a server.

A diskless client generates significant network traffic due to its continual need to procure OS software and virtual memory space from across the network. A diskless client cannot operate if it is detached from the network or if its server malfunctions.

For more overview information about diskless clients, see "Diskless Client Management Overview" on page 124.

Appliances

An appliance, such as the Sun Ray™ appliance, is an X display device that requires no administration. There is no CPU, fan, disk, and very little memory. An appliance is connected to a Sun display monitor. However, the appliance user's desktop session is run on a server and displayed back to the user. The X environment is set up automatically for the user and has the following characteristics:

- Relies on a server to access other file systems and software applications
- Provides centralized software administration and resource sharing
- Contains no permanent data, making it a field-replaceable unit (FRU)

Guidelines for Choosing System Types

You can determine which system types are appropriate for your environment by comparing each system type based on the following characteristics:

- **Centralized administration**
  - Can the system be treated as a field-replaceable unit (FRU)? This means that a broken system can be quickly replaced with a new system without any lengthy backup and restore operations and no loss of system data.
  - Does the system need to be backed up? Large costs in terms of time and resources can be associated with backing up a large number of desktop systems.
  - Can the system's data be modified from a central server?
  - Can the system be installed quickly and easily from a centralized server without handling the client system's hardware?
- **Performance**
  - Does this configuration perform well in desktop usage?
  - Does the addition of systems on a network affect the performance of other systems already on the network?

- **Disk space usage**
  - How much disk space is required to effectively deploy this configuration?

This table describes how each system type scores in terms of each characteristic. A ranking of 1 is most efficient. A ranking of 4 is least efficient.

TABLE 6–2 Comparison of System Types

System Type	Centralized Administration	Performance	Disk Space Usage
Stand-alone system	4	1	4
Diskless client	1	4	1
Appliance	1	1	1

Diskless Client Management Overview

The following sections and Chapter 7 describe how to manage diskless client support in the Solaris Operating System (Solaris OS).

A *diskless client* is a system that depends on an *OS server* for its operating system, software, and storage. A diskless client mounts its root (/), `/usr`, and other file systems from its OS server. A diskless client has its own CPU and physical memory and can process data locally. However, a diskless client cannot operate if it is detached from its network or if its OS server malfunctions. A diskless client generates significant network traffic because of its continual need to function across the network.

In previous Solaris releases, diskless clients were managed with the Solstice™ graphical user interface (GUI) management tools. In this Solaris release, the diskless client commands, `smosservice` and `smdiskless`, enable you to manage OS services and diskless client support.

OS Server and Diskless Client Support Information

The following table describes which Solaris releases and architecture types are supported by the `smosservice` and `smdiskless` commands.

Architecture Type	Solaris 7	Solaris 8	Solaris 9	Solaris 10
SPARC servers	Supported	Supported	Supported	Supported

Architecture Type	Solaris 7	Solaris 8	Solaris 9	Solaris 10
x86 based servers	Supported	Supported	Supported	Supported
SPARC based clients	Supported	Supported	Supported	Supported
x86 based clients	Not supported	Not supported	Supported	Supported

This table describes the combination of OS client-server configurations that are supported by the `smosservice` and `smdiskless` commands.

	Solaris 7 Server OS/Solaris Release	Solaris 8 Server OS/Solaris Release	Solaris 9 Server OS/Solaris Release	Solaris 10 Server OS/Solaris Release
Server OS Support for Client OS Release	Solaris 7 Server OS – Supports all Solaris 7 releases	Solaris 8 Server OS – Supports all Solaris 8 and Solaris 7 releases	Solaris 9 Server OS – Supports all Solaris 9, Solaris 8, and Solaris 7 releases	Solaris 10 Server OS – Supports all Solaris 10, Solaris 9, Solaris 8, and Solaris 7 releases

Diskless Client Management Features

You can use the `smosservice` and `smdiskless` commands to add and maintain diskless client support on a network. By using a name service, you can manage system information in a centralized manner so that important system information, such as host names, does not have to be duplicated for every system on the network.

You can perform the following tasks with the `smosservice` and `smdiskless` commands:

- Add and modify diskless client support
- Add and remove OS services
- Manage diskless client information in the LDAP, NIS, NIS+, or files name service environment

---

Note – You can only use the diskless client commands to set up diskless client booting. You cannot use these commands to set up other services, such as remote installation or profile services. Set up remote installation services by including diskless client specifications in the `sysidcfg` file. For more information, see *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

Working With Diskless Client Commands

By writing your own shell scripts and using the commands shown in the following table, you can easily set up and manage your diskless client environment.

TABLE 6–3 Diskless Client Commands

Command	Subcommand	Task
`/usr/sadm/bin/smosservice`	`add`	Add OS services
	`delete`	Delete OS services
	`list`	List OS services
	`patch`	Manage OS service patches
`/usr/sadm/bin/smdiskless`	`add`	Add a diskless client to an OS server
	`delete`	Delete a diskless client from an OS server
	`list`	List the diskless clients on an OS server
	`modify`	Modify the attributes of a diskless client

You can obtain help on these commands in two ways:

- Use the `-h` option when you type the command, subcommand, and required options. For example, to display the usage statement for `smdiskless add`, type the following command:

  ```
  % /usr/sadm/bin/smdiskless add -p my-password -u my-user-name -- -h
  ```

- View the `smdiskless`(1M) and `smosservice`(1M) man pages.

Required RBAC Rights for Diskless Client Management

You can use the `smosservice` and `smdiskless` commands as superuser. If you are using role-based access control (RBAC), you can use either a subset of or all of the diskless client commands, according to the RBAC rights to which they are assigned. The following table lists the RBAC rights that are required to use the diskless client commands.

TABLE 6–4 Required RBAC Rights for Diskless Client Management

RBAC Right	Command	Task
Basic Solaris User, Network Management	`smosservice list`	List OS services
	`smosservice patch`	List OS service patches
	`smdiskless list`	List diskless clients on an OS server
Network Management	`smdiskless add`	Add diskless clients
System Administrator	All commands	All tasks

Adding OS Services

A Solaris OS server is a server that provides operating system (OS) services to support diskless client systems. You can add support for an OS server or convert a stand-alone system to an OS server by using the `smosservice` command.

For each platform group and Solaris release that you want to support, you must add the particular OS service to the OS server. For example, if you want to support SPARC Sun-4u systems running the Solaris 10 release, you must add Sun-4mu/Solaris 10 OS services to the OS server. You would also still need to add OS services to support SPARC Sun-4m systems or x86 based systems that run the Solaris 9 release, because they are different platform groups.

You must have access to the appropriate Solaris software CD or disk image to add OS services.

Adding OS Services When the OS Server Has Been Patched

When adding OS services to an OS server, you might see an error message stating that you have inconsistent versions of the OS running on the server and the OS that you are trying to add. This error message occurs when the installed version of the OS has packages that were previously patched, and the OS services being added do not have those packages patched, because the patches have been integrated into the packages.

For example, you might have a server that is running the Solaris 10 release. You might also have additional OS services loaded on this server, including the Solaris 9 SPARC sun-4m OS services that have been patched. If you try to add the Solaris 8 SPARC sun-4u OS services from a CD-ROM to this server, you could get the following error message:

```
Error: inconsistent revision, installed package appears to have been
patched resulting in it being different than the package on your media.
You will need to backout all patches that patch this package before
retrying the add OS service option.
```

Disk Space Requirements for OS Servers

Before you set up your diskless client environment, ensure that you have the required disk space available for each diskless client directory.

In previous Solaris releases, you were prompted about diskless client support during the installation process. In the Solaris 10 and Solaris 9 releases, you must manually allocate an `/export` file system either during installation or create it after installation. See the following table for specific disk space requirements.

TABLE 6–5 Disk Space Requirements for Solaris 10 OS Servers

Server OS/Architecture Type	Directory	Required Disk Space
Solaris 10 SPARC based OS server	`/export`	2 Gbytes
Solaris 10 x86 based OS server	`/export`	1.6 Gbytes
Solaris 10 SPARC based diskless client	`/export`	300 Mbytes
Solaris 10 x86 based diskless client	`/export`	200 Mbytes

CHAPTER 7

Managing Diskless Clients (Tasks)

This chapter describes how to manage diskless clients in the Solaris Operating System (Solaris OS).

For information on the procedures associated with managing diskless clients, see "Managing Diskless Clients (Task Map)" on page 129. For overview information on managing diskless clients, see Chapter 6.

Managing Diskless Clients (Task Map)

The following table identifies the procedures that are required to manage diskless clients.

Task	Description	For Instructions
1. (Optional) Enable Solaris Management Console logging to view diskless client error messages.	Choose Log Viewer from the console main window to view diskless client error messages.	"Starting the Solaris Management Console" on page 44
2. Prepare for adding a diskless client.	Verify supported releases and identify the platform, media path, and cluster (or software group) of each diskless client.	"How to Prepare for Adding Diskless Clients" on page 131

Task	Description	For Instructions
3. Add required OS services to an OS server.	Add the OS services for the diskless clients you want to support by using the `smosservice` command. You must identify the platform, media path, and each diskless client platform that you want to support.	"How to Add OS Services for Diskless Client Support" on page 133
4. Add a diskless client.	Add diskless client support by specifying all required information by using the `smdiskless` command.	"How to Add a Diskless Client" on page 135
5. Boot the diskless client.	Verify that a diskless client was successfully added by booting the diskless client.	"How to Boot a Diskless Client" on page 136
6. (Optional) Delete diskless client support.	Delete support for a diskless client if it is no longer required.	"How to Remove Diskless Client Support" on page 137
7. (Optional) Delete OS services for a diskless client.	Delete OS services for a diskless client if they are no longer needed.	"How to Remove OS Services for Diskless Clients" on page 137
8. (Optional) Patch OS services.	Add, delete, list, or synchronize patches for diskless client OS services.	"How to Add an OS Patch for a Diskless Client" on page 139

Preparing for Managing Diskless Clients

These sections describe the preparations that are necessary for managing diskless clients.

Keep the following key points in mind when managing diskless clients:

- The Solaris installation program doesn't prompt you to set up diskless client support. You must manually create an `/export` partition to support diskless clients. You create the `/export` partition during or after the installation process.
- The `/export` partition must contain a minimum of 800–1000 Mbytes, depending upon the number of clients supported. For specific information, see "Disk Space Requirements for OS Servers" on page 128.
- The name service identified in the `smosservice` or `smdiskless` commands must match the primary name service identified in the `/etc/nsswitch.conf` file. If you don't specify a name service in the `smdiskless` or `smosservice` commands,

the default name service is `files`.

- You cannot add OS/diskless client services to a UFS file system that resides on an EFI—labeled disk. As a result, you cannot provide client services on a multiterabyte UFS file system.
- The OS server and the diskless client must be on the same subnet.

After you determine the platform, media path, and cluster for each diskless client, you are ready to add OS services. The following directories are created and populated for each OS service that you add:

- `/export/Solaris_`*version*`/Solaris_`*version*`-`*instruction-set*`.all` (symbolic link to `/export/exec/Solaris_`*version*`/Solaris_`*version*`-`*instruction-set*`.all`)
- `/export/Solaris_`*version*
- `/export/Solaris_`*version*`/var`
- `/export/Solaris_`*version*`/opt`
- `/export/share`
- `/export/root/templates/Solaris_`*version*
- `/export/root/clone`
- `/export/root/clone/Solaris_`*version*
- `/export/root/clone/Solaris_`*version*`/`*machine-class*

The following default directories are created and populated on the OS server for each diskless client that you add:

- `/export/root/`*diskless-client*
- `/export/swap/`*diskless-client*
- `/tftpboot/`*diskless-client-ipaddress-in-hex*`/export/dump/`*diskless-client* (if you specify the `-x` *dump* option)

Note – You can modify the default locations of the root (`/`), `/swap`, and `/dump` directories by using the `-x` option. However, do not create these directories under the `/export` file system.

▼ How to Prepare for Adding Diskless Clients

When you use the `smosservice add` command to add OS services, you must specify the platform, media path, and cluster (or software group) of each diskless client platform that you want to support.

Before You Begin

Ensure that the system that is intended to be the OS service is running a supported release. Also verify that the combination of OS server release and diskless client release is supported. For more information, see "OS Server and Diskless Client Support Information" on page 124.

Steps 1. **Identify the diskless client platform by using this format:**

instruction-set.machine-class.Solaris_version

For example:

`sparc.sun4u.Solaris_10`

The following are the possible platform options:

instruction-set	*machine-class*	*Solaris_version*
sparc	sun4c, sun4d, sun4m, sun4u,	Solaris_10, Solaris_9, Solaris_8, Solaris_2.7
i386	i86pc	Solaris_10, Solaris_9, Solaris_8, Solaris_2.7

Note – The sun-4c architecture is not supported in the Solaris 8, Solaris 9, or Solaris 10 releases. The sun-4d architecture is not supported in the Solaris 9 or 10 releases. The sun-4m architecture is not supported in the Solaris 10 release.

2. **Identify the media path.**

The media path is the full path to the disk image that contains the OS that you want to install for the diskless client.

The Solaris OS is delivered on multiple CDs. However, you cannot use the `smosservice` command to load OS services from a multiple CD distribution. You must run the scripts that are found on the Solaris software CDs (and optional Language CD) to do the following:

3. **Create an install image on a server. For information on setting up an install server, refer to** *Solaris 10 Installation Guide: Network-Based Installations***.**

4. **Load the required OS services from the CD image by using one of the following scripts:**

- CD 1 – `/cdrom/cdrom0/s0/Solaris_10/Tools/setup_install_server`
- Additional Solaris Software CDs – `/cdrom/cdrom0/s0/Solaris_10/Tools/add_to_install_server`
- Language CD – `/cdrom/cdrom0/s0/Solaris_10/Tools/add_to_install_server`

For example, if you are using the `setup_install_server` script from the Solaris 10 Software 1 CD on a locally connected CD-ROM device, the syntax looks similar to the following:

```
# mkdir /export/install/sparc_10
# cd /cd_mount_point/Solaris_10/Tools
```

```
# ./setup_install_server /export/install/sparc_10
```

5. **After the Solaris CD image is installed on the disk, specify the disk media path. For example:**

   ```
   /net/export/install/sparc_10
   ```

6. **Identify the `SUNWCXall` cluster when you add OS services.**

   You must use the same cluster for diskless clients that run the same OS on the same system.

   For example, consider the following Solaris 9 diskless clients:

   - `sparc.sun4m.Solaris_9`
   - `sparc.sun4u.Solaris_9`

   To set up these diskless clients, you would need to specify the `SUNWCXall` cluster for each diskless client because the `sun4u` and `sun4m` systems require the `SUNWCXall` cluster. In addition, diskless clients that run the same operating release (in this example, Solaris_9) on the same system must use the same cluster.

Note – If you are using a `sun4u` system, or if you are using a system with an accelerated 8-bit color memory frame buffer (`cgsix`), you *must* specify `SUNWCXall` as the cluster.

▼ How to Add OS Services for Diskless Client Support

Use this procedure to add OS services for a diskless client on the server.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Verify that the Solaris Management Console server is running and that the diskless client tools are available on the system.**

   ```
   # /usr/sadm/bin/smosservice list -H host-name:898 --
   ```

3. **Add the OS services.**

   ```
   # /usr/sadm/bin/smosservice add -H host-name:898 -- -o host-name
   -x mediapath=path -x platform=instruction-set.machine-class.Solaris_version
   -x cluster=cluster-name -x locale=locale-name
   ```

   `add`
   Adds the specified OS service.

`-H` *host-name*:898
Specifies the host name and port to which you want to connect. If you do not specify a port, the system connects to the default port, 898.

`--`
Identifies that the subcommand arguments start after this point.

`-x` *mediapath=path*
Specifies the full path to the Solaris image.

`-x` *platform=instruction-set.machine-class.Solaris_version*
Specifies the instruction architecture, machine class, and the Solaris version to be added.

`-x` *cluster=cluster-name*
Specifies the Solaris cluster to install.

`-x` *locale=locale-name*
Specifies the locale to install.

Note – The installation process can take about 45 minutes, depending on the server speed and the OS service configuration you choose.

For more information, see the `smosservice`(1M) man page.

4. **(Optional) Continue to add the other OS services.**

5. **When you are finished adding OS services, verify that the OS services were installed.**

   # **/usr/sadm/bin/smosservice list -H** *host-name*:**898 --**

Example 7–1 SPARC: Adding an OS Service for Diskless Client Support

This example shows how to add Solaris 10 (SPARC based) OS services on the server `jupiter`. The server `jupiter` is running the Solaris 10 release.

```
# /usr/sadm/bin/smosservice add -H jupiter:898 -- -o jupiter
-x mediapath=/net/install/export/s10/combined-s10s_wos/61
-x platform=sparc.sun4u.Solaris_10
-x cluster=SUNWCXall -x locale=en_US

# /usr/sadm/bin/smosservice list - H jupiter:898
Authenticating as user: root

Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password :: xxxxx
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli
from jupiter:898
Login to jupiter as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from jupiter:898
```

```
was successful.
```

Example 7–2 x86: Adding an OS Service for Diskless Client Support

This example shows how to add Solaris 10 (x86 based) OS services on the server `orbit`. The server `orbit` is running the Solaris 10 release.

```
# /usr/sadm/bin/smosservice add -H orbit:898 -- -o orbit -x
mediapath=/net/install/export/s10/combined.s10x_wos/61 -x
platform=i386.i86pc.Solaris_10 -x cluster=SUNWCXall -x locale=en_US
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from orbit:898
Login to orbit as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from
orbit:898 was successful.
```

▼ How to Add a Diskless Client

Use this procedure to add a diskless client after you have added OS services.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Add the diskless client.**

   ```
   # /usr/sadm/bin/smdiskless add -- -i ip-address -e ethernet-address
   -n client-name -x os=instruction-set.machine-class.Solaris_version
   -x root=/export/root/client-name -x swap=/export/swap/client-name
   -x swapsize=size -x tz=time-zone -x locale=locale-name
   ```

   `add`
   Adds the specified diskless client.

   `--`
   Identifies that the subcommand arguments start after this point.

   `-i` *ip-address*
   Identifies the IP address of the diskless client.

   `-e` *ethernet-address*
   Identifies the Ethernet address of the diskless client.

   `-n` *client-name*
   Specifies the name of the diskless client.

   `-x os=`*instruction-set.machine-class.Solaris_version*
   Specifies the instruction architecture, machine class, OS, and the Solaris version for the diskless client.

`-x root=root=/export/root/`*client-name*
Identifies the root (/) directory for the diskless client.

`-x swap=root=/export/root/`*client-name*
Identifies the swap file for the diskless client.

`-x swapsize=`*size*
Specifies the size of the swap file in Mbytes. The default is 24 Mbytes.

`-x tz=`*time-zone*
Specifies the time-zone for the diskless client.

`-x locale=`*locale-name*
Specifies the locale to install for the diskless client.

For more information, see the `smdiskless`(1M) man page.

3. **(Optional) Continue to use the `smdiskless add` command to add each diskless client.**

4. **Verify that the diskless clients were installed.**

```
# /usr/sadm/bin/smosservice list -H host-name:898 --
```

Example 7–3 Adding Diskless Client Support to a SPARC Based System

This example shows how to add Solaris 10 sun4u diskless client, `starlite`, from the server `bearclaus`.

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.28 -e 8:0:20:a6:d4:5b
-n starlite -x os=sparc.sun4u.Solaris_10 -x root=/export/root/starlite
-x swap=/export/swap/starlite -x swapsize=128 -x tz=US/Mountain
-x locale=en_US
```

Example 7–4 Adding Diskless Client Support to an x86 Based System

This example shows how to add a Solaris 10 x86 based diskless client, `mars`, from the server `bearclaus`.

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.176 -e 00:07:E9:23:56:48
-n mars -x os=i386.i86pc.Solaris_10 -x root=/export/root/mars
-x swap=/export/swap/mars -x swapsize=128 -x tz=US/Mountain
-x locale=en_US
```

▼ How to Boot a Diskless Client

Before You Begin

Verify the following prerequisites on the OS server:

- Confirm that the name service used to add the diskless client and the OS services matches the primary name in the server's `/etc/nsswitch.conf` file.

  Otherwise, the diskless client won't boot.

- Confirm that the `rpc.bootparamd` daemon is running. If it is not running, start it.

Step

- **Boot the diskless client.**

  ```
  ok boot net
  ```

▼ How to Remove Diskless Client Support

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Remove the diskless client support.**

   ```
   # /usr/sadm/bin/smdiskless delete -- -o host-name:898 -n client-name
   ```

3. **Verify that the diskless client support has been removed.**

   ```
   # /usr/sadm/bin/smosservice list -H host-name:898 --
   ```

Example 7–5 Removing Diskless Client Support

This example shows how to remove the diskless client `holoship` from the OS server `starlite`.

```
# /usr/sadm/bin/smdiskless delete -- -o starlite -n holoship
Authenticating as user: root

Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password ::
Starting SMC server version 2.0.0.
endpoint created: :898
SMC server is ready.

# /usr/sadm/bin/smosservice list -H host-name:898
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite
Login to starlite as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite
was successful.
```

▼ How to Remove OS Services for Diskless Clients

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Remove the OS services for the diskless clients.**

   # **/usr/sadm/bin/smosservice delete -H** *host-name*:898 --
   -x rmplatform=*instruction-set.machine-class.Solaris_version*

3. **Verify that the OS services have been removed.**

   # **/usr/sadm/bin/smosservice list -H** *host-name*:898 --

Example 7–6 Removing OS Services for Diskless Clients

The following example shows how to removing the diskless client OS services (`sparc.all.Solaris_9`) from the server `starlite`.

```
# /usr/sadm/bin/smosservice delete -H starlite:898 --
-x rmplatform=sparc.all.Solaris_9
Authenticating as user: root
Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password ::

# /usr/sadm/bin/smosservice list -H host-name:898 --
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite:898
Login to starlite as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite:898
was successful.
```

Patching Diskless Client OS Services

You use the `smosservice patch` command to do the following:

- Establish the `/export/diskless/Patches` patch spool directory on an OS server.
- Add patches to the patch spool directory. If the patch you are adding obsoletes an existing patch in the spool, the obsolete patch is moved to `/export/diskless/Patches/Archive`.
- Delete patches from the patch spool directory.
- List the patches in the patch spool directory.
- Synchronize spooled patches out to clients. You must reboot each synchronized client for the client to recognize the patch update.

Note – Keep your OS servers up to date by installing recommended OS patches on a timely basis.

For information on downloading patches, see "How to Download and Apply a Solaris Patch" on page 360.

Displaying OS Patches for Diskless Clients

Diskless client patches are logged in different directories, depending on the type of patch:

- Kernel patches are logged in the diskless client's `/var/sadm/patch` directory. To display kernel patches, type the following command on the diskless client:

  ```
  % patchadd -p
  ```

- `/usr` patches are logged in the OS server's `/export/Solaris_`*version*`/var/patch` directory. A directory is created for each patch ID. To display `/usr` patches, type the following command on the OS server:

  ```
  % patchadd -S Solaris_version -p
  Patch: 111879-01 Obsoletes: Requires: Incompatibles: Packages: SUNWwsr
  ```

To list all spooled patches by OS and architecture, use the `smosservice` command with the `-P` option.

▼ How to Add an OS Patch for a Diskless Client

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Log in to the diskless client system and shut it down.**

   ```
   # init 0
   ```

3. **Add the patch to a spool directory.**

   ```
   # /usr/sadm/bin/smosservice patch -- -a /var/patches/patch-ID-revision
   ```

   If the patch to add depends on another patch, adding the patch fails with the following message:

   ```
   The patch patch-ID-revision could not be added
   because it is dependent on other patches which have not yet been spooled.
   You must add all required patches to the spool first.
   ```

4. **Verify that the patch has been spooled.**

   ```
   # /usr/sadm/bin/smosservice patch -- -P
   ```

5. **Push the spooled patch to the diskless client.**

   ```
   # /usr/sadm/bin/smosservice patch -- -m -U
   ```

Note – Pushing and synchronizing the patch to the diskless client can take up to 90 minutes per patch.

6. **Verify the patch is applied to the diskless client.**

```
# /usr/sadm/bin/smosservice patch -- -P
```

Example 7–7 Adding an OS Patch for a Diskless Client

This example shows how to add a Solaris 8 patch (111879-01) to the diskless client's OS services on the server.

```
# /usr/sadm/bin/smosservice patch -- -a /var/patches/111879-01
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite
Login to starlite as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite
was successful..
.
# /usr/sadm/bin/smosservice patch -- -P
Patches In Spool Area
Os Rel Arch   Patch Id  Synopsis
--------------------------------------------------------------------------
8      sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr

Patches Applied To OS Services
Os Service                              Patch
--------------------------------------------------------------------------
Solaris_8

Patches Applied To Clone Areas
Clone Area                              Patch
--------------------------------------------------------------------------
Solaris_8/sun4u          Patches In Spool Area
Os Rel Arch   Patch Id  Synopsis
--------------------------------------------------------------------------
8      sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr
.
.
.
# /usr/sadm/bin/smosservice patch -- -m -U
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite
Login to starlite as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starlite
```

```
was successful.

# /usr/sadm/bin/smosservice patch -- -P
Authenticating as user: root
.
.
.
Patches In Spool Area
Os Rel Arch   Patch Id  Synopsis
--------------------------------------------------------------------------------
8      sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr

Patches Applied To OS Services
Os Service                              Patch
--------------------------------------------------------------------------------
Solaris_8

Patches Applied To Clone Areas
Clone Area                              Patch
--------------------------------------------------------------------------------
Solaris_8/sun4u
```

Troubleshooting Diskless Client Problems

This section lists some common problems with diskless clients and possible solutions.

Problem: The OS server fails to do the following:

- Respond to client Reverse Address Resolution Protocol (RARP) requests
- Respond to client `bootparam` requests
- Mount a diskless client root (`/`) file system

Solution: The following solutions apply in a files environment.

- Verify that `files` is listed as the first source for `hosts`, `ethers`, and `bootparams` in the `/etc/nsswitch.conf` file on the OS server.
- Verify that the client's IP address appears in the `/etc/inet/hosts` file.
- Verify that the client's Ethernet address appears in the `/etc/ethers` file.
- Verify that the `/etc/bootparams` file contains the following paths to the client's root (`/`) directory and swap areas.

  client `root=`*os-server*`:/export/root/`*client* `swap=`*os-server*`:/export/swap/`*client*

  The swap size varies depending on whether you specify the `-x` *swapsize* option when you add the diskless client. If you specify the `-x` *dump* option when you add the diskless client, the following line is present:

```
dump=os-server:/export/dump/client dumpsize=24
```

The dump size varies depending on whether you specify the `-x` *dumpsize* option when you add the diskless client.

- Verify that the OS server's IP address appears in the `/export/root/`*client*`/etc/inet/hosts` file.

Problem: The OS server fails to do the following:

- Respond to client RARP requests
- Respond to client `bootparam` requests
- Mount a diskless client root (`/`) file system

Solution: The following solutions apply in a name service environment.

- Verify that both the OS server's and the client's Ethernet address and IP address are correctly mapped.
- Verify that the `/etc/bootparams` file contains the paths to the client's root (`/`) directory and swap areas.

```
client root=os-server:/export/
root/client swap=os-server:/export/
swap/client swapsize=24
```

The swap size varies depending on whether you specify the `-x` *swapsize* option when you add the diskless client. If you specify the `-x` *dump* option when you add the diskless client, the following line is present:

```
dump=os-server:/export/dump/client dumpsize=24
```

The dump size varies depending on whether you specify the `-x` *dumpsize* option when you add the diskless client.

Problem: Diskless client panics

Solution: Verify the following:

- The OS server's Ethernet address is correctly mapped to its IP address. If you physically moved a system from one network to another, you might have forgotten to remap the system's new IP address.
- The client's host name, IP address, and Ethernet address do not exist in the database of another server *on the same subnet* that responds to the client's RARP, Trivial File Transfer Protocol (TFTP), or `bootparam` requests. Often, test systems are set up to install their OS from an install server. In these cases, the install server answers the client's RARP or `bootparam` request, returning an incorrect IP address. This incorrect address might result in the download of a boot program for the wrong architecture, or a failure to mount the client's root (`/`) file system.
- The diskless client's TFTP requests are not answered by an install server (or previous OS server) that transfers an incorrect boot program. If the boot program is of a different architecture, the client immediately panics. If the boot program loads from a non-OS server, the client might obtain its root partition from the non-OS

server and its /usr partition from the OS server. In this situation, the client panics if the root and /usr partitions are of conflicting architectures or versions.

- If you are using both an install server and an OS server, verify that the following entry exists in the /etc/dfs/dfstab file:

  ```
  share -F nfs -o -ro /export/exec/Solaris_version-instruction-set.all/usr
  ```

  where *version*=2.7, 8, 9,10, and *instruction-set*=sparc or i386.

- Verify that the diskless client's root (/), /swap, and /dump (if specified) partitions have share entries:

  ```
  share -F nfs -o rw=client,root=client /export/root/client
  share -F nfs -o rw=client,root=client /export/swap/client
  share -F nfs -o rw=client,root=client /export/dump/client
  ```

- On the OS server, type the following command to check which files are shared:

  ```
  % share
  ```

  The OS server must share /export/root/*client* and /export/swap/*client-name* (defaults), or the root, /swap, and /dump partitions that you specified when you added the diskless client.

  Verify that the following entries exist in the /etc/dfs/dfstab file:

  ```
  share -F nfs -o ro /export/exec/Solaris_version-instruction-set.all/usr
  share -F nfs -o rw=client,root=client /export/root/client
  share -F nfs -o rw=client,root=client /export/swap/client
  ```

Problem: OS server is not responding to diskless client's RARP request

Solution: From the client's intended OS server, run the snoop command as superuser (root) by using the client's Ethernet address:

```
# snoop xx:xx:xx:xx:xx:xx
```

Problem: Boot program downloads but panics early in the process

Solution: Using the snoop command, verify that the intended OS server is answering the client's TFTP and NFS requests.

Problem: Diskless client hangs.

Solution: Restart the following daemons on the OS server:

```
# /usr/sbin/rpc.bootparamd
# /usr/sbin/in.rarpd -a
```

Problem: Incorrect server responds to diskless client's RARP request

Solution: Restart the following daemons on the OS server:

```
# /network/rpc/bootparams
# svcadm enable network/rarp
```

CHAPTER 8

Shutting Down and Booting a System (Overview)

This chapter provides guidelines for shutting down and booting a system. The Solaris Operating System (Solaris OS) is designed to run continuously so that electronic mail and network resources are available to users. Occasionally, shutting down or rebooting a system is necessary because of a system configuration change, a scheduled maintenance event, or a power outage.

This is a list of the overview information in this chapter.

What's New in Shutting Down and Booting a System

This section describes features pertaining to booting a system that are new or changed in this Solaris release.

Booting and the Service Management Facility

The Service Management Facility (SMF) provides new options for booting a system. See "SMF and Booting" on page 163 for more information.

x86: Support for 64-Bit Computing

In the Solaris 10 OS, the system autodetects the appropriate kernel to boot when you type `b` at the `Select (b)oot or (i)nterpreter` boot prompt. New installations of the Solaris 10 OS autoboot to 64-bit mode on 64-bit capable hardware. Upgrade installations of the Solaris 10 OS autoboot to 64-bit mode on 64-bit capable hardware, *unless* the `eeprom boot-file` parameter was previously set to a value other than `kernel/unix`.

Note – For upgrade installations of the Solaris 10 OS, where the `eeprom boot-file` parameter was previously set to a value other than `kernel/unix`, you will need to do one of the following to boot the system to 64-bit mode:

- Manually set the system to boot to 64-bit mode. See "x86: Manually Booting a System That Is Capable of 64-Bit Computing" on page 146.
- Use the `eeprom` command to enable autodetection. See "x86: Setting the `boot-file` Parameter With the `eeprom` Command" on page 146.

x86: Manually Booting a System That Is Capable of 64-Bit Computing

To manually boot a 64-bit capable x86 based system to 64-bit mode, type the following at the `Select (b)oot or (i)nterpreter` boot prompt:

```
b kernel/amd64/unix
```

The command to manually boot a 64-bit capable x86 based system to 32-bit mode remains unchanged from previous Solaris OS versions.

```
b kernel/unix
```

x86: Setting the `boot-file` Parameter With the `eeprom` Command

For all new installations of the Solaris 10 OS, as well as upgrade installations, where the `eeprom boot-file` parameter had previously been set to `kernel/unix`, the `eeprom boot-file` parameter is set to a null value (`""`). The system will then automatically boot to 64-bit mode on x86 based systems that are capable of 64-bit computing when you type `b` at the boot prompt.

You do not need to manually specify which kernel a 64-bit capable system should boot unless one of the following conditions exists:

- The `eeprom boot-file` parameter was previously set to a value other than `kernel/unix`.

- You want to force the system to boot to a particular mode.

The following table shows the boot mode result for an x86 based system that is 64-bit capable, depending on the boot command that is used and how the `eeprom boot-file` parameter is set.

Boot Command	`eeprom boot-file` Parameter Setting	Boot Mode Result
`b kernel/unix`	The `boot-file` parameter is ignored when this command is used.	32-bit mode boot
`b kernel/amd64/unix`	The `boot-file` parameter is ignored when this command is used.	64-bit mode boot
`b`	`""`	64-bit mode boot
`b`	`kernel/unix`	32-bit mode boot
`b`	`kernel/amd64/unix`	64-bit mode boot

To manually specify which mode a 64-bit capable x86 based system boots to on future reboots, set the `eeprom boot-file` parameter. Note that you must be superuser or assume an equivalent role to run the `eeprom` command.

To manually specify that a 64-bit capable x86 system always boot a 64-bit kernel, set the `eeprom boot-file` parameter as follows:

```
# eeprom boot-file kernel/amd64/unix
```

To manually specify that a 64-bit capable x86 system always boot a 32-bit kernel, set the `eeprom boot-file` parameter as follows:

```
# eeprom boot-file kernel/unix
```

To restore the default autodetect boot behavior, type:

```
# eeprom boot-file ""
```

To determine the current `boot-file` parameter, type:

```
$ eeprom boot-file
```

For more information on the `eeprom` command, see the `eeprom`(1M) man page. For information on how to troubleshoot problems on 64-bit capable x86 based systems, see "64-bit x86: Troubleshooting a Failed 64-Bit Boot" on page 217.

x86: Booting a System With the Kernel Debugger (kmdb)

To boot a 64-bit capable x86 based system with kmdb, use the -k option to the boot specification. Although b kmdb is still a valid command, the preferred method is to use b -k.

For example, to boot a 64-bit capable x86 based system to 64-bit mode with kmdb, type the following command at the Select (b)oot or (i)nterpreter boot prompt.

```
b kernel/amd64/unix -k
```

To boot a 64-bit capable x86 based system to 32-bit mode with kmdb, type the following command at the Select (b)oot or (i)nterpreter boot prompt.

```
b kernel/unix -k
```

Note – Typing b kmdb at the boot prompt causes a system to boot the autodetected kernel type with the kernel debugger enabled, regardless of how the eeprom boot-file parameter is set.

Typing b -k at the boot prompt boots whichever kernel the system would otherwise boot if you typed b, with the kernel debugger enabled. The kernel is specified by the eeprom boot-file parameter, or by the autodetected default, if the boot-file parameter is set to a null value ("").

For an example of how to boot a 64-bit capable x86 based system with kmdb, see Example 12–8.

For more information on 64-bit computing on the x86 platform, see the isainfo(1), isalist(1), and sysinfo(2) man pages.

x86: Systems Booting From PXE, CD, or DVD Now Boot Automatically

In this release, when you perform a Preboot Execution Environment (PXE) network boot on an x86 based system, or you boot an x86 based system from the Solaris Software 1 CD or DVD, the system boots automatically. The Device Configuration Assistant menu is no longer displayed by default. If you need to access the Solaris Device Configuration Assistant, press the Escape key to interrupt the autoboot process. Doing so, enables you to access the Device Configuration Assistant menu. For more information, see "x86: How to Boot a System From the Network" on page 208.

For a fully automated JumpStart installation, boot scripts that run the Device Configuration Assistant during the boot process from CD, DVD, or a PXE network boot are no longer necessary.

Where to Find Shut Down and Boot Tasks

Use these references to find step-by-step instructions for shutting down and booting a system.

Shut Down and Boot Task	For More Information
Shut down a SPARC based system or an x86 based system	Chapter 10
Boot a SPARC based system	Chapter 11
Boot an x86 based system	Chapter 12
Manage a SPARC based system by using the power management software	`power.conf`(4), `pmconfig`(1M)

Shut Down and Boot Terminology

This section describes the terminology that is used in shutting down and booting a system.

Run levels and init states

A *run level* is a letter or digit that represents a system state in which a particular set of system services are available. The system is always running in one of a set of well-defined run levels. Run levels are also referred to as *init states* because the `init` process maintains the run level. System administrators use the `init` command or the `svcadm` command to initiate a run-level transition. This book refers to init states as run levels.

Boot options

A *boot option* describes how a system is booted. Different boot options include the following:

- **Interactive boot** – You are prompted to provide information about how the system is booted, such as the kernel and device path name.
- **Reconfiguration boot** – The system is reconfigured to support newly added hardware or new pseudo devices.

- **Recovery boot** – The system is hung or an invalid entry is prohibiting the system from booting successfully or from allowing users to log in.

Guidelines for Shutting Down a System

Keep the following in mind when you shut down a system:

- Use the `init` and `shutdown` commands to shut down a system. Both commands perform a clean system shutdown, which means that all system processes and services are terminated normally.
- Use the `shutdown` command to shut down a server. Logged-in users and systems that mount resources from the server are notified before the server is shut down. Additional notification of system shutdowns by electronic mail is also recommended so that users can prepare for system downtime.
- You need superuser privileges to use the `shutdown` or `init` command to shut down a system.
- Both `shutdown` and `init` commands take a run level as an argument. The three most common run levels are as follows:
  - **Run level 3** – All system resources are available and users can log in. By default, booting a system brings it to run level 3, which is used for normal day-to-day operations. This run level is also known as multiuser level with NFS resources shared.
  - **Run level 6** – Stops the operating system and reboots to the state that is defined by the `initdefault` entry in the `/etc/inittab` file.
  - **Run level 0** – The operating system is shut down, and it is safe to turn off power. You need to bring a system to run level 0 whenever you move a system, or add or remove hardware.

  Run levels are fully described in Chapter 9.

Guidelines for Booting a System

Keep the following in mind when you boot a system:

- SPARC: After a system is shut down, it is booted by using the `boot` command at the PROM level.

- x86: After a system is shut down, it is booted by using the `boot` command at the Primary Boot Subsystem menu.
- A system can be rebooted by turning the power off and then back on.

Caution – This method is not considered a clean shutdown. Use this shutdown method only as an alternative in emergency situations. Because system services and processes are terminated abruptly, file system damage is likely to occur. The work required to repair this type of damage could be substantial and might require the restoration of various user and system files from backup copies.

- SPARC based systems and x86 based systems use different hardware components for booting. These differences are described in Chapter 13.

Booting a System From the Network

You might need to boot a system from the network under the following situations:

- When the system is first installed
- If the system won't boot from the local disk
- If the system is a diskless client

Two network configuration boot strategies are available:

- Reverse Address Resolution Protocol (RARP) and ONC+™ RPC Bootparams Protocol
- Dynamic Host Configuration Protocol (DHCP)

The default network boot strategy for a PXE network device is DHCP. The default network boot strategy for a non-PXE device is RARP.

Note – You cannot change the default network strategy for a PXE device. However, it is possible to configure a non-PXE device to use DHCP.

Use this table if you need information on booting a system over the network.

Network Boot Task	For More Information
Boot a SPARC based system or a SPARC based diskless client.	Chapter 11
Boot an x86 based system or an x86 based diskless client.	Chapter 12
Boot a DHCP client during installation.	*Solaris 10 Installation Guide: Network-Based Installations*
Configure a DHCP client by using DHCP Manager.	*System Administration Guide: IP Services*

x86: PXE Network Boot

You can boot the Solaris 10 Operating System on x86 based systems directly from a network without the Solaris boot diskette on x86 based systems that support the Preboot Execution Environment (PXE) network booting protocol. The PXE network boot is available only for devices that implement the Intel PXE specification. The default network strategy for devices that use PXE is DHCP.

You can enable the PXE network boot on the client system by using the BIOS setup program in the system BIOS, the network adapter BIOS, or both. On some systems, you must also adjust the boot device priority list so that a network boot is attempted before a boot from other devices. See the manufacturer's documentation for each setup program, or watch for setup program entry instructions during boot.

Some PXE-capable network adapters have a feature that enables a PXE boot if you type a particular keystroke in response to a brief boot-time prompt. This feature is ideal when you use PXE for an install boot on a system that normally boots from the disk drive because you do not have to modify the PXE settings. If your adapter does not have this feature, disable PXE in the BIOS setup when the system reboots after installation. Then, the system will boot from the disk drive.

Some early versions of PXE firmware cannot boot the Solaris system. If you have an older version your system can read the PXE network bootstrap program from a boot server. However, the bootstrap will not transmit packets. If this problem occurs, upgrade the PXE firmware on the adapter. Obtain firmware upgrade information from the adapter manufacturer's web site. For more information, see the `elxl`(7D) and `iprb`(7D) man page.

For information on booting x86 based systems with or without the boot diskette, see "x86: How to Boot a System From the Network" on page 208.

When to Shut Down a System

The following table lists system administration tasks and the type of shutdown that is needed to initiate the task.

TABLE 8–1 Shutting Down a System

Reason for System Shutdown	Appropriate Run Level	For More Information
To turn off system power due to anticipated power outage	Run level 0, where it is safe to turn off power	Chapter 10
To change kernel parameters in the `/etc/system` file	Run level 6 (reboot the system)	Chapter 10
To perform file system maintenance, such as backing up or restoring system data	Run level S (single-user level)	Chapter 10
To repair a system configuration file such as `/etc/system`	See "When to Boot a System" on page 154	N/A
To add or remove hardware from the system	Reconfiguration boot (also to turn off power when adding or removing hardware)	"Adding a Peripheral Device to a System" in *System Administration Guide: Devices and File Systems*
To repair an important system file that is causing system boot failure	See "When to Boot a System" on page 154	N/A
To boot the kernel debugger (`kmdb`) to track down a system problem	Run level 0, if possible	Chapter 10
To recover from a hung system and force a crash dump	See "When to Boot a System" on page 154	N/A
Reboot the system by using the kernel debugger (`kmdb`), if the debugger can't be loaded at runtime.	Run level 6 (reboot the system)	"SPARC: How to Boot the System With the Kernel Debugger (`kmdb`)" on page 197,"x86: How to Boot a System With the Kernel Debugger (`kmdb`)" on page 213

For examples of shutting down a server or a stand-alone system, see Chapter 10.

When to Boot a System

The following table lists system administration tasks and the corresponding boot option that is used to complete the task.

TABLE 8–2 Booting a System

Reason for System Reboot	Appropriate Boot Option	Information for SPARC Based System Procedure	Information for x86 Based Systems Procedure
Turn off system power due to anticipated power outage.	Turn system power back on	Chapter 10	Chapter 10
Change kernel parameters in the `/etc/system` file.	Reboot the system to run level 3 (multiuser level with NFS resources shared)	"SPARC: How to Boot a System to Run Level 3 (Multiuser Level)" on page 188	"x86: How to Boot a System to Run Level 3 (Multiuser Level)" on page 202
Perform file system maintenance, such as backing up or restoring system data.	Press Control-D from run level S to bring the system back to run level 3	"SPARC: How to Boot a System to Run Level S (Single-User Level)" on page 189	"x86: How to Boot a System to Run Level S (Single-User Level)" on page 204
Repair a system configuration file such as `/etc/system`.	Interactive boot	"SPARC: How to Boot a System Interactively" on page 190	"x86: How to Boot a System Interactively" on page 206
Add or remove hardware from the system.	Reconfiguration boot (also to turn on system power after adding or removing hardware)	"Adding a System Disk or a Secondary Disk (Task Map)" in *System Administration Guide: Devices and File Systems*	"Adding a System Disk or a Secondary Disk (Task Map)" in *System Administration Guide: Devices and File Systems*
Boot the system by using the kernel debugger (`kmdb`) to track down a system problem.	Booting `kmdb`	"x86: How to Boot a System With the Kernel Debugger (`kmdb`)" on page 213	"x86: How to Boot a System With the Kernel Debugger (`kmdb`)" on page 213
To repair an important system file that is causing system boot failure	Recovery boot	"SPARC: How to Boot a System for Recovery Purposes" on page 193	"x86: How to Boot a System for Recovery Purposes" on page 211
To recover from a hung system and force a crash dump	Recovery boot	See example for "SPARC: How to Force a Crash Dump and Reboot of the System" on page 196	See example for "x86: How to Force a Crash Dump and Reboot of the System" on page 215

CHAPTER 9

Managing Services (Overview)

This chapter provides an overview of the Service Management Facility (SMF). In addition, information that is related to run levels is provided.

This is a list of the overview information in this chapter.

For information on the procedures associated with SMF, see "Managing SMF Services (Task Map)" on page 229. For information on the procedures associated with run levels, see "Using Run Control Scripts (Task Map)" on page 242.

Introduction to SMF

SMF provides an infrastructure that augments the traditional UNIX start-up scripts, `init` run levels, and configuration files. SMF provides the following functions:

- Automatically restarts failed services in dependency order, whether they failed as the result of administrator error, software bug, or were affected by an uncorrectable hardware error. The dependency order is defined by dependency statements.
- Makes services objects that can be viewed, with the new `svcs` command, and managed, with `svcadm` and `svccfg` commands. You can also view the relationships between services and processes using `svcs -p`, for both SMF services and legacy `init.d` scripts.

- Makes it easy to backup, restore, and undo changes to services by taking automatic snapshots of service configurations.
- Makes it easy to debug and ask questions about services by providing an explanation of why a service isn't running by using `svcs -x`. Also, this process is eased by individual and persistent log files for each service.
- Allows for services to be enabled and disabled using `svcadm`. These changes can persist through upgrades and reboots. If the `-t` option is used, the changes are temporary.
- Enhances the ability of administrators to securely delegate tasks to non-root users, including the ability to modify properties and enable, disable, or restart services on the system.
- Boots faster on large systems by starting services in parallel according to the dependencies of the services. The opposite process occurs during shutdown.
- Allows you to customize the boot console output to either be as quiet as possible, which is the default, or to be verbose by using `boot -m verbose`.
- Preserves compatibility with existing administrative practices wherever possible. For example, most customer and ISV-supplied rc scripts still work as usual.

Dependency statements define the relationships between services. These relationships can be used to provide precise fault containment by restarting only those services that are directly affected by a fault, rather than restarting all of the services. Another advantage of dependency statements is that the statements allow for scalable and reproducible initialization processes. In addition, by defining all of the dependencies, you can take advantage of modern, highly parallel machines, because all independent services can be started in parallel.

SMF defines a set of actions that can be invoked on a service by an administrator. These actions include enable, disable, refresh, restart, and maintain. Each service is managed by a service restarter which carries out the administrative actions. In general, the restarters carry out actions by executing methods for a service. Methods for each service are defined in the service configuration repository. These methods allow the restarter to move the service from one state to another state.

The service configuration repository provides a per-service snapshot at the time that each service is successfully started so that fallback is possible. In addition, the repository provides a consistent and persistent way to enable or disable a service, as well as a consistent view of service state. This capability helps you debug service configuration problems.

Changes in Behavior When Using SMF

Most of the features that are provided by SMF happen behind the scenes, so users are not aware of them. Other features are accessed by new commands. Here is a list of the behavior changes that are most visable.

- The boot process creates many fewer messages now. Services do not display a message by default when they are started. All of the information that was provided by the boot messages can now be found in a log file for each service that is in `/var/svc/log`. You can use the `svcs` command to help diagnose boot problems. In addition, you can use the `-v` option to the `boot` command, which generates a message when each service is started during the boot process.
- Since services are automatically restarted if possible, it may seem that a process refuses to die. If the service is defective, the service will be placed in maintenance mode, but normally a service is restarted if the process for the service is killed. The `svcadm` command should be used to disable any SMF service that should not be running.
- Many of the scripts in `/etc/init.d` and /etc/rc*.d have been removed. The scripts are no longer needed to enable or disable a service. Entries from `/etc/inittab` have also been removed, so that the services can be administered using SMF. Scripts and `inittab` entries that are provided by an ISV or are locally developed will continue to run. The services may not start at exactly the same point in the boot process, but they are not started before the SMF services, so that any service dependencies should be OK.

SMF Concepts

This section presents terms and their definitions within the SMF framework. These terms are used throughout the documentation. To grasp SMF concepts, an understanding of these terms is essential.

SMF Service

The fundamental unit of administration in the SMF framework is the *service instance*. Each SMF service has the potential to have multiple versions of it configured. As well, multiple instances of the same version can run on a single Solaris system. An *instance* is a specific configuration of a service. A web server is a service. A specific web server

daemon that is configured to listen on port 80 is an instance. Each instance of the web server service could have different configuration requirements. The service has system-wide configuration requirements, but each instance can override specific requirements, as needed. Multiple instances of a single service are managed as child objects of the service object.

Services are not just the representation for standard long-running system services such as `in.dhcpd` or `nfsd`. Services also represent varied system entities that include ISV applications such as Oracle software. In addition, a service can include less traditional entities such as the following:

- A physical network device
- A configured IP address
- Kernel configuration information
- Milestones that correspond to system init state, such as the multiuser run level

Generically, a service is an entity that provides a list of capabilities to applications and other services, local and remote. A service is dependent on an implicitly declared list of local services.

A *milestone* is a special type of service. Milestone services represent high-level attributes of the system. For example, the services which constitute run levels S, 2, and 3 are each represented by milestone services.

Service Identifiers

Each service instance is named with a Fault Management Resource Identifier or FMRI. The FMRI includes the service name and the instance name. For example, the FMRI for the `rlogin` service is `svc:/network/login:rlogin`, where `network/login` identifies the service and `rlogin` identifies the service instance.

Equivalent formats for an FMRI are as follows:

- `svc://localhost/system/system-log:default`
- `svc:/system/system-log:default`
- `system/system-log:default`

In addition, some SMF commands can use the following FMRI format: `svc:/system/system-log`. Some commands infer what instance to use, when there is no ambiguity. See the SMF command man pages, such as `svcadm`(1M) or `svcs`(1), for instructions about which FMRI formats are appropriate.

The service names usually include a general functional category. The categories include the following:

- `application`
- `device`
- `milestone`

- network
- platform
- site
- system

Legacy init.d scripts are also represented with FMRIs that start with lrc instead of svc, for example: lrc:/etc/rcS_d/S35cacheos_sh. The legacy services can be monitored using SMF. However, you cannot administer these services.

When booting a system for the first time with SMF, services listed in /etc/inetd.conf are automatically converted into SMF services. The FMRIs for these services are slightly different. The syntax for a converted inetd services is:

network/<*service-name*>/<*protocol*>

In addition, the syntax for a converted service that uses the RPC protocol is:

network/rpc-<*service-name*>/rcp_<*protocol*>

Where <*service-name*> is the name defined in /etc/inetd.conf and <*protocol*> is the protocol for the service. For instance, the FMRI for the rpc.cmsd service is network/rpc-100068_2-5/rpc_udp.

Service States

The svcs command displays the state, start time, and FMRI of service instances. The state of each service is one of the following:

- degraded – The service instance is enabled, but is running at a limited capacity.
- disabled – The service instance is not enabled and is not running.
- legacy_run – The legacy service is not managed by SMF, but the service can be observed. This state is only used by legacy services.
- maintenance – The service instance has encountered an error that must be resolved by the administrator.
- offline – The service instance is enabled, but the service is not yet running or available to run.
- online – The service instance is enabled and has successfully started.
- uninitialized – This state is the initial state for all services before their configuration has been read.

SMF Manifests

An SMF *manifest* is an XML file that contains a complete set of properties that are associated with a service or a service instance. The files are stored in /var/svc/manifest. Manifests should not be used to modify the properties of a

service. The service configuration repository is the authoritative source of configuration information. To incorporate information from the manifest into the repository, you must either run `svccfg import` or allow the service to import the information during a system boot.

See the `service_bundle`(4) man page for a complete description of the contents of the SMF manifests.

SMF Profiles

An SMF *profile* is an XML file that lists the set of service instances that are enabled when a system is booted. The profiles are stored in `/var/svc/profile`. These are some the profiles that are included:

- `generic_open.xml` — This profile enables most of the standard internet services that have been enabled by default in earlier Solaris releases. This is the default profile.
- `generic_limited_net.xml` — This profile disables many of the standard internet services. The `sshd` service and the NFS services are started, but most of the rest of the internet services are disabled.

For more information about using profiles, see "How to Use a Different SMF Profile" on page 237.

Service Configuration Repository

The *service configuration repository* stores persistent configuration information as well as SMF runtime data for services. The repository is distributed among local memory and local files. SMF is designed so that eventually, service data can be represented in the network directory service. The network directory service is not yet available. The data in the service configuration repository allows for the sharing of configuration information and administrative simplicity across many Solaris instances. The service configuration repository can only be manipulated or queried using SMF interfaces. For more information about manipulating and accessing the repository, see the `svccfg`(1M) and `svcprop`(1) man pages. The service configuration repository daemon is covered in the `svc.configd`(1M) man page. The service configuration library is documented in the `libscf`(3LIB) man page.

SMF Snapshots

The data in the service configuration repository includes *snapshots*, as well as a configuration that can be edited. Data about each service instance is stored in the snapshots. The standard snapshots are as follows:

- `initial` – Taken on the first import of the manifest
- `running` – Used when the service methods are executed
- `start` – Taken at the last successful start

The SMF service always executes with the `running` snapshot. This snapshot is automatically created if it does not exist.

The `svcadm refresh` command, sometimes followed by the `svcadm restart` command, makes a snapshot active. The `svccfg` command is used to view or revert to instance configurations in a previous snapshot.

SMF Administrative and Programming Interfaces

This section introduces the interfaces that are available when you use SMF.

SMF Command-Line Administrative Utilities

SMF provides a set of command-line utilities that interact with SMF and accomplish standard administrative tasks. The following utilities can be used to administer SMF.

TABLE 9–1 Service Management Facility Utilities

Command Name	Function
`inetadm`	Provides the ability to observe or configure services controlled by `inetd`
`svcadm`	Provides the ability to perform common service management tasks, such as enabling, disabling, or restarting service instances
`svccfg`	Provides the ability to display and manipulate the contents of the service configuration repository
`svcprop`	Retrieves property values from the service configuration repository with a output format appropriate for use in shell scripts
`svcs`	Gives detailed views of the service state of all service instances in the service configuration repository

Service Management Configuration Library Interfaces

SMF provides a set of programming interfaces that are used to interact with the service configuration repository through the `svc.configd` daemon. This daemon is the arbiter of all requests to the local and remote repository datastores. A set of fundamental interfaces is defined as the lowest level of interaction possible with services in the service configuration repository. The interfaces provide access to all service configuration repository features such as transactions and snapshots.

Many developers only need a set of common tasks to interact with SMF. These tasks are implemented as convenience functions on top of the fundamental services to ease the implementation burden.

SMF Components

SMF includes a master restarter daemon and delegated restarters.

SMF Master Restarter Daemon

The `svc.startd` daemon is the master process starter and restarter for the Solaris OS. The daemon is responsible for managing service dependencies for the entire system. The daemon takes on the previous responsibility that `init` held of starting the appropriate `/etc/rc*.d` scripts at the appropriate run levels. First, `svc.startd` retrieves the information in the service configuration repository. Next, the daemon starts services when their dependencies are met. The daemon is also responsible for restarting services that have failed and for shutting down services whose dependencies are no longer satisfied. The daemon keeps track of service state through an operating system view of availability through events such as process death.

SMF Delegated Restarters

Some services have a set of common behaviors on startup. To provide commonality among these services, a delegated restarter might take responsibility for these services. In addition, a delegated restarter can be used to provide more complex or application-specific restarting behavior. The delegated restarter can support a different set of methods, but exports the same service states as the master restarter. The restarter's name is stored with the service. A current example of a delegated restarter is `inetd`, which can start Internet services on demand, rather than having the services always running.

SMF and Booting

SMF provides new methods for booting a system. For instance:

- There is a additional system state which is associated with the `all` milestone. This milestone is different than the multiuser `init` state because SMF only knows about the services that are defined. If you have added services, such as third party products, they may not be started automatically unless you use the following command:

  ```
  # boot -m milestone=all
  ```

- If you boot a system using one of the milestones, it is important to use the `-s` option as well. If you do not include the `-s`, then the system will stay in the milestone state that you booted the system in. The system will not go into multiuser state automatically by typing Control-D. You can get into the multiuser state by using the following command:

  ```
  # svcadm milestone -t all
  ```

- When booting a system, you can choose to use the verbose option to see more messages. By default, the system will not display these messages. To boot in the verbose mode, use the following command:

  ```
  # boot -mverbose
  ```

SMF Compatibility

While many standard Solaris services are now managed by SMF, the scripts placed in `/etc/rc*.d` continue to be executed on run-level transitions. Most of the `/etc/rc*.d` scripts that were included in previous Solaris releases have been removed as part of SMF. The ability to continue to run the remaining scripts allows for third-party applications to be added without having to convert the services to use SMF.

In addition, `/etc/inittab` and `/etc/inetd.conf` must be available for packages to amend with postinstall scripts. These are called legacy-run services. The `inetconv` command is run to add these legacy-run services to the service configuration repository. The status of these services can be viewed, but no other changes are supported through SMF. Applications that use this feature will not benefit from the precise fault containment provided by SMF.

Applications converted to utilize SMF should no longer make modifications to the `/etc/inittab` and `/etc/inetd.conf` files. The converted applications will not use the `/etc/rc*.d` scripts. Also, the new version of `inetd` does not look for entries in `/etc/inetd.conf`.

Run Levels

A system's *run level* (also known as an *init state*) defines what services and resources are available to users. A system can be in only one run level at a time.

The Solaris OS has eight run levels, which are described in the following table. The default run level is specified in the `/etc/inittab` file as run level 3.

TABLE 9–2 Solaris Run Levels

Run Level	Init State	Type	Purpose
0	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system.
s or S	Single-user state	Single-user	To run as a single user with some file systems mounted and accessible.
1	Administrative state	Single-user	To access all available file systems. User logins are disabled.
2	Multiuser state	Multiuser	For normal operations. Multiple users can access the system and all file system. All daemons are running except for the NFS server daemons.
3	Multiuser level with NFS resources shared	Multiuser	For normal operations with NFS resources shared. This is the default run level for the Solaris OS.
4	Alternative multiuser state		Not configured by default, but available for customer use.
5	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system. If possible, automatically turns off power on systems that support this feature.
6	Reboot state	Reboot	To shut down the system to run level 0, and then reboot to multiuser level with NFS resources shared (or whatever level is the default in the `inittab` file).

In addition, the `svcadm` command can be used to change the run level of a system, by selecting a milestone at which to run. The following table shows which run level corresponds to each milestone.

TABLE 9–3 Solaris Run Levels and SMF Milestones

Run Level	SMF Milestone FMRI
S	`milestone/single-user:default`
2	`milestone/multi-user:default`
3	`milestone/multi-user-server:default`

Determining a System's Run Level

Display run level information by using the `who -r` command.

```
$ who -r
```

Use the `who -r` command to determine a system's current run level for any level.

EXAMPLE 9–1 Determining a System's Run Level

This example displays information about a system's current run level and previous run levels.

```
$ who -r
 .    run-level 3  Dec 13 10:10  3  0 S
$
```

Output of `who -r` command	Description
`run-level 3`	Identifies the current run level
`Dec 13 10:10`	Identifies the date of last run level change
`3`	Also identifies the current run level
`0`	Identifies the number of times the system has been at this run level since the last reboot
`S`	Identifies the previous run level

`/etc/inittab` File

When you boot the system or change run levels with the `init` or `shutdown` command, the `init` daemon starts processes by reading information from the `/etc/inittab` file. This file defines these important items for the `init` process:

- That the `init` process will restart

- What processes to start, monitor, and restart if they terminate
- What actions to take when the system enters a new run level

Each entry in the `/etc/inittab` file has the following fields:

id : *rstate* : *action* : *process*

The following table describes the fields in an `inittab` entry.

TABLE 9–4 Fields Descriptions for the `inittab` File

Field	Description
id	Is a unique identifier for the entry.
rstate	Lists the run levels to which this entry applies.
action	Identifies how the process that is specified in the process field is to be run. Possible values include: `sysinit`, `boot`, `bootwait`, `wait`, and `respawn`. For a description of the other action keywords, see `inittab`(4).
process	Defines the command or script to execute.

EXAMPLE 9–2 Default `inittab` File

The following example shows a default `inittab` file that is installed with the Solaris release. A description for each line of output in this example follows.

```
ap::sysinit:/sbin/autopush -f /etc/iu.ap                                    (1)
sp::sysinit:/sbin/soconfig -f /etc/sock2path                                (2)
smf::sysinit:/lib/svc/bin/svc.startd    >/dev/msglog 2<>/dev/msglog         (3)
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/...   (4)
```

1. Initializes STREAMS modules
2. Configures socket transport providers
3. Initializes the master restarter for SMF
4. Describes a power fail shutdown

What Happens When the System Is Brought to Run Level 3

1. The `init` process is started and reads the `/etc/default/init` file to set any environment variables. By default, only the `TIMEZONE` variable is set.
2. Then, `init` reads the `inittab` file and does the following:
   a. Executes any process entries that have `sysinit` in the `action` field so that any special initializations can take place before users login.
   b. Passes the startup activities to `svc.startd`.

For a detailed description of how the `init` process uses the `inittab` file, see `init`(1M).

The following table describes the keywords used for run level 3's `action` field.

TABLE 9–5 Run Level 3 Action Keyword Descriptions

Key Word	Description
`powerfail`	Starts the process when the `init` process receives a power failure signal
`respawn`	Starts the process and restarts it when it dies
`wait`	Starts the process and waits for it to finish before going on to the next entry for this run level

The following table describes the processes (or commands) that are executed at run level 3.

TABLE 9–6 Command Descriptions for Run Level 3

Command or Script Name	Description
`/usr/sbin/shutdown`	Shuts down the system. The `init` process runs the `shutdown` command only if the system has received a power fail signal.
`/sbin/rcS`	Checks and mounts root (`/`), `/usr`, `/tmp`, `/var`, `/var/adm`, and `/var/run` file systems.
`/sbin/rc2`	Starts the standard system processes and brings the system up into run level 2 (multiuser level).
`/sbin/rc3`	Starts NFS resource sharing for run level 3.

Run Control Scripts

The Solaris software provides a detailed series of run control (`rc`) scripts to control run-level changes. Each run level has an associated `rc` script that is located in the `/sbin` directory:

- `rc0`
- `rc1`
- `rc2`
- `rc3`
- `rc5`
- `rc6`

- `rcS`

For each `rc` script in the `/sbin` directory, there is a corresponding directory named `/etc/rc`*n*`.d` that contains scripts to perform various actions for that run level. For example, `/etc/rc2.d` contains files that are used to start and stop processes for run level 2.

```
# ls /etc/rc2.d
K03samba                S42ncakmod              S81dodatadm.udaplt
K05volmgt               S47pppd                 S89PRESERVE
K06mipagent             S65ipfboot              S89bdconfig
K07dmi                  S69mrouted              S90wbem
K07snmpdx               S70sckm                 S93cacheos.finish
K16apache               S70uucp                 S94ncalogd
K27boot.server          S72autoinstall          S95ncad
README                  S73cachefs.daemon       S95networker
S10lu                   S75savecore             S98deallocate
S20sysetup              S80lp                   S99audit
S40llc2                 S80spc
```

The `/etc/rc`*n*`.d` scripts are always run in ASCII sort order. The scripts have names of the form:

```
[KS][0-9][0-9]*
```

Files that begin with `K` are run to terminate (kill) a system service. Files that begin with `S` are run to start a system service.

Run control scripts are located in the `/etc/init.d` directory. These files are linked to corresponding run control scripts in the `/etc/rc`*n*`.d` directories.

The actions of each run control script are summarized in the following section.

Run Control Script Summaries

The following sections summarize the run control scripts that are used to start and stop system services when you change run levels.

The `/sbin/rc0` Script

The `/sbin/rc0` script runs the `/etc/rc0.d` scripts to perform the following tasks:

- Stops system services and daemons
- Terminates all running processes
- Unmounts all file systems

The `/sbin/rc1` Script

The `/sbin/rc1` script runs the `/etc/rc1.d` scripts to perform the following tasks:

- Stops system services and daemons

- Terminates all running user processes
- Unmounts all remote file systems
- Mounts all local file systems if the previous run level was S

The /sbin/rc2 Script

The /sbin/rc2 script runs the /etc/rc2.d scripts to perform the following tasks, grouped by function:

Local system-related tasks:

- Starts system accounting and system auditing, if configured
- Sets the default scheduling class if the /etc/dispadmin.conf file exists
- Configures serial device stream
- Configures WBEM services

Network service or security-related tasks:

- Starts the logical link controller (llc2), if configured
- Configures the Solaris Network Cache and Accelerator (NCA) and NCA logging, if appropriate
- Starts the Solaris PPP server or client daemons (pppoed or pppd), if configured
- Starts directory server (slapd) daemon, if configured
- Configures system resource controls and system pools if the /etc/rctladm.conf and /etc/pooladm.conf files exist
- Starts the htt_server process

Install-related tasks:

- Configures the boot environment for the Live Upgrade software upon system startup or system shutdown
- Checks for the presence of the /etc/.UNCONFIGURE file to see if the system should be reconfigured
- Reboots the system from the installation media or a boot server if either /.PREINSTALL or /AUTOINSTALL exists

Hardware-related tasks:

- Starts the Sun Fire™ 15000 key management daemon (sckmd), if appropriate
- Runs the flash PROM update script
- Configures any graphic frame buffers or graphic accelerators

Transitions the following services between run level changes:

- Apache (tomcat)
- Mobile IP (mipagent)
- Samba (smdb) and (nmdb)
- Solstice Enterprise Agents™ daemon (dmispd) and (snmpXdmid)

Note – Many of the system services and applications that are started at run level 2 depend on what software is installed on the system.

The /sbin/rc3 Script

The /sbin/rc3 script runs the /etc/rc3.d scripts to perform the following tasks:

- Starts the Apache server daemon (tomcat), if configured
- Starts Mobile IP daemon (mipagent), if configured
- Starts the Samba daemons (smdb and nmdb), if configured
- Starts the Solstice Enterprise Agents daemons (dmispd and snmpXdmid)

The /sbin/rc5 and /sbin/rc6 Scripts

The /sbin/rc5 and /sbin/rc6 scripts run the /etc/rc0.d/K* scripts to perform the following tasks:

- Kills all active processes
- Unmounts the file systems

The /sbin/rcS Script

The /sbin/rcS script runs the /etc/rcS.d scripts to bring the system up to run level S. The following tasks are performed by these scripts:

- Starts wrsmconf to manage WCI RSM controller configurations

CHAPTER 10

Shutting Down a System (Tasks)

This chapter describes the procedures for shutting down systems. This is a list of the step-by-step instructions in this chapter.

This is a list of the overview information in this chapter.

For overview information about system run levels, see Chapter 9.

For information on the procedures associated with run levels and boot files, see "Shutting Down the System (Task Map)" on page 171.

Shutting Down the System (Task Map)

Task	Description	For Instructions
Determine who is logged in to a system.	Use the `who` command to determine who is logged in to a system.	"How to Determine Who Is Logged in to a System" on page 174
Shut down a server.	Use the `shutdown`command with the appropriate options to shut down a server.	"How to Shut Down a Server" on page 174
Shut down a stand-alone system.	Use the `init` command and indicate the appropriate run-level to shut down a stand-alone system.	"How to Shut Down a Stand-Alone System" on page 177

Task	Description	For Instructions
Turn off power to all devices.	Powering down a system includes the following devices: ■ CPU ■ Monitor ■ External devices, such as disks, tapes, and printers.	"How to Turn Off Power to All Devices" on page 179

Shutting Down the System

Solaris software is designed to run continuously so that the electronic mail and network software can work correctly. However, some system administration tasks and emergency situations require that the system is shut down to a level where it is safe to remove power. In some cases, the system needs to be brought to an intermediate level, where not all system services are available. Such cases include the following:

- Adding or removing hardware
- Preparing for an expected power outage
- Performing file system maintenance, such as a backup

For a complete list of system administration tasks that require a system shutdown, see Chapter 8.

For information on using your system's power management features, see the `pmconfig`(1M) man page.

System Shutdown Commands

The use of the `init` and `shutdown` commands are the primary ways to shut down a system. Both commands perform a *clean shutdown* of the system. As such, all file system changes are written to the disk, and all system services, processes, and the operating system are terminated normally.

The use of a system's Stop key sequence or turning a system off and then on are not clean shutdowns because system services are terminated abruptly. However, sometimes these actions are needed in emergency situations. For instructions on system recovery techniques, see Chapter 11 or Chapter 12.

The following table describes the various shutdown commands and provides recommendations for using them.

TABLE 10–1 Shutdown Commands

Command	Description	When To Use
`shutdown`	An executable shell script that calls the `init` program to shut down the system. The system is brought to run level S by default.	Recommended for servers operating at run level 3 because users are notified of the impending shutdown. Also notified are the systems that are mounting resources from the server that is being shut down.
`init`	An executable that kills all active processes and synchronizes the disks before changing run levels.	Recommended for stand-alone systems when other users will not be affected. Provides a faster system shutdown because users are not notified of the impending shutdown.
`reboot`	An executable that synchronizes the disks and passes boot instructions to the `uadmin` system call. In turn, this system call stops the processor.	The `init` command is the preferred method.
`halt`, `poweroff`	An executable that synchronizes the disks and stops the processor.	Not recommended because it doesn't shutdown all processes, and unmount any remaining file systems. Stopping the services, without doing a clean shutdown, should only be done in an emergency or if most of the services are already stopped.

User Notification of System Down Time

When the `shutdown` command is initiated, a warning followed by a final shutdown message is broadcast to all users who are currently logged in to the system and all systems that are mounting resources from the affected system.

For this reason, the `shutdown` command is preferred instead of the `init` command when you need to shut down a server. When you use either command, you might want to give users more notice by sending them a mail message about any scheduled system shutdown.

Use the `who` command to determine which users on the system need to be notified. This command is also useful for determining a system's current run level. For more information, see "Determining a System's Run Level" on page 165 and the `who`(1) man page.

▼ How to Determine Who Is Logged in to a System

Steps

1. **Log into the system to be shut down.**
2. **Display all users who are logged in to the system.**

   ```
   $ who
   ```

Example 10–1 Determining Who Is Logged in to a System

The following example shows how to display who is logged in to the system.

```
$ who
holly       console      May  7 07:30
kryten      pts/0        May  7 07:35   (starlite)
lister      pts/1        May  7 07:40   (bluemidget)
```

- Data in the first column identifies the user name of the logged-in user
- Data in the second column identifies the terminal line of the logged-in user
- Data in the third column identifies the date and time that the user logged in
- Data in the forth column, if present, identifies the host name if a user is logged in from a remote system

▼ How to Shut Down a Server

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Find out if users are logged in to the system.**

   ```
   # who
   ```

   A list of all logged-in users is displayed. You might want to send mail or broadcast a message to let users know that the system is being shut down.

3. **Shut down the system.**

   ```
   # shutdown -iinit-level -ggrace-period -y
   ```

   -iinit-level Brings the system to an init level that is different from the default of S. The choices are 0, 1, 2, 5, and 6.

   Run levels 0 and 5 are reserved states for shutting the system down. Run level 6 reboots the system. Run level 2 is available as a multi-user operating state.

`-g`*grace-period*	Indicates a time (in seconds) before the system is shut down. The default is 60 seconds.
`-y`	Continues to shut down the system without intervention. Otherwise, you are prompted to continue the shutdown process after 60 seconds.

For more information, see the `shutdown`(1M) man page.

4. **If you are asked for confirmation, type `y`.**

```
Do you want to continue? (y or n): y
```

If you used the `shutdown -y` command, you will not be prompted to continue.

5. **Type the superuser password, if prompted.**

```
Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxxxxx
```

6. **After you have finished the system administration tasks, press Control-D to return to the default system run level.**

7. **Use the following table to verify that the system is at the run level that you specified in the `shutdown` command.**

Specified Run Level	SPARC Based System Prompt	x86 Based System Prompt
S (single-user level)	#	#
0 (power-down level)	`ok` or `>`	`Press any key to reboot`
Run level 3 (multiuser level with remote resources shared)	*hostname* `console login:`	*hostname* `console login:`

Example 10–2 SPARC: Bringing a Server to Run Level S

In the following example, the `shutdown` command is used to bring a SPARC based system to run level S (single-user level) in three minutes.

```
# who
root   console      Jun 14 15:49    (:0)

# shutdown -g180 -y

Shutdown started.    Mon Jun 14 15:46:16 MDT 2004

Broadcast Message from root (pts/4) on venus Mon Jun 14 15:46:16...
The system venus will be shut down in 3 minutes .
.
.
Broadcast Message from root (pts/4) on venus Mon Jun 14 15:46:16...
The system venus will be shut down in 30 seconds .
```

```
.
.
INIT: New run level: S
The system is coming down for administration.  Please wait.
Unmounting remote filesystems: /vol nfs done.
Shutting down Solaris Management Console server on port 898.
Print services stopped.
Jun 14 15:49:00 venus syslogd: going down on signal 15
Killing user processes: done.

Requesting System Maintenance Mode
SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
#
```

Example 10–3 SPARC: Bringing a Server to Run Level 0

In the following example, the `shutdown` command is used to bring a SPARC based system to run level 0 in 5 minutes without requiring additional confirmation.

```
# who
root       console       Jun 17 12:39
userabc          pts/4         Jun 17 12:39   (:0.0)
# shutdown -i0 -g300 -y
Shutdown started.    Thu Jun 17 12:40:25 MST 2004

Broadcast Message from root (console) on pretend Thu Jun 17 12:40:25...
The system pretend will be shut down in 5 minutes
.
.
.
Changing to init state 0 - please wait
#
INIT: New run level: 0
The system is coming down.  Please wait.
System services are now being stopped.
.
.
.
The system is down.
syncing file systems... done
Program terminated
Type  help  for more information
ok
```

If you are bringing the system to run level 0 to turn off power to all devices, see "How to Turn Off Power to All Devices" on page 179.

Example 10–4 SPARC: Rebooting a Server to Run Level 3

In the following example, the `shutdown` command is used to reboot a SPARC based system to run level 3 in two minutes. No additional confirmation is required.

```
# who
root           console      Jun 14 15:49    (:0)
userabc    pts/4        Jun 14 15:46    (:0.0)
# shutdown -i6 -g120 -y
Shutdown started.    Mon Jun 14 15:46:16 MDT 2004

Broadcast Message from root (pts/4) on venus Mon Jun 14 15:46:16...
The system venus will be shut down in 2 minutes


Changing to init state 6 - please wait
#
INIT: New run level: 6
The system is coming down.  Please wait.
.
.
.
The system is down.
syncing file systems... done
rebooting...
.
.
.
venus console login:
```

See Also Regardless of why you shut down a system, you'll probably want to return to run level 3 where all file resources are available and users can log in. For instructions on bringing a system back to a multiuser level, see Chapter 11 or Chapter 12.

▼ How to Shut Down a Stand-Alone System

Use this procedure when you need to shut down a stand-alone system.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Shut down the system.**

   # **init** *run-level*

   where *run-level* identifies the new run level.

   For more information, see the `init`(1M) man page.

3. **Use the following table to verify that the system is at the run level that you specified in the `init` command.**

Specified Run Level	SPARC Based System Prompt	x86 Based System Prompt
S (single-user level)	#	#
2 (multiuser level)	#	#
0 (power-down level)	`ok` or `>`	`Press any key to reboot`
3 (multiuser level with NFS resources shared)	*hostname* `console login:`	*hostname* `console login:`

Example 10–5 Bringing a Stand-Alone System to Run Level 0

In this example, the `init` command is used to bring an x86 based stand-alone system to the level where it is safe to turn off power.

```
# init 0
#
INIT: New run level: 0
The system is coming down.  Please wait.
.

.

.
The system is down.
syncing file systems... [11] [10] [3] done
Press any key to reboot
```

If you are bringing the system to run level 0 to turn off power to all devices, see "How to Turn Off Power to All Devices" on page 179.

Example 10–6 SPARC: Bringing a Stand-Alone System to Run Level S

In this example, the `init` command is used to bring a SPARC based stand-alone system to run level S (single-user level).

```
# init s
#
INIT: New run level: S
The system is coming down for administration.  Please wait.
Unmounting remote filesystems: /vol nfs done.
Print services stopped.
syslogd: going down on signal 15
Killing user processes: done.

SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
#
```

See Also Regardless of why you shut down the system, you'll probably want to return to run level 3 where all file resources are available and users can log in. For instructions on bringing a system back to a multiuser level, see Chapter 11 or Chapter 12.

Turning Off Power to All Devices

You need turn off power to all system devices when you do the following:

- Replace or add hardware.
- Move the system from one location to another.
- Prepare for an expected power outage or natural disaster such as an approaching electrical storm.

Turn the power off for system devices, including the CPU, the monitor, and external devices such as disks, tapes, and printers.

Before you turn off power to all system devices, you should shut down the system cleanly, as described in the preceding sections.

▼ How to Turn Off Power to All Devices

Steps

1. **Select one of the following methods to shut down the system:**
   - If you are shutting down a server, see "How to Shut Down a Server" on page 174.
   - If you are shutting down a stand-alone system, see "How to Shut Down a Stand-Alone System" on page 177.
2. **Turn off the power to all devices after the system is shutdown. If necessary, also unplug the power cables.**
3. **After power can be restored, use the following steps to turn on the system and devices.**
   a. **Plug in the power cables.**
   b. **Turn on the monitor.**
   c. **Turn on disk drives, tape drives, and printers.**
   d. **Turn on the CPU.**
      The system is brought to run level 3.

CHAPTER 11

SPARC: Booting a System (Tasks)

This chapter describes the procedures for using the OpenBoot™ PROM monitor and the procedures for booting a SPARC based system to different run levels.

For information on the procedures associated with booting a SPARC based system, see "SPARC: Booting a System (Task Map)" on page 182.

For overview information about the boot process, see Chapter 8. To troubleshoot boot problems, see "What to Do If Rebooting Fails" in *System Administration Guide: Advanced Administration*.

For step-by-step instructions on booting an x86 based system, see Chapter 12.

SPARC: Booting a System (Task Map)

Task	Description	For Instructions
Use the Boot PROM.	The boot PROM is used to boot a system. You might need to change the way the system boots. For example, you might want to reset the device to boot from or run hardware diagnostics before you bring the system to a multiuser level. Associated tasks include the following: ■ Identify the PROM revision number. ■ Identify devices on the system to boot from. ■ Change the default boot device when a new disk is added or when you need to change the system boot method.	"SPARC: How to Find the PROM Revision Number for a System" on page 184 "SPARC: How to Identify Devices on a System" on page 184 "SPARC: How to Change the Default Boot Device" on page 186
Reset the system.	When you reset the system, the system runs diagnostic tests on the hardware, then reboots.	"SPARC: How to Reset the System" on page 187
Boot a system.	Select one of the following boot methods: ■ Boot to run level 3 – Used after shutting down the system or performing a system hardware maintenance task. ■ Boot to run level S – Used after performing a system maintenance task such as backing up a file system. At this level, only local file systems are mounted and users cannot log in to the system. ■ Boot interactively – Used after making temporary changes to a system file or the kernel for testing purposes. ■ Boot from the network – Used to boot a system from the network. This method is used for booting a diskless client.	"SPARC: How to Boot a System to Run Level 3 (Multiuser Level)" on page 188 "SPARC: How to Boot a System to Run Level S (Single-User Level)" on page 189 "SPARC: How to Boot a System Interactively" on page 190 "SPARC: How to Boot a System From the Network" on page 191

Task	Description	For Instructions
Boot a system for recovery purposes.	Boot for recovery purposes – Used to boot the system when a damaged file or file system is preventing the system from booting. You might need to do one or both of the following to boot for recovery purposes: ■ Stop the system to attempt recovery. ■ Boot to repair an important system file that is preventing the system from booting successfully.	"SPARC: How to Stop the System for Recovery Purposes" on page 193 "SPARC: How to Boot a System for Recovery Purposes" on page 193
Force a crash dump and reboot a system.	Force a crash dump and reboot the system - Used to force a crash dump for troubleshooting purposes.	"SPARC: How to Force a Crash Dump and Reboot of the System" on page 196
Troubleshoot problems with the `kmdb` command.	Boot kmdb – Used to troubleshoot system problems.	"SPARC: How to Boot the System With the Kernel Debugger (`kmdb`)" on page 197 Use the `halt` command with the `-d` option if you do not have time to debug the system interactively. Running the `halt` command with the `-d` option requires a manual reboot afterwards. Whereas, if you use the `reboot` command, the system will reboot automatically.

SPARC: Using the Boot PROM

System administrators typically use the PROM level to boot a system. You need to change the default boot device to do the following:

- Add a new drive to the system either permanently or temporarily
- Change the network boot strategy
- Temporarily boot a stand-alone system from the network

For a complete list of PROM commands, see `monitor`(1M) or `eeprom`(1M).

▼ SPARC: How to Find the PROM Revision Number for a System

Step

- **Display a system's PROM revision number by using the `banner` command.**

```
ok banner
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.
```

Hardware configuration information, including the revision number of the PROM, is displayed. In this example, the PROM revision number is 3.15.

▼ SPARC: How to Identify Devices on a System

You might need to identify the devices on the system to determine what are the appropriate devices to boot from.

Before You Begin

Before you can safely use the `probe` commands to determine what devices are attached to the system, you need to do the following:

- Change the PROM `auto-boot?` parameter to false.

```
ok setenv auto-boot? false
```

- Issue the `reset-all` command to clear system registers.

```
ok reset-all
```

You can view the `probe` commands that are available on your system by using the `sifting probe` command:

```
ok sifting probe
```

If you run the `probe` commands without clearing the system registers, the following message is displayed:

```
ok probe-scsi
This command may hang the system if a Stop-A or halt command
has been executed.  Please type reset-all to reset the system
before executing this command.
Do you wish to continue? (y/n) n
```

Steps

1. **Identify the devices on the system.**

```
ok probe-device
```

2. **(Optional) If you want the system to reboot after a power failure or after using the `reset` command, then reset the `auto-boot?` parameter to true.**

```
ok setenv auto-boot? true
auto-boot? =          true
```

3. **Boot the system back to multiuser mode.**

```
ok reset
```

Example 11–1 SPARC: Identifying the Devices on a System

The following example shows how to identify the devices connected to an Ultra™ 10 system.

```
ok setenv auto-boot? false
auto-boot? =          false
ok reset-all
Resetting ...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #10933339.
Ethernet address 8:0:20:a6:d4:5b, Host ID: 80a6d45b.

ok probe-ide
  Device 0  ( Primary Master )
          ATA Model: ST34321A

  Device 1  ( Primary Slave )
         Not Present

  Device 2  ( Secondary Master )
         Removable ATAPI Model: CRD-8322B

  Device 3  ( Secondary Slave )
         Not Present

ok setenv auto-boot? true
auto-boot? =          true
```

Alternatively, you can use the `devalias` command to identify the device aliases and the associated paths of devices that *might* be connected to the system. For example:

```
ok devalias
screen                   /pci@1f,0/pci@1,1/SUNW,m64B@2
net                      /pci@1f,0/pci@1,1/network@1,1
cdrom                    /pci@1f,0/pci@1,1/ide@3/cdrom@2,0:f
disk                     /pci@1f,0/pci@1,1/ide@3/disk@0,0
disk3                    /pci@1f,0/pci@1,1/ide@3/disk@3,0
disk2                    /pci@1f,0/pci@1,1/ide@3/disk@2,0
disk1                    /pci@1f,0/pci@1,1/ide@3/disk@1,0
disk0                    /pci@1f,0/pci@1,1/ide@3/disk@0,0
ide                      /pci@1f,0/pci@1,1/ide@3
floppy                   /pci@1f,0/pci@1,1/ebus@1/fdthree
ttyb                     /pci@1f,0/pci@1,1/ebus@1/se:b
ttya                     /pci@1f,0/pci@1,1/ebus@1/se:a
keyboard!                /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8:forcemode
keyboard                 /pci@1f,0/pci@1,1/ebus@1/su@14,3083f8
mouse                    /pci@1f,0/pci@1,1/ebus@1/su@14,3062f8
name                     aliases
```

▼ SPARC: How to Change the Default Boot Device

You might need to identify the devices on the system before you can change the default boot device to some other device. For information on identifying devices on the system, see "SPARC: How to Identify Devices on a System" on page 184.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Change to run level 0.**

   ```
   # init 0
   ```

   The `ok` PROM prompt is displayed. For more information, see the`init`(1M) man page.

3. **Change the value of the `boot-device` parameter.**

   `ok` **`setenv boot-device`** *device*[*n*]

`boot-device`	Identifies the parameter for setting the device from which to boot.
device[*n*]	Identifies the `boot-device` value such as a disk or the network. The *n* can be specified as the *disk number*.

   Use one of the `probe` commands if you need help identifying the disk number.

4. **Verify that the default boot device has been changed.**

   ```
   ok printenv boot-device
   ```

5. **Save the new `boot-device` value.**

   ```
   ok reset
   ```

   The new `boot-device` value is written to the PROM.

Example 11–2 SPARC: Changing the Default Boot Device

In this example, the default boot device is set to disk.

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device disk
```

```
boot-device =         disk
ok printenv boot-device
boot-device           disk                    disk
ok reset
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.

Boot device: disk  File and args:
SunOS Release 5.9 Version 64-bit
.
.
.
pluto console login:
```

In this example, the default boot device is set to the network.

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device net
boot-device =         net
ok printenv boot-device
boot-device           net                     disk
ok reset
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.



Boot device: net  File and args:
.
.
.
pluto console login:
```

SPARC: How to Reset the System

Run the `reset` command from the `ok` prompt.

```
ok reset
```

This self-test program, which runs diagnostic tests on the hardware, is executed. Then, the system is rebooted.

SPARC: Booting a System

If a system is turned off, turning it on starts the multiuser boot sequence. The following procedures show how to boot to different run levels from the `ok` PROM prompt. These procedures assume that the system has been cleanly shut down, unless stated otherwise.

Use the `who -r` command to verify that the system is brought to the specified run level. For a description of run levels, see Chapter 9.

▼ SPARC: How to Boot a System to Run Level 3 (Multiuser Level)

Use this procedure to boot a system that is currently at run level 0 to run level 3.

Steps

1. **Boot the system to run level 3.**

   `ok` **`boot`**

   The automatic boot procedure displays a series of startup messages, and brings the system to run level 3. For more information, see the `boot`(1M) man page.

2. **Verify that the system has booted to run level 3.**

   The login prompt is displayed when the boot process has finished successfully.

   hostname `console login:`

Example 11–3 SPARC: Booting a System to Run Level 3 (Multiuser Level)

The following example displays the messages from booting a system to run level 3.

```
ok boot
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.

Rebooting with command: boot
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a  File and args: kernel/sparcv9/unix
SunOS Release 5.10 Version s10_60 64-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: hme0.
add net default: gateway 172.20.27.248
Hostname: starlite
The system is coming up.  Please wait.
```

```
NIS domain name is example.com
starting rpc services: rpcbind keyserv ypbind done.
Setting netmask of hme0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4: gateway starlite
syslog service starting.The system is ready.
Starting Sun(TM) Web Console Version 2.1-dev..
volume management starting.
The system is ready.
starlite console login:
```

In the preceding example, *sparcv9* was used as an example only. This string matches the output of the `isainfo- k` command.

▼ SPARC: How to Boot a System to Run Level S (Single-User Level)

Use this procedure to boot a system that is currently at run level 0 to run level S. This run level is used for system maintenance tasks, such as backing up a file system.

Steps

1. **Boot the system to run level S.**

   ```
   ok boot -s
   ```

2. **Type the superuser password when the following message is displayed:**

   ```
   SINGLE USER MODE

   Root password for system maintenance (control-d to bypass): xxxxxx
   ```

3. **Verify that the system is at run level S.**

   ```
   # who -r
   .       run-level S  Jun 10 15:27     3      0
   ```

4. **Perform the maintenance task that required the run level change to S.**

5. **After you complete the system maintenance task, type Control-D to bring the system to the multiuser state.**

Example 11–4 SPARC: Booting a System to Run Level S (Single-User Level)

The following example displays the messages from booting a system to run level S.

```
ok boot -s
.
.
.
Sun Microsystems Inc.   SunOS 5.10
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
```

```
Use is subject to license terms.
configuring IPv4 interfaces: hme0.
Hostname: starlite

SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode
Oct 14 15:01:28 su: 'su root' succeeded for root on /dev/console
Sun Microsystems Inc.   SunOS 5.10
# who -r
   .       run-level S  Sep 19 08:49     S      0  ?
    (Perform some maintenance task)
# ^D
```

▼ SPARC: How to Boot a System Interactively

Use this boot option when you need to specify an alternate kernel or `/etc/system` file.

Steps

1. **Boot the system interactively.**

   ```
   ok boot -a
   ```

2. **Answer the following system prompts:**

   a. **When prompted, enter the name of the kernel to use for booting.**

      Press enter to use the default kernel file name. Otherwise, provide the name of an alternate kernel, press Enter.

   b. **When prompted, provide an alternate path for the `modules` directories.**

      Press enter to use the default module directories. Otherwise, provide the alternate paths to module directories, press Enter.

   c. **When prompted, provide the name of an alternate system file.**

      Type `/dev/null` if your `/etc/system` file has been damaged.

   d. **When prompted, enter the `root` filesystem type.**

      Press enter to select UFS for local disk booting, which is the default, or enter `NFS` for network booting.

   e. **When prompted, enter the physical name of `root` device.**

      Provide an alternate device name or press return to use the default.

3. **If you are not prompted to answer these questions, verify that you typed the `boot -a` command correctly.**

Example 11–5 SPARC: Booting a System Interactively

In the following example, the default choices (shown in square brackets `[]`) are accepted.

```
ok boot -a
.
.
.
Rebooting with command: boot -a
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a
File and args: -a
Enter filename [kernel/sparcv9/unix]:      Press Return
Enter default directory for modules [/platform/SUNW,Ultra-5_10/kernel
     /platform/sun4u/kernel /kernel /usr/kernel]:      Press Return
Name of system file [etc/system]:      Press Return
SunOS Release 5.10 Version S10_60 64-bit
Copyright (c) 1983-2004 by Sun Microsystems, Inc. All rights reserved
Use is subject to license terms.
root filesystem type [ufs]:      Press Return
Enter physical name of root device
[/pci@1f,0/pci@1,1/ide@3/disk@0,0:a]:      Press Return
configuring IPv4 interfaces: hme0.
Hostname: starlite
The system is coming up.  Please wait.
checking ufs filesystems
.
.
.
The system is ready.
starlite console login:
```

▼ SPARC: How to Boot a System From the Network

Any system can boot from the network if a boot server is available. You might want to boot a stand-alone system from the network if the system cannot boot from the local disk. For information on changing or resetting the default boot device, see "SPARC: How to Change the Default Boot Device" on page 186.

Two network configuration boot strategies are available on sun–4u systems:

- RARP – Reverse Address Resolution Protocol and ONC+ RPC Bootparams Protocol
- DHCP – Dynamic Host Configuration Protocol

The default network boot strategy is set to RARP. You can use either protocol, depending on whether a RARP boot server or a DHCP boot server is available on your network.

Note – Sun Ultra systems must have PROM version 3.25.*nn* or later to use the DHCP network boot strategy. For information on determining your PROM version, see "SPARC: How to Find the PROM Revision Number for a System" on page 184.

If both protocols are available, you can temporarily specify which protocol to use in the `boot` command. Or, you can save the network boot strategy across system reboots at the PROM level by setting up an NVRAM alias. The following example uses the `nvalias` command to set up a network device alias for booting DHCP by default on a Sun Ultra 10 system.

```
ok nvalias net    /pci@1f,4000/network@1,1:dhcp
```

As a result, when you type `boot net`, the system boots by using the DHCP network book strategy.

Note – You should not use the `nvalias` command to modify the `NVRAMRC` file, unless you are very familiar with the syntax of this command and the `nvunalias` command. For information on using these commands, see the *OpenBoot 3.x Command Reference Manual*.

Before You Begin

You must have already set up a RARP or DHCP boot server in your network to use either protocol to boot successfully.

Steps

1. **If necessary, shut down the system.**
2. **Determine the method for booting from the network, and select one of the following:**
   a. **Boot the system from the network by using the DHCP strategy.**

      ```
      ok boot net[:dhcp]
      ```

      If you have changed the PROM setting to boot DHCP by default, as in the preceding `nvalias` example, you only have to specify `boot net`.

   b. **Boot the system from the network by using the RARP strategy.**

      ```
      ok boot net[:rarp]
      ```

      Because RARP is the default network boot strategy, you only have to specify `boot net:rarp` if you have changed the PROM value to boot DHCP.

▼ SPARC: How to Stop the System for Recovery Purposes

Steps

1. **Type the Stop key sequence for your system.**

   The monitor displays the `ok` PROM prompt.

   ```
   ok
   ```

   The specific Stop key sequence depends on your keyboard type. For example, you can press Stop-A or L1-A. On terminals, press the Break key.

2. **Synchronize the file systems.**

   ```
   ok sync
   ```

3. **When you see the `syncing file systems...` message, press the Stop key sequence again.**

4. **Type the appropriate `boot` command to start the boot process.**

   For more information, see the `boot`(1M) man page.

5. **Verify that the system was booted to the specified run level.**

   ```
   # who -r
    .       run-level 3  May  2 07:39     3      0  S
   ```

Example 11–6 SPARC: Stopping the System for Recovery Purposes

```
    Press Stop-A
ok sync
syncing file systems...
    Press Stop-A
ok boot
```

▼ SPARC: How to Boot a System for Recovery Purposes

Use this procedure when an important file, such as `/etc/passwd`, has an invalid entry and causes the boot process to fail.

Use the stop sequence described in this procedure if you do not know the `root` password or if you can't log in to the system. For more information, see "SPARC: How to Stop the System for Recovery Purposes" on page 193.

Substitute the device name of the file system to be repaired for the *device-name* variable in the following procedure. If you need help identifying a system's device names, refer to "Displaying Device Configuration Information" in *System Administration Guide: Devices and File Systems*.

Steps

1. **Stop the system by using the system's Stop key sequence.**
2. **Boot the system in single-user mode.**
   - Boot the system from the Solaris Software 1 CD or DVD,
     - Insert the Solaris installation media into the drive.
     - Boot from the installation media in single-user mode.

       ok **boot cdrom -s**
   - Boot the system from the network if an installation server or remote CD or DVD drive is not available.

     ok **boot net -s**
3. **Mount the file system that contains the file with an invalid entry.**

   # **mount /dev/dsk/***device-name* **/a**
4. **Change to the newly mounted file system.**

   # **cd /a/***file-system*
5. **Set the terminal type.**

   # **TERM=***sun*
   # **export TERM**
6. **Remove the invalid entry from the file by using an editor.**

   # **vi** *filename*
7. **Change to the root (/) directory.**

   # **cd /**
8. **Unmount the /a directory.**

   # **umount /a**
9. **Reboot the system.**

   # **init 6**
10. **Verify that the system booted to run level 3.**

    The login prompt is displayed when the boot process has finished successfully.

    hostname console login:

Example 11–7 SPARC: Booting a System for Recovery Purposes (Damaged Password File)

The following example shows how to repair an important system file (in this case, `/etc/passwd`) after booting from a local CD-ROM.

```
ok boot cdrom -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi passwd
    (Remove invalid entry)
# cd /
# umount /a
# init 6
```

Example 11–8 SPARC: Booting a System if You Forgot the `root` Password

The following example shows how to boot the system from the network when you have forgotten the `root` password. This example assumes that the network boot server is already available. Be sure to apply a new `root` password after the system has rebooted.

```
ok boot net -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi shadow
    (Remove root's encrypted password string)
# cd /
# umount /a
# init 6
```

SPARC: Forcing a Crash Dump and Rebooting the System

Forcing a crash dump and rebooting the system is sometimes necessary for troubleshooting purposes. The `savecore` feature is enabled by default.

For more information on system crash dumps, see Chapter 24, "Managing System Crash Information (Tasks)," in *System Administration Guide: Advanced Administration*.

▼ SPARC: How to Force a Crash Dump and Reboot of the System

Use this procedure to force a crash dump of the system. The example that follows this procedure shows how to use the `halt -d` command to force a crash dump of the system. You will need to manually reboot the system after running this command.

Steps

1. **Type the stop key sequence for your system.**

   The specific stop key sequence depends on your keyboard type. For example, you can press Stop-A or L1-A. On terminals, press the Break key.

   The PROM displays the `ok` prompt.

2. **Synchronize the file systems and write the crash dump.**

   ```
   > n
   ok sync
   ```

   After the crash dump is written to disk, the system will continue to reboot.

3. **Verify the system boots to run level 3.**

   The login prompt is displayed when the boot process has finished successfully.

   hostname `console login:`

Example 11–9 SPARC: Forcing a Crash Dump and Reboot of the System by Using the `halt -d` Command

This example shows how to force a crash dump and reboot of the system `jupiter` by using the `halt -d` and `boot` command. Use this method to force a crash dump and reboot of the system.

```
# halt -d
Jul 21 14:13:37 jupiter halt: halted by root

panic[cpu0]/thread=30001193b20: forced crash dump initiated at user request

000002a1008f7860 genunix:kadmin+438 (b4, 0, 0, 0, 5, 0)
  %l0-3: 0000000000000000 0000000000000000 0000000000000004 0000000000000004
  %l4-7: 00000000000003cc 0000000000000010 0000000000000004 0000000000000004
000002a1008f7920 genunix:uadmin+110 (5, 0, 0, 6d7000, ff00, 4)
  %l0-3: 0000030002216938 0000000000000000 0000000000000001 0000004237922872
  %l4-7: 000000423791e770 0000000000004102 0000030000449308 0000000000000005

syncing file systems... 1 1 done
dumping to /dev/dsk/c0t0d0s1, offset 107413504, content: kernel
100% done: 5339 pages dumped, compression ratio 2.68, dump succeeded
Program terminated
ok boot
Resetting ...
```

```
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #10933339.
Ethernet address 8:0:20:a6:d4:5b, Host ID: 80a6d45b.

Rebooting with command: boot
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a
File and args: kernel/sparcv9/unix
SunOS Release 5.10 Version s10_60 64-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: hme0.
add net default: gateway 172.20.27.248
Hostname: jupiter
The system is coming up.  Please wait.
NIS domain name is example.com
.
.
.
System dump time: Wed Jul 21 14:13:41 2004
Jul 21 14:15:23 jupiter savecore: saving system crash dump
in /var/crash/jupiter/*.0
Constructing namelist /var/crash/jupiter/unix.0
Constructing corefile /var/crash/jupiter/vmcore.0
100% done: 5339 of 5339 pages saved

Starting Sun(TM) Web Console Version 2.1-dev...
.
.
.
```

▼ SPARC: How to Boot the System With the Kernel Debugger (kmdb)

This procedure shows you the basics for loading the kernel debugger (kmdb). For more detailed information, see the *Solaris Modular Debugger Guide*.

Steps

1. **Halt the system, causing it to display the ok prompt.**

   To halt the system gracefully, use the /usr/sbin/halt command.

2. **Type either boot kmdb or boot -k to request the loading of the kernel debugger. Press return.**

3. **Enter the kernel debugger.**

   The method used to enter the debugger is dependent upon the type of console that is used to access the system:

   - If a locally attached keyboard is being used, press Stop-A or L1–A, depending upon the type of keyboard.

- If a serial console is being used, send a break by using the method that is appropriate for the type of serial console that is being used.

A welcome message is displayed when you enter the kernel debugger for the first time.

```
Rebooting with command: kadb
Boot device: /iommu/sbus/espdma@4,800000/esp@4,8800000/sd@3,0
.
.
.
```

Example 11–10 SPARC: Booting the System With the Kernel Debugger (`kmdb`)

```
ok boot kmdb
Resetting...

Executing last command: boot kmdb -d
Boot device: /pci@1f,0/ide@d/disk@0,0:a File and args: kmdb -d
Loading kmdb...
```

CHAPTER 12

x86: Booting a System (Tasks)

This chapter describes the procedures for booting an x86 based system. See "x86: Booting a System (Task Map)" on page 199 for information on the procedures associated with booting an x86 based system.

For more information about 64-bit computing on the x86 based platform, see "x86: Support for 64-Bit Computing" on page 146.

For overview information about the boot process, see Chapter 8. For step-by-step instructions on booting a SPARC based system, see Chapter 11.

x86: Booting a System (Task Map)

Task	Description	For Instructions
Boot a system.	Select one of the following boot options:	

Task	Description	For Instructions
	■ Boot to run level 3 – Used after shutting down the system or performing some system hardware maintenance task. ■ Boot to run level S – Used after performing a system maintenance task such as backing up a file system. ■ Boot interactively – Used after making temporary changes to a system file or the kernel for testing purposes. ■ Used to boot a PXE or non-PXE device from the network with the default network configuration strategy. This method is used for booting a diskless client.	"x86: How to Boot a System to Run Level 3 (Multiuser Level)" on page 202 "x86: How to Boot a System to Run Level S (Single-User Level)" on page 204 "x86: How to Boot a System Interactively" on page 206 "x86: How to Boot a System From the Network" on page 208
Use the Device Configuration Assistant on a Solaris Operating System x86 based system.	Used after changing the hardware configuration of the system. This utility enables you to boot the Solaris system from a different boot device, configure new or misconfigured hardware, or perform other device-related or boot-related tasks.	"x86: How to Enter the Device Configuration Assistant" on page 210

Task	Description	For Instructions
Boot a system for recovery purposes.	Boot for recovery purposes - Used to boot the system when a damaged file is preventing the system from booting. You might need to do one or both of the following to boot for recovery purposes: 1. First, stop the system to attempt recovery. 2. Force a crash dump and reboot the system - Used to force a crash dump for troubleshooting purposes. 3. Boot to repair an important system file that is preventing the system from booting successfully. Boot `kmdb` – Used to troubleshoot system problems.	"x86: How to Stop a System for Recovery Purposes" on page 210 "x86: Forcing a Crash Dump and Rebooting the System" on page 215 "x86: How to Boot a System for Recovery Purposes" on page 211 "x86: How to Boot a System With the Kernel Debugger (`kmdb`)" on page 213 Use the `reboot` and `halt` command with the `-d` option if you do not have time to debug the system interactively. Running the `halt` command with the `-d` option requires a manual reboot of the system afterwards. Whereas, if you use the `reboot` command, the system boots automatically.
Troubleshoot boot problems on systems that have 64-bit computing capabilities.	If you have hardware that requires the system to load one or more device drivers that are not available in 64-bit mode, booting the system to 64-bit mode could fail. You would then need to boot the system to 32-bit mode.	"64-bit x86: Troubleshooting a Failed 64-Bit Boot" on page 217

x86: Booting a System

The following procedures use the reset button to restart the system. If your system does not have a reset button, use the power switch to restart the system. You might be able to press Ctrl-Alt-Del to interrupt system operation, depending upon the state of the system.

▼ x86: How to Boot a System to Run Level 3 (Multiuser Level)

Use this procedure to boot a system that is currently at run level 0 to run level 3.

Steps

1. **If the system displays the `Press any key to reboot` prompt, press any key to reboot the system.**

   You can also use the Reset button at this prompt. If the system is shut down, turn the system on with the power switch.

   The Current Boot Parameters menu is displayed after a few minutes.

2. **Type `b` to boot the system to run level 3. Press Enter.**

   If you do not make a selection within five seconds, the system is automatically booted to run level 3.

3. **Verify that the system has booted to run level 3.**

   The login prompt is displayed when the boot process has finished successfully.

   hostname `console login:`

Example 12–1 x86: Booting a System to Run Level 3 (Multiuser Level)

For new installations of the Solaris 10 OS, typing `b` at the boot prompt automatically boots 64-bit capable x86 based systems to 64-bit mode. For upgrade installations of the Solaris 10 OS, typing `b` at the boot prompt also boots 64-bit capable x86 based systems to 64-bit mode, unless the `eeprom boot-file` parameter was previously set to a value other than `kernel/unix`.

This example shows how to boot an x86 based system that has 64-bit computing capabilities to run level 3.

```
Press any key to reboot
            .
            .
            .
                        <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>      to boot with options
or      i <ENTER>                               to enter boot interpreter
or      <ENTER>                                 to boot with defaults

                    <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b
SunOS Release 5.10 Version amd64-gate-2004-09-27 64-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
DEBUG enabled
```

```
Hostname: venus
NIS domain name is example.com
checking ufs filesystems
/dev/rdsk/c1d0s7: is logging.
venus console login:
```

Example 12–2 64-bit x86: Manually Booting a System That Has 64-Bit Computing Capabilities in 64-Bit Mode to Run Level 3 (Multiuser Level)

For new installations of the Solaris 10 OS, typing `b` at the boot prompt automatically boots 64-bit capable x86 based systems to 64-bit mode. For upgrade installations of the Solaris 10 OS, typing `b` at the boot prompt also boots 64-bit capable x86 based systems to 64-bit mode, unless the `eeprom boot-file` parameter was previously set to a value other than `kernel/unix`.

This example shows how to *manually* boot this type of system in 64-bit mode to run level 3.

```
# init 0
# svc.startd: The system is coming down.  Please wait.
svc.startd: 68 system services are now being stopped.
umount: /etc/svc/volatile busy
svc.startd: The system is down.
syncing file systems... done
Press any key to reboot.

Initializing system
Please wait...


                      <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type    b [file-name] [boot-flags] <ENTER>    to boot with options
or      i <ENTER>                             to enter boot interpreter
or      <ENTER>                               to boot with defaults

                  <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b kernel/amd64/unix
SunOS Release 5.10 Version amd64-gate-2004-09-27 64-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
DEBUG enabled
Hostname: venus
NIS domain name is example.com
checking ufs filesystems
/dev/rdsk/c1d0s7: is logging.
venus console login:
```

Example 12–3 32-bit x86: Manually Booting a System That Has 64-Bit Computing Capabilities in 32-Bit Mode to Run Level 3 (Multiuser Level)

For new installations of the Solaris 10 OS, typing `b` at the boot prompt automatically boots 64-bit capable x86 based systems to 64-bit mode. For upgrade installations of the Solaris 10 OS, typing `b` at the boot prompt also boots 64-bit capable x86 based systems to 64-bit mode, unless the `eeprom boot-file` parameter was previously set to a value other than `kernel/unix`.

This example shows how to *manually* boot this type of system in 32-bit mode to run level 3.

```
# init 0
# svc.startd: The system is coming down.  Please wait.
svc.startd: 68 system services are now being stopped.
umount: /etc/svc/volatile busy
svc.startd: The system is down.
syncing file systems... done
Press any key to reboot.
Resetting...
If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.


Initializing system
Please wait...


                     <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type    b [file-name] [boot-flags] <ENTER>     to boot with options
or      i <ENTER>                              to enter boot interpreter
or      <ENTER>                                to boot with defaults

                  <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b kernel/unix
SunOS Release 5.10 Version amd64-gate-2004-09-30 32-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
DEBUG enabled
Hostname: venus
NIS domain name is example.com
checking ufs filesystems
/dev/rdsk/c1d0s7: is logging.
venus console login:
```

▼ x86: How to Boot a System to Run Level S (Single-User Level)

Use this procedure to boot a system that is currently at run level 0 to run level S.

Steps

1. **If the system displays the `Press any key to reboot` prompt, press any key to reboot the system.**

   You can also use the Reset button at this prompt. If the system is shut down, turn the system on with the power switch.

   The Current Boot Parameters menu is displayed after a few minutes.

2. **Type `b -s` to boot the system to run level S. Press Enter.**

   If you do not make a selection within five seconds, the system is automatically booted to run level 3.

3. **Type the superuser password, if prompted.**

4. **Verify that the system is at run level S.**

   ```
   # who -r
      .       run-level S  Jul 19 14:37     S      0  3
   ```

5. **Perform the maintenance task that required the run level change to S.**

6. **After you complete the system maintenance task, type Control-D to bring the system to the multiuser state.**

Example 12–4 x86: Booting a System to Run Level S (Single-User Level)

```
Press any key to reboot.
Resetting...
.
.
.
Initializing system
Please wait...


                     <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type    b [file-name] [boot-flags] <ENTER>     to boot with options
or      i <ENTER>                              to enter boot interpreter
or      <ENTER>                                to boot with defaults

                  <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -s
SunOS Release 5.10 Version amd64-gate-2004-09-30 32-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
DEBUG enabled
Booting to milestone "milestone/single-user:default".
Hostname: venus
NIS domain name is example.com
Requesting System Maintenance Mode
```

```
SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
Entering System Maintenance Mode
.
.
.
# who -r
   .       run-level S  Jul 19 14:37     S      0  3
    (Perform some maintenance task)
# ^D
```

▼ x86: How to Boot a System Interactively

Use this procedure to boot a system when you need to specify an alternate kernel or `/etc/system` file.

Steps

1. **If the system displays the `Press any key to reboot` prompt, press any key to reboot the system.**

   You can also use the reset button at this prompt. If the system is shut down, turn the system on with the power switch.

   The Primary Boot Subsystem menu is displayed after a few minutes.

2. **Select the Solaris partition (if not marked as active) from the list. Press Enter.**

   If you do not make a selection within five seconds, the active boot partition is selected automatically.

   The Current Boot Parameters menu is displayed after a few minutes.

3. **Type `b -a` to boot the system interactively. Press Enter.**

   If you do not make a selection within five seconds, the system is automatically booted to run level 3.

4. **Answer the following system prompts.**

   a. **When prompted, enter the name of the kernel to use for booting.**

      Press enter to use the default kernel file name. Otherwise, provide the name of an alternate kernel, press Enter.

   b. **When prompted, provide an alternate path for the module directories.**

      Press enter to use the default module directories. Otherwise, provide the alternate paths to module directories, press Enter.

   c. **When prompted, provide the name of an alternate system file.**

      Type `/dev/null` if your `/etc/system` file has been damaged.

   d. **When prompted, enter the `root` file system type.**

Press enter to select local disk booting with UFS, which is the default, or enter NFS for network booting.

e. When prompted, enter the physical name of `root` device.

Provide an alternate device name or press return to use the default.

5. If you are not prompted to answer these questions, verify that you typed the `boot -a` command correctly.

Example 12–5 x86: Booting a System Interactively

In the following example, the default choices (shown in square brackets `[]`) are accepted.

```
Press any key to reboot.
Resetting...
.
.
.
Autobooting from bootpath: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a


If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.


Initializing system
Please wait...


                     <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type    b [file-name] [boot-flags] <ENTER>     to boot with options
or      i <ENTER>                              to enter boot interpreter
or      <ENTER>                                to boot with defaults
Running Configuration Assistant...
                  <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -a
Enter default directory for modules [/platform/i86pc/kernel /kernel /usr/kernel]:
    Press Enter
Name of system file [etc/system]:     Press Enter
SunOS Release 5.10 Version amd64-gate-2004-09-30 32-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
DEBUG enabled
root filesystem type [ufs]:     Press Enter
Enter physical name of root device[/pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a]:     Press Enter
Hostname: venus
NIS domain name is example.com
checking ufs filesystems
```

```
/dev/rdsk/c1d0s7: is logging.
venus console login:
```

x86: Booting From the Network

Any system can boot from the network if a boot server is available. You might want to boot a stand-alone system from the network for recovery purposes if the system cannot boot from the local disk.

You can boot Solaris OS x86 based systems directly from a network without the Solaris boot diskette on x86 based systems that support the Preboot Execution Environment (PXE) network booting protocol. The PXE network boot is available only for devices that implement the Intel Preboot Execution Environment specification. If the system is capable of a PXE network boot, you might want to boot the system directly from the network without using either the Device Configuration Assistant boot diskette or the Solaris Software 1 CD or DVD.

▼ x86: How to Boot a System From the Network

There are two network configuration strategies, Reverse Address Resolution Protocol (RARP) or Dynamic Host Configuration Protocol (DHCP). The default network boot strategy for a PXE network boot is DHCP. The default network boot strategy for non-PXE devices is RARP. For non-PXE devices, you can use either strategy, depending on whether a RARP boot server or a DHCP boot server is available on your network.

In this release, the system boots automatically if you are performing a PXE network boot, or if you are booting the system from the Solaris Software 1 CD or DVD. The Device Configuration Assistant menu is no longer displayed by default. If you are booting a non-PXE device, you will need to follow the steps in this procedure that describe how to enter the Device Configuration Assistant menu to change the network configuration.

Steps

1. **Insert the Device Configuration Assistant boot diskette or the Solaris Software 1 CD or DVD that you want to boot from. Or, use the system or network adapter BIOS configuration program to enable the PXE network boot.**

   - If you are using the boot diskette, the first menu of the Device Configuration Assistant is displayed.
   - If you are using the Solaris Software 1 CD, DVD, or booting a PXE device from the network, the system boots automatically.

     If you choose to change the network configuration and enter the Device Configuration Assistant menu, press Esc when the following message is displayed:

```
If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.
```

The Device Configuration Assistant screen is displayed.

2. **If the system displays the `Press any key to reboot` prompt, press any key to reboot the system.**

   You can also use the reset button at this prompt. If the system is shut down, turn the system on with the power switch.

3. **Press the F2 key (F2_Continue) to scan for devices.**

   Device identification is performed. Then, the Identified Devices screen is displayed.

4. **Press the F2 key (F2_Continue) to load drivers.**

   Bootable drivers are loaded. Then, the Boot Solaris menu is displayed.

5. **Use the Device Configuration Assistant to change the network configuration.**

   a. **Press the F4 key (F4_Boot Tasks).**

   b. **Select `Set Network Configuration Strategy`. Press the F2 key (F2_Continue).**

   c. **Select either `RARP` or `DHCP` and press the F2 key (F2_Continue).**

   Note – The previous step applies only if you are booting a non-PXE device from the network. For a PXE network boot, you must use DHCP, which is the default network boot strategy.

   A screen that confirms your new network boot strategy is displayed. Your network boot strategy selection is saved as the default network boot method for the next time this diskette is used for booting.

   d. **Press F3_Back to return to the Boot Solaris menu.**

6. **Select `NET` as the boot device. Then, press F2_Continue to boot the network device.**

   The Solaris boot option screen is displayed.

x86: Using the Device Configuration Assistant

The Device Configuration Assistant for Solaris Operating System x86 based systems is a program that enables you to perform various hardware configuration and booting tasks. You can access the Device Configuration Assistant menu from either of the following:

- Solaris boot diskette
- Solaris Software 1 CD or DVD
- PXE network boot
- Hard disk with Solaris OS installed

For the procedures in this chapter, you might be requested to insert the Device Configuration Assistant boot diskette to boot the Configuration Assistant. Alternately, if your system's BIOS supports booting from the CD or DVD, you can insert the Solaris Software 1 CD or DVD to boot the Device Configuration Assistant.

▼ x86: How to Enter the Device Configuration Assistant

Steps

1. **Boot the system.**
   - If you are booting from the Device Configuration boot diskette, the first menu of the Device Configuration Assistant is displayed after a few minutes.
   - If you are booting from the Solaris Software 1 CD, DVD, hard disk, or performing a PXE network boot, the following message is displayed:

     ```
     If the system hardware has changed, or to boot from a different
     device, interrupt the autoboot process by pressing ESC.

     Press ESCape to interrupt autoboot in 5 seconds.
     ```

     If you choose to enter the Device Configuration Assistant menu, press Esc to interrupt the autoboot process.

     The Device Configuration Assistant menu is displayed.
2. **If the system displays the `Press any key to reboot` prompt, press any key to reboot the system.**

   You can also use the reset button at this prompt. If the system is shut down, turn the system on with the power switch.

▼ x86: How to Stop a System for Recovery Purposes

Steps

1. **Stop the system by using one of the following commands, if possible:**
   - If the system is running, become superuser and type `init 0` to stop the system. After the `Press any key to reboot` prompt appears, press any key to reboot the system.
   - If the system is running, become superuser and type `init 6` to reboot the system.
2. **If the system doesn't respond to any input from the mouse or keyboard, press the Reset key, if it exists, to reboot the system. Or, you can use the power switch**

to reboot the system.

▼ x86: How to Boot a System for Recovery Purposes

Follow these steps to boot the system to repair a critical system resource. The example shows you how to boot from a Solaris Software 1 CD or from the network, mount the root (/) file system on the disk, and repair the `/etc/passwd` file.

Substitute the device name of the file system to be repaired for the *device-name* variable. the following procedure. If you need help identifying a system's device names, refer to "Displaying Device Configuration Information" in *System Administration Guide: Devices and File Systems*.

Steps

1. **Stop the system by using the system's Stop key sequence.**

   Use the Stop key sequence for your system if you don't know the `root` password, or if you can't log in to the system. For more information, see "x86: How to Stop a System for Recovery Purposes" on page 210.

2. **Boot the system from the Solaris Software 1 CD, DVD, or from the network, to single-user mode.**

   a. **Insert the Device Configuration Assistant boot diskette or the Solaris Software 1 CD or DVD that you want to boot from.**

   Note – If you are using the boot diskette the Device Configuration Assistant menu is displayed. If you are using the Solaris Software 1 CD or DVD, the system boots automatically. To enter the Device Configuration Assistant menu, press Esc to interrupt the boot process, when prompted by the system.

   ---

   b. **If the system displays the `Press any key to reboot` prompt, press any key to reboot the system.**

      You can also use the Reset button at this prompt. If the system is shut down, turn the system on with the power switch.

3. **The Current Boot Parameters menu is displayed after a few minutes.**

4. **Type `b -s` at the prompt. Press Enter.**

   After a few minutes, the single-user mode # prompt is displayed.

5. **Mount the root (/) file system that contains the invalid `passwd` file.**

6. **Change to the newly mounted `etc` directory.**

7. **Make the necessary change to the file by using an editor.**

8. **Change to the root (/) directory.**

9. **Unmount the `/a` directory.**

10. **Reboot the system.**

11. **Verify that the system has booted to run level 3.**

    The login prompt is displayed when the boot process has finished successfully.

    host-name `console login:`

Example 12–6 x86: Booting a System for Recovery Purposes

The following example shows how to repair the `/etc/passwd` file after booting the system automatically from a local CD-ROM.

```
SunOS Secondary Boot version 3.00


                                Solaris Booting System


Running Configuration Assistant...



If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.



Initializing system
Please wait...


                           <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@1/sd@0,0:a
Boot args:


Select the type of installation you want to perform:

               1 Solaris Interactive
               2 Custom JumpStart
               3 Solaris Interactive Text (Desktop session)
               4 Solaris Interactive Text (Console session)
```

```
Enter the number of your choice followed by the <ENTER> key.
Alternatively, enter custom boot arguments directly.

If you wait for 30 seconds without typing anything,
an interactive installation will be started.

Select type of installation: b -s
      .
      .
      .
# mount /dev/dsk/c0t0d0s0 /a
      .
      .
      .
# cd /a/etc
# vi passwd
    (Remove invalid entry)
# cd /
# umount /a
# init 6
```

▼ x86: How to Boot a System With the Kernel Debugger (kmdb)

This procedure shows the basics for loading the kernel debugger (kmdb). The savecore feature is enabled by default. For more detailed information about using the kernel debugger, see the *Solaris Modular Debugger Guide*.

Steps

1. **Boot the system.**
2. **Type b -k at the Select (b)oot or (i)nterpreter prompt. Press Enter.**
3. **Access the kernel debugger.**

   The method used to enter the debugger is dependent upon the type of console that is used to access the system:

   - If a locally attached keyboard is being used, press F1–A.
   - If a serial console is being used, send a break by using the method appropriate to the type of serial console that is being used.

   A welcome message is displayed when you access the kernel debugger for the first time.

Example 12–7 x86: Booting a System With the Kernel Debugger (kmdb)

Typing b -k at the Select (b)oot or (i)nterpreter boot prompt boots a system to its default mode and also loads kmdb. This example shows how to boot an x86 based system that has 32–bit computing capabilities to 32–bit mode and also load kmdb.

```
Press any key to reboot.
.
.
.
<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:

Type    b [file-name] [boot-flags] <ENTER>     to boot with options
or      i <ENTER>                              to enter boot interpreter
or      <ENTER>                                to boot with defaults
Running Configuration Assistant...
                  <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -k
Loading kmdb...
SunOS Release 5.10 Version gate:2004-10-21 32-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
.
.
.
```

Example 12–8 64-bit x86: Manually Booting a System That Has 64-Bit Computing Capabilities to 64-Bit Mode With the Kernel Debugger (kmdb)

This example shows how to manually boot an x86 based system that has 64-bit computing capabilities to 64-bit mode with kmdb.

```
Press any key to reboot
        .
        .
        .
            <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>     to boot with options
or      i <ENTER>                              to enter boot interpreter
or      <ENTER>                                to boot with defaults

            <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b kernel/amd64/unix -k
Loading kmdb...
```

Example 12–9 32-bit x86: Manually Booting a System That Has 64-Bit Computing Capabilities to 32-Bit Mode With the Kernel Debugger (kmdb)

This example shows how to manually boot an x86 based system that has 64-bit computing capabilities to 32-bit mode with kmdb.

```
Press any key to reboot
        .
        .
```

```
        .
            <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>     to boot with options
or      i <ENTER>                              to enter boot interpreter
or      <ENTER>                                to boot with defaults

            <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b kernel/unix -k
Loading kmdb...
```

x86: Forcing a Crash Dump and Rebooting the System

Forcing a crash dump and rebooting the system is sometimes necessary for troubleshooting purposes. The `savecore` feature is enabled by default.

For more information on system crash dumps, see Chapter 24, "Managing System Crash Information (Tasks)," in *System Administration Guide: Advanced Administration*.

▼ x86: How to Force a Crash Dump and Reboot of the System

If you cannot use the `reboot -d` or the `halt -d` command, you can use the kernel debugger, `kmdb`, to force a crash dump. The kernel debugger must have been loaded, either at boot, or with the `mdb -k` command, for the following procedure to work.

Note – You must be in text mode to enter the kernel debugger (`kmdb`). So, first exit any window system.

Steps

1. **If a locally-attached keyboard is being used as the system console, press F1-A on that keyboard. If the system is configured to use a remote (serial) console, use the mechanism that is appropriate to that console to send a break character.**

   The `kmdb` prompt is displayed.

2. **Use the `systemdump` macro to induce a crash.**

   ```
   [0]> $<systemdump
   ```

   Panic messages are displayed, the crash dump is saved, and the system reboots.

3. **Verify that the system has rebooted by logging in at the console login prompt.**

Example 12–10 x86: Forcing a Crash Dump and Reboot of the System by Using `halt -d`

This example shows how to force a crash dump and reboot of the x86 based system `neptune` by using the `halt -d` and `boot` commands. Use this method to force a crash dump of the system. You will need to manually reboot the system after running the `halt` command with the `-d` option.

```
# halt -d
Aug 11 12:51:27 neptune halt:
halted by <user> panic[cpu45]/thread=d3971a00: forced crash dump initiated at user request

d363ae58 genunix:kadmin+bd (5, 0, 0, d3fefac0)
d363af88 genunix:uadmin+88 (5, 0, 0, 0, 0, d363afb4)

syncing file systems... done
dumping to /dev/dsk/c0t0d0s1, offset 107806720, content: kernel
100% done: 40223 pages dumped, compression ratio 4.11, dump succeeded
Press any key to reboot.
Resetting...
.
.
.
SunOS Secondary Boot version 3.00
Autobooting from bootpath: /pci@0,0/pci1028,10a@3/sd@0,0:a
Running Configuration Assistant...
If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.


Initializing system
Please wait...

<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci1028,10a@3/sd@0,0:a
Boot args:

Type    b [file-name] [boot-flags] <ENTER>    to boot with options
or      i <ENTER>                             to enter boot interpreter
or      <ENTER>                               to boot with defaults

                  <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter:
Loading kmdb...
SunOS Release 5.10 Version s10_62 32-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: iprb0.
add net default: gateway 172.20.26.248
Hostname: neptune
The system is coming up.  Please wait.
checking ufs filesystems
/dev/rdsk/c0t0d0s7: is logging.
```

```
NIS domain name is example.com
starting rpc services: rpcbind keyserv ypbind done.
Setting netmask of iprb0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4: gateway venus
syslog service starting.
System dump time: Wed Aug 11 12:51:29 2004
Aug 11 13:13:26 venus savecore: saving system crash dump in /var/crash/venus/*.1
Constructing namelist /var/crash/venus/unix.1
Constructing corefile /var/crash/venus/vmcore.1
100% done: 42157 of 42157 pages saved
volume management starting.
The system is ready.
.
.
.
```

64-bit x86: Troubleshooting a Failed 64-Bit Boot

In some instances, an attempt to boot a 64-bit capable x86 based system to 64-bit mode might fail. This failure might produce an error similar to the following:

```
Select (b)oot or (i)nterpreter: b kernel/amd64/unix
.
.
.
pci: cannot load driver
Cannot load drivers for /pci@0,0/pci1022,7450@a/pci17c2,10@4/sd@0,0:a
(Can't load the root filesystem)
Press any key to reboot.
.
.
.
```

In the event such a failure occurs, boot the system to 32-bit mode by typing the following command at the `Select (b)oot or (i)nterpreter` boot prompt:

```
Select (b)oot or (i)nterpreter: b kernel/unix
```

For more information, see "x86: Manually Booting a System That Is Capable of 64-Bit Computing" on page 146.

CHAPTER 13

The Boot Process (Reference)

This chapter describes the firmware used for booting SPARC based and x86 based systems. This chapter also provides an overview of the boot process on each platform.

This is a list of the reference information in this chapter.

For step-by-step instructions on booting a system, see Chapter 11 or Chapter 12.

SPARC: The Boot PROM

Each SPARC based system has a programmable read-only memory (PROM) chip with a program called the *monitor*. The monitor controls the operation of the system before the Solaris kernel is available. When a system is turned on, the monitor runs a quick self-test procedure to check the hardware and memory on the system. If no errors are found, the system begins the automatic boot process.

Note – Some older systems might require PROM upgrades before they will work with the Solaris system software. Contact your local service provider for more information.

SPARC: The Boot Process

The following table describes the boot process on SPARC based systems.

TABLE 13–1 SPARC: Description of the Boot Process

Boot Phase	Description
Boot PROM	1. The PROM displays system identification information and then runs self-test diagnostics to verify the system's hardware and memory.
	2. The PROM loads the primary boot program, `bootblk`. This program's purpose is to load the secondary boot program (that is located in the UFS file system) from the default boot device.
Boot programs	3. The `bootblk` program finds and executes the secondary boot program, `ufsboot`, and loads it into memory.
	4. After the `ufsboot` program is loaded, the `ufsboot` program loads the kernel.
Kernel initialization	5. The kernel initializes itself and begins loading modules by using `ufsboot` to read the files. When the kernel has loaded enough modules to mount the root (`/`) file system, the kernel unmaps the `ufsboot` program and continues, using its own resources.
	6. The kernel creates a user process and starts the `/sbin/init` process. This process starts other processes by reading the `/etc/inittab` file.
`init`	7. In this Solaris release, the `/sbin/init` process starts `/lib/svc/bin/svc.startd`, which starts system services that do the following: ■ Check and mount file systems ■ Configure network and devices ■ Start various processes and perform system maintenance tasks In addition, `svc.startd` executes the run control (`rc`) scripts for compatibility.

x86: The PC BIOS

Before the kernel is started, the system is controlled by the read-only-memory (ROM) Basic Input/Output System (BIOS), which is the firmware interface on a PC.

Hardware adapters can have an on-board BIOS that displays the physical characteristics of the device and can be used to access the device.

During the startup sequence, the PC BIOS checks for the presence of any adapter BIOS, and if found, loads and executes each adapter BIOS. Each individual adapter's BIOS runs self-test diagnostics and displays device information.

x86: Boot Subsystems

During the boot process, the boot subsystem menus allow you to customize boot choices. If the system receives no response during the timeout periods, it continues to boot automatically using the default selections. You can stop the boot process when each boot subsystem menu is displayed. Or, you can let the boot process continue automatically.

At three points during the Solaris boot process, you can make the following choices about a booting system:

- **Primary Boot Subsystem (Partition Boot Menu)** – This first menu appears if multiple operating systems exist on the disk. The menu enables you to boot any of the operating systems installed. By default, the operating system that is designed as *active* is booted.

  Note that if you choose to boot a system other than the Solaris Operating System, you cannot reach the next two menus.
- **Interrupt the Autoboot Process** – If the autoboot process is interrupted, you can access the Device Configuration Assistant menu.

  The Solaris Device Configuration Assistant enables you to boot the Solaris system from a different boot device, configure new or misconfigured hardware, or perform other device-related or boot-related tasks.
- **Current Boot Parameters menu** – Two forms of this menu exist, one menu for a normal Solaris boot and one menu for a Solaris installation boot:
  - The normal Current Boot Parameters menu enables you to boot the Solaris system with options, or enter the boot interpreter.
  - The install Current Boot Parameters menu enables you to select the type of installation to be performed or to customize the boot process.

The following table summarizes the purpose of the primary x86 based system boot interfaces. See the sections that follow for a detailed description and example of each boot interface.

TABLE 13–2 x86: Boot Subsystems

Boot Subsystem	Purpose
Primary Boot Subsystem (Partition Boot menu)	This menu appears if the disk you are booting from contains multiple operating systems, including the Solaris Operating System (Solaris OS).
Secondary Boot Subsystem	This menu appears each time you boot the Solaris release. The Solaris release is booted automatically unless you choose to run the Solaris Device Configuration Assistant by interrupting the autoboot process.
Solaris Device Configuration Assistant/Boot Diskette	There are two ways to access the Device Configuration Assistant menus: ■ Use the Device Configuration Assistant boot diskette or the Solaris Software 1 CD (on systems that can boot from the CD-ROM drive) to boot the system. ■ Interrupt the autoboot process when you boot the Solaris software from an installed disk.
Current Boot Parameters menu	This menu appears when you boot the Solaris release from the disk, CD-ROM, or the network. The menu presents a list of boot options.

Note – If you need to create the Solaris Device Configuration Assistant boot diskette, go to `http://www.sun.com/bigadmin/hcl/drivers/dca_diskettes/`.

x86: Booting the Solaris Release

In this release, if you are booting an x86 based system with the Solaris Software 1 CD, DVD, or performing a PXE network boot, the system will boot automatically. To use the Device Configuration Assistant, you must interrupt the boot process by pressing Esc when prompted by the system.

During the device identification phase, the Device Configuration Assistant does the following:

- Scans for devices that are installed on the system
- Displays the identified devices
- Enables you to perform optional tasks such as selecting a keyboard type or editing devices and their resources

During the boot phase, the Device Configuration Assistant does the following:

- Displays a list of devices from which to boot. The device marked with an asterisk (*) is the default boot device.

- Enables you to perform optional tasks, such as editing autoboot settings and property settings, and choosing the network configuration strategy.

The following section provides examples of menus that appear during the device identification phase. The device output varies based on your system configuration.

x86: Screens Displayed During the Device Identification Phase

Several screens are displayed as the Device Configuration Assistant attempts to identify devices on the system. This section provides examples of the following boot subsystem screens:

- Device Configuration Assistant screen
- Bus Enumeration screen
- Scanning Devices screen
- Identified Devices screen

x86: Device Configuration Assistant Screen

In this release, the autoboot process bypasses the Device Configuration Assistant menus, unless you press Esc when prompted by the system during the boot phase. If you choose to use the Device Configuration Assistant, the following screen is displayed.

```
Solaris Device Configuration Assistant

The Solaris(TM)Device Configuration Assistant
scans to identify system hardware, lists identified devices, and can
boot the Solaris software from a specified device. This program must be
used to install the Solaris operating environment, add a driver,
or change the hardware on the system.

> To perform a full scan to identify all system hardware, choose Continue.
> To diagnose possible full scan failures, choose Specific Scan.
> To add new or updated device drivers, choose Add Driver.

About navigation...
    - The mouse cannot be used.
    - If the keyboard does not have function keys or they do not respond,
      press ESC. The legend at the bottom of the screen will change to
      show the ESC keys to use for navigation.
    - The F2 key performs the default action.

F2_Continue        F3_Specific Scan        F4_Add Driver        F6_Help
```

x86: Bus Enumeration Screen

The Bus Enumeration screen appears briefly while the Device Configuration Assistant gathers hardware configuration data for devices that can be detected automatically.

```
Bus Enumeration

Determining bus types and gathering hardware configuration data ...

Please wait ...
```

x86: Scanning Devices Screen

The Scanning Devices screen appears while the Device Configuration Assistant manually scans for devices that can only be detected with special drivers.

```
Scanning Devices

The system is being scanned to identify system hardware.

If the scanning stalls, press the system's reset button. When the
system reboots, choose Specific Scan or Help.


Scanning: Floppy disk controller

#######################
|      |        |         |         |         |
0              20       40        60          80       100

Please wait ...
```

x86: Identified Devices Screen

The Identified Devices screen displays which devices have been identified on the system. From here, you can continue to the Boot Solaris menu. Or, you can perform the following optional device tasks:

- Setting a keyboard configuration
- Viewing and editing devices
- Setting up a serial console
- Saving and deleting configurations

```
Identified Devices

The following devices have been identified on this system. To identify
devices not on this list or to modify device characteristics, such as
keyboard configuration, choose Device Tasks. Platform types may be
included in this list.

ISA: Floppy disk controller
    ISA: Motherboard
    ISA: PnP bios: 16550-compatible serial controller
    ISA: PnP bios: 16550-compatible serial controller
    ISA: PnP bios: Mouse controller
    ISA: PnP bios: Parallel port
```

```
    ISA: System keyboard (US-English)
    PCI: Bus Mastering IDE controller
    PCI: Universal Serial Bus
    PCI: VGA compatible display adapter

F2_Continue   F3_Back   F4_Device Tasks   F6_Help
```

x86: Menus Displayed During the Boot Phase

During this phase, you can determine the way in which the system is booted. The following menus are displayed during the boot phase:

- Boot Solaris menu
- Current Boot Parameters menu

x86: Boot Solaris Menu

The Boot Solaris menu allows you to select the device from which to boot the Solaris release. You can also perform optional tasks, such as viewing and editing autoboot and property settings. Once you select a boot device and you choose Continue, the Solaris kernel begins to boot.

```
Boot Solaris
Select one of the identified devices to boot the Solaris kernel and
choose Continue.

To perform optional features, such as modifying the autoboot and property
settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

[X]  DISK: (*) Target 0:QUANTUM  FIREBALL1280A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  DISK: Target 1:ST5660A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  DISK: Target 0:Maxtor 9 0680D4
    on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  CD  : Target 1:TOSHIBA  CD-ROM XM-5602B  1546
    on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1

F2_Continue   F3_Back   F4_Boot Tasks   F6_Help
```

x86: Current Boot Parameters Menu

This menu appears each time you boot the Solaris release from the local disk. Let the five-second timeout elapse if you want to boot the default Solaris kernel. If you want to boot with different options, select an appropriate option before the timeout period elapses.

```
                    <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>       to boot with options
or      i <ENTER>                                to enter boot interpreter
or      <ENTER>                                  to boot with defaults


                    <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter:
```

x86: The Boot Process

The following table describes the boot process on x86 based systems.

TABLE 13–3 x86: Description of the Boot Process

Boot Phase	Description
BIOS	1. When the system is turned on, the BIOS runs self-test diagnostics to verify the system's hardware and memory. The system begins to boot automatically if no errors are found. If errors are found, error messages are displayed that describe recovery options.
	The BIOS of additional hardware devices are run at this time.
	2. The BIOS boot program tries to read the first disk sector from the boot device. This first disk sector on the boot device contains the master boot record `mboot`, which is loaded and executed. If no `mboot` file is found, an error message is displayed.
Boot Programs	3. The master boot record, `mboot`, contains disk information needed to find the active partition and the location of the Solaris boot program, `pboot`, loads and executes `pboot`, `mboot`.
	4. The Solaris boot program, `pboot`, loads `bootblk`, the primary boot program. The purpose of `bootblk` is to load the secondary boot program, which is located in the UFS file system.
	5. If there is more than one bootable partition, `bootblk` reads the `fdisk` table to locate the default boot partition, and builds and displays a menu of available partitions. You have a 30 seconds to select an alternate partition from which to boot. This step occurs only if there is more than one bootable partition present on the system.

TABLE 13–3 x86: Description of the Boot Process *(Continued)*

Boot Phase	Description
	6. `bootblk` finds and executes the secondary boot program, `boot.bin` or `ufsboot`, in the root (`/`) file system. You have five seconds to interrupt the autoboot to start the Solaris Device Configuration Assistant.
	7. The secondary boot program, `boot.bin` or `ufsboot`, starts a command interpreter that executes the `/etc/bootrc` script. This script provides a menu of choices for booting the system. The default action is to load and execute the kernel. You have a 5–second interval to specify a boot option or to start the boot interpreter.
Kernel initialization	8. The kernel initializes itself and begins loading modules by using the secondary boot program (`boot.bin` or `ufsboot`) to read the files. When the kernel has loaded enough modules to mount the root (`/`) file system, the kernel unmaps the secondary boot program and continues, using its own resources.
	9. The kernel creates a user process and starts the `/sbin/init` process, which starts other processes by reading the `/etc/inittab` file.
`init`	10. In this Solaris release, the `/sbin/init` process starts `/lib/svc/bin/svc.startd`, which starts system services that do the following: ■ Check and mount file systems ■ Configure network and devices ■ Start various processes and perform system maintenance tasks In addition, `svc.startd` executes the run control (`rc`) scripts for compatibility.

x86: Boot Files

In addition to the run control scripts and boot files, there are additional boot files that are associated with booting a Solaris x86 based system.

TABLE 13–4 x86: Boot Files

File	Description
`/etc/bootrc`	Contains menus and options for booting the Solaris release.
`/boot`	Contains files and directories needed to boot the system.

TABLE 13–4 x86: Boot Files *(Continued)*

File	Description
/boot/mdboot	DOS executable that loads the first-level bootstrap program (strap.com) into memory from disk.
/boot/mdbootbp	DOS executable that loads the first-level bootstrap program (strap.com) into memory from diskette.
/boot/rc.d	Directory that contains install scripts. Do not modify the contents of this directory.
/boot/solaris	Directory that contains items for the boot subsystem.
/boot/solaris/boot.bin	Loads the Solaris kernel or stand-alone kmdb. In addition, this executable provides some boot firmware services.
/boot/solaris/boot.rc	Prints the Solaris Operating System on an x86 system and runs the Device Configuration Assistant in DOS-emulation mode.
/boot/solaris/bootconf.exe	DOS executable for the Device Configuration Assistant.
/boot/solaris/bootconf.txt	Text file that contains internationalized messages for Device Configuration Assistant (bootconf.exe).
/boot/solaris/bootenv.rc	Stores eeprom variables that are used to set up the boot environment.
/boot/solaris/devicedb	Directory that contains the master file, a database of all possible devices supported with realmode drivers.
/boot/solaris/drivers	Directory that contains realmode drivers.
/boot/solaris/itup2.exe	DOS executable run during install time update (ITU) process.
/boot/solaris/machines	Obsolete directory.
/boot/solaris/nbp	File associated with network booting.
/boot/solaris/strap.rc	File that contains instructions on what load module to load and where in memory it should be loaded.
/boot/strap.com	DOS executable that loads the second-level bootstrap program into memory.

CHAPTER 14

Managing Services (Tasks)

This chapter covers the tasks required to manage and monitor the Service Management Facility (SMF). In addition, information that is related to managing run level scripts is provided. The following topics are covered:

Managing SMF Services (Task Map)

The following task map describes the procedures that are needed to use SMF.

Task	Description	For Instructions
Display the status of a service instance.	Displays the status of all running service instances.	"How to List the Status of a Service" on page 230
Display the service dependents.	Display the services that are dependent on the specified service.	"How to Show Which Services Are Dependent on a Service Instance" on page 232
Display the dependencies of a service.	Display the services that a specified service is dependent on. This information can be used to help identify what is preventing a service from starting.	"How to Show Which Services a Service Is Dependent On" on page 232

Task	Description	For Instructions
Disable a service instance.	Turns off a service that is not functioning properly or needs to be off to increase security.	"How to Disable a Service Instance" on page 233
Enable a service instance	Starts a service.	"How to Enable a Service Instance" on page 234
Restart a service instance.	Restart a service, without having to use separate commands to disable and then enable the service.	"How to Restart a Service" on page 235
Modify a service instance.	Modifies the configuration parameters of a specified service instance. Changes a configuration property of a service controlled by `inetd`. Changes the startup options of a service controlled by `inetd`.	"How to Modify a Service" on page 237 "How to Change a Property for an `inetd` Controlled Service" on page 239 "How to Modify a Command-Line Argument for an `inetd` Controlled Service" on page 240
Converts `inetd.conf` entries.	Converts `inetd` services into legacy-run services that can be monitored using SMF.	"How to Convert `inetd.conf` Entries" on page 241
Repairs a corrupt service configuration repository.	Replaces a corrupt repository with a default version.	"How to Repair a Corrupt Repository" on page 245
Boot a system with no milestones.	Boot a system with no milestones so that configuration problems that prevent booting can be fixed.	"How to Start Services Interactively During Boot" on page 246

Monitoring SMF Services

The following tasks show how to monitor SMF services.

▼ How to List the Status of a Service

This procedure can be used to show what services are running.

Step ● **Run the `svcs` command.**

Running this command without any options displays a status report of the service specified by the FMRI.

```
% svcs -l FMRI
```

Example 14–1 Showing the Status of the `rlogin` Service

This example shows the status of a service that includes many contracts.

```
% svcs -l network/login:rlogin
fmri         svc:/network/login:rlogin
enabled      true
state        online
next_state   none
restarter    svc:/network/inetd:/default
contract_id  42325 41441 40776 40348 40282 40197 39025 38381 38053\
 33697 28625 24652 23689 15352 9889 7194 6576 6360 5387 1475 3015\
 6545 6612 9302 9662 10484 16254 19850 22512 23394 25876 26113 27326\
 34284 37939 38405 38972 39200 40503 40579 41129 41194
```

Example 14–2 Showing the Status of the `sendmail` Service

This example shows the status of a service that includes dependencies.

```
% svcs -l network/smtp:sendmail
fmri         svc:/network/smtp:sendmail
enabled      true
state        online
next_state   none
restarter    svc:/system/svc/restarter:default
contract_id  29462
dependency   require_all/refresh file://localhost/etc/nsswitch.conf (-)
dependency   require_all/refresh file://localhost/etc/mail/sendmail.cf (-)
dependency   optional_all/none svc:/system/system-log (online)
dependency   require_all/refresh svc:/system/identity:domain (online)
dependency   require_all/refresh svc:/milestone/name-services (online)
dependency   require_all/none svc:/network/service (online)
dependency   require_all/none svc:/system/filesystem/local (online)
```

Example 14–3 Showing the Status of all Services

The following command lists all services that are installed on the system as well as the status of each service. The command displays those services that are disabled as well as those that are enabled.

```
% svcs -a
```

Example 14–4 Showing the Status of Services Controlled by `inetd`

The following command lists services that are controlled by `inetd`. Each service's FMRI is listed, along with the run state and whether the service is enabled or disabled.

```
% inetadm
```

▼ How to Show Which Services Are Dependent on a Service Instance

This procedure shows how to determine which service instances depend on the specified service.

Step ● **Display the service dependents.**

```
% svcs -D FMRI
```

Example 14–5 Displaying the Service Instances That Are Dependent on the Multiuser Milestone

The following example shows how to determine which service instances are dependent on the multiuser milestone.

```
% svcs -D milestone/multi-user
STATE          STIME    FMRI
online         Apr_08   svc:/milestone/multi-user-server:default
```

▼ How to Show Which Services a Service Is Dependent On

This procedure shows how to determine which services a specified service instance is dependent on.

Step ● **Display the service dependencies.**

```
% svcs -d FMRI
```

Example 14–6 Displaying the Service Instances That the Multiuser Milestone Is Dependent On

The following example shows the services instances that the multiuser milestone is dependent on.

```
% svcs -d milestone/multi-user:default
STATE          STIME    FMRI
disabled       Aug_24   svc:/platform/sun4u/sf880drd:default
online         Aug_24   svc:/milestone/single-user:default
online         Aug_24   svc:/system/utmp:default
online         Aug_24   svc:/system/system-log:default
online         Aug_24   svc:/system/system-log:default
online         Aug_24   svc:/system/rmtmpfiles:default
online         Aug_24   svc:/network/rpc/bind:default
online         Aug_24   svc:/milestone/name-services:default
```

```
online         Aug_24   svc:/system/filesystem/local:default
online         Aug_24   svc:/system/mdmonitor:default
```

Managing SMF Services

Using RBAC Rights Profiles With SMF

You can use RBAC rights profiles to allow users to manage some of the SMF services, without having to give the user `root` access. The rights profiles define what commands the user can run. For SMF, the following profiles have been created:

- `Service Management` — User can add, delete or modify services.
- `Service Operator` — User can request state changes of any service instance, such as restart and refresh.

For specific information about the authorizations, see the `smf_security`(5) man page. For instructions to assign a rights profile, see "How to Change the RBAC Properties of a User" in *System Administration Guide: Security Services*.

▼ How to Disable a Service Instance

Use the following procedure to disable a service. The service status change is recorded in the service configuration repository. Once the service is disabled, the disabled state will persist across reboots. The only way to get the service running again is to enable it.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Check the dependents of the service you want to disable.**

   If this service has dependents that you need, then you cannot disable this service.

   ```
   # svcs -D FMRI
   ```

3. **Disable the service.**

   ```
   # svcadm disable FMRI
   ```

Example 14–7 Disabling the `rlogin` Service

The output from the first command shows that the `rlogin` service has no dependents. The second command in this example disables the `rlogin` service. The third command shows that the state of the `rlogin` service instance is disabled.

```
# svcs -D network/login:rlogin
# svcadm disable network/login:rlogin
STATE          STIME    FMRI
# svcs network/login:rlogin
STATE          STIME    FMRI
disabled         11:17:24 svc:/network/login:rlogin
```

▼ How to Enable a Service Instance

Use the following procedure to enable a service. The service status change is recorded in the service configuration repository. Once the service is enabled, the enabled state will persist across system reboots if the service dependencies are met.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Determine whether service dependencies are satisfied.**

   If the service is enabled, then the service dependencies are satisfied. If not, use `svcadm enable -r` FMRI to recursively enable all dependencies.

   ```
   # svcs -l FMRI|grep enabled
   ```

3. **Enable a service.**

   ```
   # svcadm enable FMRI
   ```

Example 14–8 Enabling the `rlogin` Service

The first command in this example enables the `rlogin` service. The second command shows that the state of the `rlogin` service instance is online.

```
# svcs -l network/login:rlogin|grep enabled
enabled      true
# svcadm enable network/login:rlogin
# svcs network/login:rlogin
STATE          STIME    FMRI
online         12:09:16 svc:/network/login:rlogin
```

▼ How to Restart a Service

If a service is currently running but needs to be restarted due to a configuration change or some other reason, the service can be restarted without you having to type separate commands to stop and start the service. The only reason to specifically disable and then enable a service is if changes need to be made before the service is enabled, and after the service is disabled.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Restart a service.**

   ```
   # svcadm restart FMRI
   ```

▼ How to Restore a Service That Is in the Maintenance State

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Determine if any process that are dependent to the service have not stopped.**

   Normally, when a service instance is in a maintenance state, all processes associated with that instance have stopped. However, you should make sure before you proceed. The following command lists all of the processes that are associated with a service instance as well as the PIDs for those processes.

   ```
   # svcs -p FMRI
   ```

3. **(Optional) Kill any remaining processes.**

   Repeat this step for all processes that are displayed by the `svcs` command.

   ```
   # pkill -9 PID
   ```

4. **If necessary, repair the service configuration.**

   Consult the appropriate service log files in `/var/service/log` for a list of errors.

5. **Restore the service.**

   ```
   # svcadm clear FMRI
   ```

▼ How to Revert to Another SMF Snapshot

If the service configuration is wrong, the problem can be fixed by reverting to the last snapshot that started successfully. In this procedure, a previous snapshot of the `console-login` service is used.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Run the `svccfg` command.**

   ```
   # svccfg
   svc:>
   ```

   a. **Select the service instance that you want to fix.**

   Note – You must use an FMRI that fully defines the instance. No shortcuts are allowed.

   ```
   svc:> select system/console-login:default
   svc:/system/console-login:default>
   ```

   b. **Generate a list of available snapshots.**

   ```
   svc:/system/console-login:default> listsnap
   initial
   running
   start
   svc:/system/console-login:default>
   ```

   c. **Select to revert to the `start` snapshot.**

   The `start` snapshot is the last snapshot in which the service successfully started.

   ```
   svc:/system/console-login:default> revert start
   svc:/system/console-login:default>
   ```

   d. **Quit `svccfg`.**

   ```
   svc:/system/console-login:default> quit
   #
   ```

3. **Update the information in the service configuration repository.**

   This step updates the repository with the configuration information from the `start` snapshot.

   ```
   # svcadm refresh system/console-login
   ```

4. **Restart the service instance.**

```
# svcadm restart system/console-login
```

▼ How to Use a Different SMF Profile

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Select SMF profile to use.**

   In this example, the `generic_limited_net.xml` profile is used.

```
# svccfg apply /var/svc/profile/generic_limited_net.xml
```

Configuring SMF Services

▼ How to Modify a Service

The following procedure shows how to change the configuration of a service that is not managed by the `inetd` service.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Make changes to the configuration files, as needed.**

   Many of the services have one or more configuration files that are used to define the startup or other configuration information. These files can be changed while the service is running. The contents of the files is only checked when the service is started.

3. **Restart the service.**

```
# svcadm restart FMRI
```

Example 14–9 Sharing an NFS File System

To share a file system using the NFS service, you must define the file system in the `/etc/dfs/dfstab` file and then restart the NFS service. This example shows you what the `dfstab` file could look like, as well as how to restart the service.

```
# cat /etc/dfs/dfstab
 .
 .
share -F nfs -o rw /export/home
# svcadm restart svc:/network/nfs/server
```

▼ How to Change an Environment Variable for a Service

This procedure shows how to modify `cron` environment variables to help with debugging.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Verify that the service is running.**

   ```
   # svcs system/cron
   STATE          STIME    FMRI
   online         Dec_04   svc:/system/cron:default
   ```

3. **Set environment variables.**

   In this example the `UMEM_DEBUG` and `LD_PRELOAD` environment variables are set. For information about the `setenv` subcommand refer to the `svccfg`(1M) man page.

   ```
   # svccfg -s system/cron:default setenv UMEM_DEBUG default
   # svccfg -s system/cron:default setenv LD_PRELOAD libumem.so
   ```

4. **Refresh and restart the service.**

   ```
   # svcadm refresh system/cron
   # svcadm restart system/cron
   ```

5. **Verify that the change has been made.**

   ```
   # pargs -e `pgrep -f /usr/sbin/cron`
   100657: /usr/sbin/cron
   envp[0]: LOGNAME=root
   envp[1]: LD_PRELOAD=libumem.so
   ```

```
envp[2]: PATH=/usr/sbin:/usr/bin
envp[3]: SMF_FMRI=svc:/system/cron:default
envp[4]: SMF_METHOD=/lib/svc/method/svc-cron
envp[5]: SMF_RESTARTER=svc:/system/svc/restarter:default
envp[6]: TZ=GB
envp[7]: UMEM_DEBUG=default
#
```

▼ How to Change a Property for an `inetd` Controlled Service

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **List the properties for the specific service.**

   This command displays all of the properties for the service identified by the FMRI.

   ```
   # inetadm -l FMRI
   ```

3. **Change the property for the service.**

   Each property for an `inetd` controlled service is defined by a property name and an assigned value. Supplying the property name without a specified value resets the property to the default value. Specific information about the properties for a service should be covered in the man page associated with the service.

   ```
   # inetadm -m FMRI  property-name=value
   ```

4. **Verify that the property has changed.**

   List the properties again to make sure that the appropriate change has occurred.

   ```
   # inetadm -l FMRI
   ```

5. **Confirm that the change has taken effect.**

   Confirm the property change that the change has the desired effect.

Example 14–10 Changing the `tcp_trace` Property for `telnet`

The following example shows how to set the `tcp_trace` property for `telnet` to `true`. Checking the `syslog` output after running a `telnet` command shows that the change has taken effect.

```
# inetadm -l svc:/network/telnet:default
SCOPE    NAME=VALUE
         name="telnet"
```

```
 .
 .
default  inherit_env=TRUE
default  tcp_trace=FALSE
default  tcp_wrappers=FALSE
# inetadm -m svc:/network/telnet:default tcp_trace=TRUE
# inetadm -l svc:/network/telnet:default
SCOPE    NAME=VALUE
         name="telnet"
 .
 .
default  inherit_env=TRUE
         tcp_trace=TRUE
default  tcp_wrappers=FALSE
# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
login: root
Password:
Last login: Mon Jun 21 05:55:45 on console
Sun Microsystems Inc.   SunOS 5.10      s10_57  May 2004
# ^D
Connection to localhost closed by foreign host.
# tail -1 /var/adm/messages
Jun 21 06:04:57 yellow-19 inetd[100308]: [ID 317013 daemon.notice] telnet[100625]
    from 127.0.0.1 32802
```

▼ How to Modify a Command-Line Argument for an `inetd` Controlled Service

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **List the `exec` property for the specific service.**

   This command displays all the properties for the service identified by the FMRI. Adding the `grep` command restricts the output to the `exec` property for the service.

   ```
   # inetadm -l FMRI|grep exec
   ```

3. **Change the `exec` property for the service.**

   The *command-syntax* set with the `exec` property defines the command string that is run when the service is started.

   ```
   # inetadm -m FMRI exec="command-syntax"
   ```

4. **Verify that the property has changed.**

   List the properties again to make sure that the appropriate change has occurred.

   ```
   # inetadm -l FMRI
   ```

Example 14–11 Adding the Connection Logging (`-l`) Option to the `ftp` Command

In this example, the `-l` option is added to the `ftp` daemon when it is started. The effect of this change can be seen by reviewing the `syslog` output after a `ftp` login session has been completed.

```
# inetadm -l svc:/network/ftp:default | grep exec
        exec="/usr/sbin/in.ftpd -a"
# inetadm -m svc:/network/ftp:default exec="/usr/sbin/in.ftpd -a -l"
# inetadm -l svc:/network/ftp:default
SCOPE    NAME=VALUE
         name="ftp"
         endpoint_type="stream"
         proto="tcp6"
         isrpc=FALSE
         wait=FALSE
         exec="/usr/sbin/in.ftpd -a -l"
 .
 .
# ftp localhost
Connected to localhost.
220 yellow-19 FTP server ready.
Name (localhost:root): mylogin
331 Password required for mylogin.
Password:
230 User mylogin logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
221-You have transferred 0 bytes in 0 files.
221-Total traffic for this session was 236 bytes in 0 transfers.
221-Thank you for using the FTP service on yellow-19.
221 Goodbye.
# tail -2 /var/adm/messages
Jun 21 06:54:33 yellow-19 ftpd[100773]: [ID 124999 daemon.info] FTP LOGIN FROM localhost
     [127.0.0.1], mylogin
Jun 21 06:54:38 yellow-19 ftpd[100773]: [ID 528697 daemon.info] FTP session closed
```

▼ How to Convert `inetd.conf` Entries

The following procedure converts `inetd.conf` entries into SMF service manifests. This procedure needs to be run anytime a third-party application that depends on `inetd` is added to a system. Also run this procedure, if you need to make configuration changes to the entry in `/etc/inetd.conf`.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Convert the `inetd.conf` entries.**

   The `inetconv` command converts each entry in the selected file into service manifests.

   ```
   # inetconv -i filename
   ```

Example 14–12 Converting `/etc/inet/inetd.conf` Entries into SMF Service Manifests

```
# inetconv -i /etc/inet/inetd.conf
```

Using Run Control Scripts (Task Map)

Task	Description	For Instructions
Stop or start a service.	Use a run control script to stop or start a service.	"How to Use a Run Control Script to Stop or Start a Legacy Service" on page 242
Add a run control script.	Create a run control script and add it to the `/etc/init.d` directory.	"How to Add a Run Control Script" on page 243
Disable a run control script.	Disable a run control script by renaming the file.	"How to Disable a Run Control Script" on page 244

Using Run Control Scripts

▼ How to Use a Run Control Script to Stop or Start a Legacy Service

One advantage of having individual scripts for each run level is that you can run scripts in the `/etc/init.d` directory individually to stop system services without changing a system's run level.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Stop the system service.**

   # **/etc/init.d/***filename* **stop**

3. **Restart the system service.**

   # **/etc/init.d/***filename* **start**

4. **Verify that the service has been stopped or started.**

   # **pgrep -f** *service*

Example 14–13 Using a Run Control Script to Stop or Start a Service

For example, you can stop the NFS server daemons by typing the following:

```
# /etc/init.d/nfs.server stop
# pgrep -f nfs
```

Then, you can restart the NFS server daemons by typing the following:

```
# /etc/init.d/nfs.server start
# pgrep -f nfs
101773
101750
102053
101748
101793
102114
# pgrep -f nfs -d, | xargs ps -fp
     UID    PID   PPID   C    STIME TTY         TIME CMD
  daemon 101748      1   0   Sep 01 ?           0:06 /usr/lib/nfs/nfsmapid
  daemon 101750      1   0   Sep 01 ?          26:27 /usr/lib/nfs/lockd
  daemon 101773      1   0   Sep 01 ?           5:27 /usr/lib/nfs/statd
    root 101793      1   0   Sep 01 ?          19:42 /usr/lib/nfs/mountd
  daemon 102053      1   0   Sep 01 ?        2270:37 /usr/lib/nfs/nfsd
  daemon 102114      1   0   Sep 01 ?           0:35 /usr/lib/nfs/nfs4cbd
```

▼ How to Add a Run Control Script

If you want to add a run control script to start and stop a service, copy the script into the `/etc/init.d` directory. Then, create links in the `rc`*n*`.d` directory where you want the service to start and stop.

See the `README` file in each `/etc/rc`*n*`.d` directory for more information on naming run control scripts. The following procedure describes how to add a run control script.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Add the script to the `/etc/init.d` directory.**

   ```
   # cp filename /etc/init.d
   # chmod 0744 /etc/init.d/filename
   # chown root:sys /etc/init.d/filename
   ```

3. **Create links to the appropriate `rc`*n*`.d` directory.**

   ```
   # cd /etc/init.d
   # ln filename /etc/rc2.d/Snnfilename
   # ln filename /etc/rcn.d/Knnfilename
   ```

4. **Verify that the script has links in the specified directories.**

   ```
   # ls /etc/init.d/*filename /etc/rc2.d/*filename /etc/rcn.d/*filename
   ```

Example 14–14 Adding a Run Control Script

The following example shows how to add a run control script for the `xyz` service.

```
# cp xyz /etc/init.d
# chmod 0744 /etc/init.d/xyz
# chown root:sys /etc/init.d/xyz
# cd /etc/init.d
# ln xyz /etc/rc2.d/S99xyz
# ln xyz /etc/rc0.d/K99xyz
# ls /etc/init.d/*xyz /etc/rc2.d/*xyz /etc/rc0.d/*xyz
```

▼ How to Disable a Run Control Script

You can disable a run control script by renaming it with an underscore (_) at the beginning of the file name. Files that begin with an underscore or dot are not executed. If you copy a file by adding a suffix to it, both files will be run.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Rename the script by adding an underscore (_) to the beginning of the new file.**

   ```
   # cd /etc/rcn.d
   # mv filename _filename
   ```

3. **Verify that the script has been renamed.**

```
# ls _*
_filename
```

Example 14–15 Disabling a Run Control Script

The following example shows how to rename the `S99datainit` script.

```
# cd /etc/rc2.d
# mv S99datainit _S99datainit
# ls _*
_S99datainit
```

Troubleshooting the Service Management Facility

▼ How to Repair a Corrupt Repository

This procedure shows how to replace a corrupted repository with a default copy of the repository. The following message is displayed if the configuration repository is corrupt:

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Stop the `svc.startd` daemon.**

   This command finds the PID for `svc.startd` and then stops the process.

   ```
   # pstop `pgrep svc.startd`
   ```

3. **Stop the `svc.configd` daemon.**

   ```
   # pkill svc.configd
   ```

4. **(Optional) Save the current repository for debugging.**

   ```
   # cp /etc/svc/repository.db /etc/svc/repository.bad
   ```

5. **Copy the default repository.**

   ```
   # cp /lib/svc/seed/global.db /etc/svc/repository.db
   ```

6. **Reboot the system to restart all of the services.**

```
# reboot
```

Example 14–16 Repairing a Corrupt Repository in a Non-Global Zone

When fixing a repository in a non-global zone, in step 5 you would want to use the following command:

```
# cp /lib/svc/seed/nonglobal.db /etc/svc/repository.db
```

▼ How to Start Services Interactively During Boot

If problems with starting services occur, sometimes a system will hang during the boot. This procedure shows how to troubleshoot this problem.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Boot without starting any milestones.**

   This command instructs the `svc.startd` daemon to temporarily disable all services and start `sulogin` on the console.

   ```
   ok boot -m milestone=none
   ```

3. **Log in to the system as `root`.**

4. **Enable all services.**

   ```
   # svcadm milestone -t all
   ```

5. **Determine where the boot process is hanging.**

   When the boot process hangs, determine which services are not running by running `svcs -l`. Look for error messages in the log files in `/var/svc/log`.

6. **After fixing the problems, verify that all services are ready to be started.**

   a. **Verify that all needed services are online.**

      ```
      # svcs -l
      ```

   b. **Verify that the `console-login` service dependencies are satisfied.**

      This command verifies that the `login` process on the console will run.

      ```
      # svcs -l system/console-login:default
      ```

7. **Continue the normal booting process.**

▼ Debugging a Service That Is Not Starting

In this procedure, the print service is disabled.

Steps

1. **Become superuser or assume a role that includes the `Service Management` rights profile.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC" in *System Administration Guide: Security Services* in System Administration Guide: Security Services.

2. **Request information about the hung service.**

```
# svcs -xv
svc:/application/print/server:default (LP Print Service)
 State: disabled since Wed 13 Oct 2004 02:20:37 PM PDT
Reason: Disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-05
   See: man -M /usr/share/man -s 1M lpsched
Impact: 2 services are not running:
        svc:/application/print/rfc1179:default
        svc:/application/print/ipp-listener:default
```

   The `-x` option provides additional information about the service instances that are impacted.

3. **Enable the service.**

```
# svcadm enable application/print/server
```

CHAPTER 15

Managing Software (Overview)

Software management involves adding and removing software from stand-alone systems, servers, and their clients. This chapter describes the various tools available for installing and managing software.

This chapter does not describe installing the Solaris Operating System (Solaris OS) on a new system, nor does it describe installing or upgrading a new version of the Solaris OS. For information on installing or upgrading the Solaris OS, see *Solaris 10 Installation Guide: Basic Installations*.

This is a list of the overview information in this chapter.

For step-by-step instructions on managing software, see Chapter 16 and Chapter 17.

For information about managing software in the Solaris zones environment, see Chapter 24, "Adding and Removing Packages and Patches in a Solaris Zones Environment (Tasks)," in *System Administration Guide: Solaris Containers—Resource Management and Solaris Zones*.

What's New in Software Management?

This section describes the new software management features in this Solaris release.

Package and Patch Tool Enhancements

The Solaris package and patch tools have been enhanced, providing improved performance and extended functionality.

As a part of these enhancements, the `pkgchk` command now provides a new option to assist you in mapping files to packages. To map files to packages, use the `pkgchk -P` option instead of `grep` *pattern* `/var/sadm/install/contents`. The `-P` option enables you to use a partial path. Use this option with the `-l` option to list the information about the files that contain the partial path. For more information see "How to Check the Integrity of Installed Objects (`pkgchk -p, pkgchk -P`)" on page 304 and the `pkgchk`(1M) man page.

Sun Patch Manager Enhancements

The Sun Patch Manager tool (Patch Manager) is the standard tool for managing patches on Solaris systems. Use this tool to apply patches to Solaris systems.

You can access Patch Manager by using a graphical user interface or by using the `smpatch` command-line interface.

Patch Manager has been enhanced with these features:

- **PatchPro analysis engine –** Patch Manager now incorporates PatchPro functionality to automate the patch management process. This process includes performing patch analysis on systems, then downloading and applying the resulting patches. This automation functionality was previously available for Solaris 9 as a separate PatchPro product and is now part of the standard Solaris release.
- **Local-mode command-line interface –** The command-line interface, `smpatch`, can be used even when the Solaris WBEM services are not running on your system. This capability enables you to use `smpatch` to apply patches while your system is in single-user mode.
- **Patch list operations –** Patch Manager enables you to generate, save, edit, order, and resolve patch lists. These lists can be used to perform patch operations, such as downloading and applying patches.

You must install at least the Developer Software Support Group of Solaris 10 to use Sun Patch Manager.

For more information about how to use Patch Manager, see Chapter 19 and the `smpatch`(1M) man page.

Where to Find Software Management Tasks

Use this table to find step-by-step instructions for managing software.

Software Management Topics	For More Information
Installing Solaris software	*Solaris 10 Installation Guide: Basic Installations*
Adding or removing Solaris software packages after installation	Chapter 16 and Chapter 17
Adding or removing Solaris patches after installation	Chapter 19 and Chapter 20
Troubleshooting software package problems	Chapter 29, "Troubleshooting Software Package Problems (Tasks)," in *System Administration Guide: Advanced Administration*

Overview of Software Packages

Software management involves installing or removing software products. Sun and its third-party vendors deliver software products in a form called a *package*.

The term *packaging* generically refers to the method for distributing and installing software products to systems where the products will be used. A package is a collection of files and directories in a defined format. This format conforms to the application binary interface (ABI), which is a supplement to the System V Interface Definition. The Solaris OS provides a set of utilities that interpret this format and provide the means to install a package, to remove a package, or to verify a package installation.

A *patch* is a collection of files and directories that replace or update existing files and directories that are preventing proper execution of the existing software. For more information about patches, see Chapter 18.

Signed Packages and Patches

Packages can include a digital signature. A package with a valid digital signature ensures that the package has not been modified since the signature was applied to the package. Using signed packages is a secure method of downloading or adding packages because the digital signature can be verified before the package is added to your system.

The same holds true for signed patches. A patch with a valid digital signature ensures that the patch has not been modified since the signature was applied to the patch. Using signed patches is a secure method of downloading or applying patches because the digital signature can be verified before the patch is applied to your system.

For more information about *applying* signed patches to your system, see "Managing Solaris Patches by Using the Sun Patch Manager Command-Line Interface (Task Map)" on page 331 or "Managing Solaris Patches by Using the `patchadd` Command (Task Map)" on page 357.

For information about *creating* signed packages, see *Application Packaging Developer's Guide*.

A signed package is identical to an unsigned package, except for the digital signature. The package can be installed, queried, or removed with existing Solaris packaging tools. A signed package is also binary-compatible with an unsigned package.

Before you can use `pkgadd` and `patchadd` to add a package or patch with a digital signature to your system, you must set up a package keystore with trusted certificates. These certificates are used to identify that the digital signature on the package or patch is valid. Note that the keystore and certificates are automatically set up when you use Patch Manager to apply signed patches.

The following describes the general terms associated with signed packages and patches.

Keystore — A repository of certificates and keys that is queried when needed.

- Java keystore – A repository of certificates that is installed by default with the Solaris release. The Java keystore is usually stored in the `/usr/j2se/jre/lib/security` directory.
- Package keystore – A repository of certificates that you import when adding signed packages and patches to your system.

  The package keystore is stored in the `/var/sadm/security` directory by default.

Trusted certificate — A certificate that holds a public key that belongs to another entity. The *trusted certificate* is named as such because the keystore owner trusts that the public key in the certificate

	indeed belongs to the identity identified by the subject or owner of the certificate. The issuer of the certificate vouches for this trust by signing the certificate. Trusted certificates are used when verifying signatures, and when initiating a connection to a secure (SSL) server.
User key	Holds sensitive cryptographic key information. This information is stored in a protected format to prevent unauthorized access. A user key consists of both the user's private key and the public key certificate that corresponds to the private key.

The process of using the `pkgadd` or `patchadd` command to add a signed package or patch to your system involves three basic steps:

1. Adding the certificates to your system's package keystore by using the `pkgadm` command
2. (Optional) Listing the certificates by using the `pkgadm` command
3. Adding the package with the `pkgadd` command or applying the patch by using the `patchadd` command

If you use Patch Manager to apply patches to your system, you do not need to manually set up the keystore and certificates, as it is automatically set up.

For step-by-step instructions on adding signed packages to your system, see "Adding and Removing Signed Packages by Using the `pkgadd` Command (Task Map)" on page 289.

For step-by-step instructions on applying signed patches to your system, see "Managing Solaris Patches by Using the Sun Patch Manager Command-Line Interface (Task Map)" on page 331, or "Managing Solaris Patches by Using the `patchadd` Command (Task Map)" on page 357.

Using Sun's Certificates to Verify Signed Packages and Patches

Digital certificates, issued and authenticated by Sun Microsystems, are used to verify that the downloaded package or patch with the digital signature has not been compromised. These certificates are imported into your system's package keystore.

A *stream-formatted* SVR4-signed package or patch contains an embedded PEM-encoded PKCS7 signature. This signature contains at a minimum the encrypted digest of the package or patch, along with the signer's X.509 public key certificate. The package or patch can also contain a *certificate chain* that is used to form a chain of trust from the signer's certificate to a locally stored trusted certificate.

The PEM-encoded PKCS7 signature is used to verify the following information:

- The package came from the entity that signed it.
- The entity indeed signed it.
- The package hasn't been modified since the entity signed it.
- The entity that signed it is a trusted entity.

All Sun certificates are issued by Baltimore Technologies, which recently bought GTE CyberTrust.

Access to a package keystore is protected by a special password that you specify when you import the Sun certificates into your system's package keystore.

If you use the `pkgadm listcert` command, you can view information about your locally stored certificates in the package keystore. For example:

```
# pkgadm listcert -P pass:store-pass
    Keystore Alias: GTE CyberTrust Root
       Common Name: GTE CyberTrust Root
  Certificate Type: Trusted Certificate
Issuer Common Name: GTE CyberTrust Root
    Validity Dates: <Feb 23 23:01:00 1996 GMT> - <Feb 23 23:59:00 2006 GMT>
   MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
  SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91:BC:65:A6...
```

The following describes the output of the `pkgadm listcert` command.

Keystore Alias — When you retrieve certificates for printing, signing, or removing, this name must be used to reference the certificate.

Command Name — The common name of the certificate. For trusted certificates, this name is the same as the keystore alias.

Certificate Type — Can be one of two types:

- Trusted certificate – A certificate that can be used as a trust anchor when verifying other certificates. No private key is associated with a trusted certificate.
- Signing certificate – A certificate that can be used when signing a package or patch. A private key is associated with a signing certificate.

Issuer Command Name — The name of the entity that issued, and therefore signed, this certificate. For trusted certificate authority (CA) certificates, the issuer common name and common name are the same.

Validity Dates — A date range that identifies when the certificate is valid.

MD5 Fingerprint — An MD5 digest of the certificate. This digest can be used to verify that the certificate has not been altered during transmission from the source of the certificate.

SHA1 Fingerprint	Similar to an MD5 fingerprint, except that it is calculated using a different algorithm.

Each certificate is authenticated by comparing its MD5 and SHA1 hashes, also called *fingerprints*, against the known correct fingerprints published by the issuer.

Importing Sun's Trusted Certificates

You can obtain Sun's trusted certificates for adding signed packages and patches in the following ways:

- **Java keystore** – Import Sun's Root CA certificate that is included by default in the Java keystore when you install the Solaris release.
- **Sun's Public Key Infrastructure (PKI) site** – If you do not have a Java keystore available on your system, you can import the certificates from this site.

  https://ra.sun.com:11005/
- **PatchPro's keystore** – If you have installed PatchPro for applying signed patches with the `smpatch` command, you can import Sun's Root CA certificate from the Java keystore.

Setting Up a Package Keystore

In previous Solaris releases, you could download the patch management tools and create a Java keystore, for use by PatchPro, by importing the certificates with the `keytool` command.

If your system already has a populated Java keystore, you can now export the Sun Microsystems root CA certificate from the Java keystore with the `keytool` command. Then, use the `pkgadm` command to import this certificate into the package keystore.

After the Root CA certificate is imported into the package keystore, you can use the `pkgadd` and `patchadd` commands to add signed packages and patches to your system.

Note – The Sun Microsystems root-level certificates are only required when adding Sun-signed patches and packages.

For step-by-step instructions on importing certificates into the package keystore, see "How to Import a Trusted Certificate From the Java Keystore (`pkgadm addcert`)" on page 290.

For complete instructions on adding signed packages with the `pkgadd` command, see "How to Add a Signed Package (`pkgadd`)" on page 294.

Tools for Managing Software Packages

The following table describes the tools for adding and removing software packages from a system after the Solaris release is installed on a system.

TABLE 15–1 Tools or Commands for Managing Software Packages

Tool or Command	Description	Man Page
`installer`	Launches an installer, such as Solaris installation GUI, to add software from the Solaris media. The installer must be available either locally or remotely.	`installer`(1M)
`prodreg (GUI)`	Launches an installer to add, remove, or display software product information. Use Solaris Product Registry to remove or display information about software products that were originally installed by using the Solaris installation GUI or the Solaris `pkgadd` command.	`prodreg`(1M)
Solaris Product Registry `prodreg` Viewer (CLI)	Use the `prodreg` command to remove or display information about software products that were originally installed by using the Solaris installation GUI or the Solaris `pkgadd` command.	`prodreg`(1M)
`pkgadd`	Installs a signed or unsigned software package.	`pkgadd`(1M)
`pkgadm`	Maintains the keys and certificates used to manage signed packages and signed patches.	`pkgadm`(1M)
`pkgchk`	Checks the installation of a software package.	`pkgchk`(1M)
`pkginfo`	Lists software package information.	`pkginfo`(1)

TABLE 15–1 Tools or Commands for Managing Software Packages *(Continued)*

Tool or Command	Description	Man Page
`pkgparam`	Displays software package parameter values.	`pkgparam`(1)
`pkgrm`	Removes a software package.	`pkgrm`(1M)
`pkgtrans`	Translates an installable package from one format to another format. The `-g` option instructs the `pkgtrans` command to generate and store a signature in the resulting data stream.	`pkgtrans`(1)

For more information about these commands, see Chapter 16 and Chapter 17.

Adding or Removing a Software Package (`pkgadd`)

All the software management tools that are listed in Table 15–1 are used to add, remove, or query information about installed software. The Solaris Product Registry `prodreg` viewer and the Solaris installation GUI both access install data that is stored in the Solaris Product Registry. The package tools, such as the `pkgadd` and `pkgrm` commands, also access or modify install data.

When you add a package, the `pkgadd` command uncompresses and copies files from the installation media to a local system's disk. When you remove a package, the `pkgrm` command deletes all files associated with that package, unless those files are also shared with other packages.

Package files are delivered in package format and are unusable as they are delivered. The `pkgadd` command interprets the software package's control files, and then uncompresses and installs the product files onto the system's local disk.

Although the `pkgadd` and `pkgrm` commands do not log their output to a standard location, they do keep track of the package that is installed or removed. The `pkgadd` and `pkgrm` commands store information about a package that has been installed or removed in a software product database.

By updating this database, the `pkgadd` and `pkgrm` commands keep a record of all software products installed on the system.

Key Points for Adding Software Packages (pkgadd)

Keep the following key points in mind before you install or remove packages on your system:

- **Package naming conventions** – Sun packages always begin with the prefix SUNW, as in SUNWaccr, SUNWadmap, and SUNWcsu. Third-party packages usually begin with a prefix that corresponds to the company's stock symbol.
- **What software is already installed** – You can use the Solaris installation GUI, Solaris Product Registry prodreg viewer (either GUI or CLI) or the pkginfo command to determine the software that is already installed on a system.
- **How servers and clients share software** – Clients might have software that resides partially on a server and partially on the client. In such cases, adding software for the client requires that you add packages to both the server and the client.

Guidelines for Removing Packages (pkgrm)

You should use one of the tools listed in Table 15–1 to remove a package, even though you might be tempted to use the rm command instead. For example, you could use the rm command to remove a binary executable file. However, doing so is not the same as using the pkgrm command to remove the software package that includes that binary executable. Using the rm command to remove a package's files will corrupt the software products database. If you really only want to remove one file, you can use the removef command. This command will update the software product database correctly so that the file is no longer a part of the package. For more information, see the removef(1M) man page.

If you intend to keep multiple versions of a package, install new versions into a different directory than the already installed package by using the pkgadd command. For example, if you intended to keep multiple versions of a document processing application. The directory where a package is installed is referred to as the base directory. You can manipulate the base directory by setting the basedir keyword in a special file called an administration file. For more information on using an *administration file* and on setting the base directory, see the "Avoiding User Interaction When Adding Packages (pkgadd)" on page 259 and admin(4) man page.

Note – If you use the upgrade option when installing Solaris software, the Solaris installation software checks the software product database to determine the products that are already installed on the system.

Avoiding User Interaction When Adding Packages (pkgadd)

Using an Administration File

When you use the `pkgadd -a` command, the command consults a special administration file for information about how the installation should proceed. Normally, the `pkgadd` command performs several checks and prompts the user for confirmation before it actually adds the specified package. You can, however, create an administration file that indicates to the `pkgadd` command that it should bypass these checks and install the package without user confirmation.

The `pkgadd` command, by default, checks the current working directory for an administration file. If the `pkgadd` command doesn't find an administration file in the current working directory, it checks the `/var/sadm/install/admin` directory for the specified administration file. The `pkgadd` command also accepts an absolute path to the administration file.

Note – Use administration files judiciously. You should know where a package's files are installed and how a package's installation scripts run before using an administration file to avoid the checks and prompts that the `pkgadd` command normally provides.

The following example shows an administration file that prevents the `pkgadd` command from prompting the user for confirmation before installing the package.

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
```

```
setuid=nocheck
conflict=nocheck
action=nocheck
networktimeout=60
networkretries=3
authentication=quit
keystore=/var/sadm/security
proxy=
basedir=default
```

Besides using administration files to avoid user interaction when you add packages, you can use them in several other ways. For example, you can use an administration file to quit a package installation (without user interaction) if there's an error or to avoid interaction when you remove packages by using the `pkgrm` command.

You can also assign a special installation directory for a package, which you might do if you wanted to maintain multiple versions of a package on a system. To do so, set an alternate base directory in the administration file by using the `basedir` keyword. The keyword specifies where the package will be installed. For more information, see the `admin`(4) man page.

Using a Response File (`pkgadd`)

A response file contains your answers to specific questions that are asked by an *interactive package*. An interactive package includes a `request` script that asks you questions prior to package installation, such as whether optional pieces of the package should be installed.

If you know prior to installation that the package is an interactive package, and you want to store your answers to prevent user interaction during future installations, use the `pkgask` command to save your response. For more information on this command, see `pkgask`(1M).

Once you have stored your responses to the questions asked by the `request` script, you can use the `pkgadd -r` command to install the package without user interaction.

CHAPTER 16

Managing Software With Solaris System Administration Tools (Tasks)

This chapter describes how to add, verify, and remove software packages by using the Solaris installation graphical user interface (GUI) and the Solaris Product Registry.

For information on the procedures associated with performing software management tasks, see:

- "Adding Software With the Solaris Installation GUI" on page 262
- "Managing Software With the Solaris Product Registry GUI (Task Map)" on page 264
- "Managing Software With the Solaris Product Registry Command-Line Interface (Task Map)" on page 268

Solaris Product Registry and Solaris GUI Installation Tools for Managing Software

The following table lists the commands to use for adding, removing, and checking the installation of software packages the Solaris installation GUI and Solaris Package Registry tools.

TABLE 16–1 System Administration Tools for Managing Software Packages

Tool	Description	Man Page
`installer`	Installs or removes a software package with an installer	`installer`(1M)

TABLE 16–1 System Administration Tools for Managing Software Packages *(Continued)*

Tool	Description	Man Page
`prodreg`	Enables you to browse, unregister, and uninstall software in the Solaris Product Registry	`prodreg`(1M)

Adding Software With the Solaris Installation GUI

This section describes how to use the Solaris installation GUI to add software to a system on which you have installed the Solaris Operating System (Solaris OS). The Solaris installation GUI installs only the components of the software groups that you skipped when you initially installed the Solaris OS. You cannot upgrade to another software group after installing or upgrading the OS. For a description of the four software groups, see "System Requirements and Recommendations" in *Solaris 10 Installation Guide: Custom JumpStart and Advanced Installations*.

▼ How to Install Software With the Solaris Installation GUI Program

Note – This procedure assumes that the system is running volume management (`vold`). If your system is not running volume management, see Chapter 2, "Accessing Removable Media (Tasks)," in *System Administration Guide: Devices and File Systems*. This chapter provides information about accessing removable media without volume management.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Decide to install software from a CD, a DVD, or from the network. Select one of the following:**

   - If you are installing from a CD, insert the CD into the CD-ROM drive.

     If you insert the Solaris 10 Languages CD, the Solaris installation GUI starts automatically. Proceed to Step 5.

- If you are installing from a DVD, insert the DVD into the DVD-ROM drive.
- If you are installing from the network, locate the net image of the software you want to install.

3. **Change directories to find the Solaris installation GUI installer.**

   Solaris installation GUI installers are located in various directories on the CDs and on the DVD.

   - Solaris 10 Software CDs or DVD.
   - Solaris 10 Documentation DVD.
   - Solaris 10 Languages CD. The Solaris installation GUI starts automatically when the CD is inserted.

4. **Follow the instructions to install the software.**

   - From the command line, type the following command

     % ./installer [*options*]

     `-nodisplay` Runs the installer without a GUI.

     `-noconsole` Runs without any interactive text console device. Use this option with the `-nodisplay` option when you include the `installer` command in a UNIX script for installing software.

   - From a file manager, double-click Installer or installer.

     An Installer window is displayed, followed by the Solaris installation GUI dialog box.

5. **Follow the directions on the screen to install the software.**

6. **When you have finished adding software, click Exit.**

   The Solaris installation GUI exits.

Managing Software With the Solaris Product Registry GUI (Task Map)

The following task map describes the software management tasks that you can perform with the Solaris Product Registry.

Task	Description	For Instructions
View installed or uninstalled software with the Solaris Product Registry.	Used for learning about installed or uninstalled software.	"How to View Installed or Uninstalled Software Information With the Solaris Product Registry GUI" on page 266
Install software with the Solaris Product Registry.	You can use the Solaris Product Registry to find software and launch the Solaris installation GUI. This program takes you through the installation of that software.	"How to Install Software With the Solaris Product Registry GUI" on page 267
Uninstall software with the Solaris Product Registry.	Use tor uninstall software with the Solaris Product Registry.	"How to Uninstall Software With the Solaris Product Registry GUI" on page 268

The Solaris Product Registry is a tool to help you manage installed software. After you have installed the software, Product Registry provides a list of all the installed software by using the Solaris installation GUI or the Solaris `pkgadd` command.

You can use the Solaris Product Registry in a GUI or with a command-line interface (CLI). For more information on how to use the Solaris Product Registry CLI, see "Managing Software With the Solaris Product Registry Command-Line Interface (Task Map)" on page 268.

The Solaris Product Registry GUI interface enables you to do the following:

- View a list of installed and registered software and some software attributes.
- View all Solaris system products that you installed in their localized version in the System Software Localizations directory.
- Find and launch an installer.
- Install additional software products.
- Uninstall software and individual software packages.

The Solaris Product Registry GUI main window consists of three panes of information:

- Installed, registered, and removed software
- Standard attributes of the currently selected software
- Attributes that are customized and attributes that are internal to the registered software

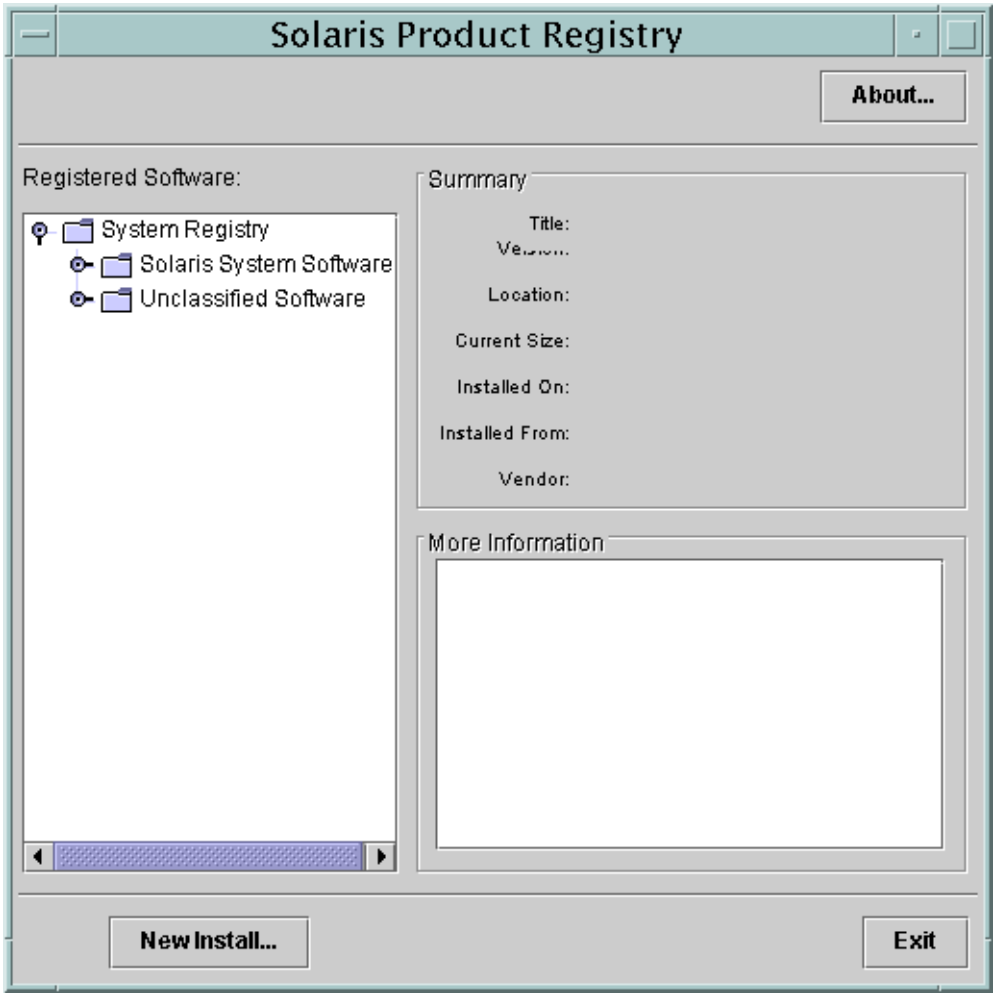

FIGURE 16–1 Solaris Product Registry Main Window

▼ How to View Installed or Uninstalled Software Information With the Solaris Product Registry GUI

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start the Solaris Product Registry tool.**

   ```
   # prodreg &
   ```

   The Solaris Product Registry main window is displayed.

3. **Click the turner control to the left of the System Registry directory in the Registered Software box.**

   The turner control changes from pointing to the right to pointing downward. You can expand or collapse any item in the registry, except an item that has a text file icon to its left.

   The Software Installed in Registered Software box always contains the following components:

   - The configuration software group that you chose when you installed the Solaris release. Software groups that can be displayed include Reduced Network Support, Core, End User System Support, Developer System Support, Entire Distribution, or Entire Distribution Plus OEM Support.
   - Additional system software, which contains Solaris products that are not part of the software group you chose.
   - Unclassified software that is not a Solaris product or part of the software group. This software includes any package that you installed by using the `pkgadd` command.

4. **Select directories until you find a software application to view.**

   The list expands as you open directories.

5. **To view the attributes, select a directory or file.**

   The Product Registry displays attribute information in the System Registry box.

   - For software products that were installed with the Solaris installation GUI, the Solaris Product Registry contains values for at least the following: Title, Version, Location, and Installed on. Items in an expanded list under a product or software group inherit the version information of the product.
   - If all or part of the product was removed with the `pkgrm` command, a cautionary icon appears next to the software product's name.

▼ How to Install Software With the Solaris Product Registry GUI

You can use Solaris Product Registry to find software and launch the Solaris installation GUI program. This program takes you through the installation of that software.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start the Solaris Product Registry tool.**

   ```
   # prodreg
   ```

   The Solaris Product Registry main window is displayed.

3. **Decide if you are installing from a CD, a DVD, or from the network. Select one of the following:**

   - If you are installing from a CD, insert the CD into the CD-ROM drive.
   - If you are installing from a DVD, insert the DVD into the DVD-ROM drive.
   - If you are installing from the network, locate the net image of the software that you want to install.

4. **To view the list of installed and registered software, click the turner control.**

5. **Click the New Install button at the bottom of the Solaris Product Registry window.**

   The Select Installer dialog box is displayed. This box initially points to the `/cdrom` directory or the directory you are in.

6. **Select directories to find the Solaris installation GUI installer.**

   Solaris installation GUI installers are located in various directories on the CDs and on the DVD.

   - Solaris 10 Software CDs or DVD.
   - Solaris 10 Documentation DVD.
   - Solaris 10 Languages CD. The Solaris installation GUI automatically starts when the CD is inserted.

7. **When you find the installer you want, select its name in the Files box.**

8. **Click OK.**

   The installer you selected is launched.

9. **Follow the directions that are displayed by the installer to install the software.**

▼ How to Uninstall Software With the Solaris Product Registry GUI

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start the Solaris Product Registry tool.**

   ```
   # prodreg
   ```

   The Solaris Product Registry main window is displayed.

3. **To view the list of installed and registered software, click the turner control.**

4. **Select directories until you find the name of the software that you want to uninstall.**

5. **Read the software attributes to make sure that this software is the software that you want to uninstall.**

6. **Click the Uninstall** *software-product-name* **button at the bottom of the Solaris Product Registry window.**

   The software product you selected is uninstalled.

Managing Software With the Solaris Product Registry Command-Line Interface (Task Map)

The following task map describes the software management tasks that you cab perform with the Solaris Product Registry command-line interface.

Task	Description	For Instructions
View installed or uninstalled software.	You can view software information by using the `browse` subcommand.	"How to View Installed or Uninstalled Software Information (`prodreg`)" on page 270

Task	Description	For Instructions
View software attributes.	You can view specific software attributes by using the `info` subcommand.	"How to View Software Attributes (`prodreg`)" on page 273
Check dependencies between software components.	You can view the components that depend on a specific software component by using the `info` subcommand.	"How to Check for Software Dependencies (`prodreg`)" on page 274
Identify damaged software products.	If you remove installed software files or packages without using the appropriate uninstaller, you can damage the software on your system.	"How to Identify Damaged Software Products (`prodreg`)" on page 276
Uninstall software	You can remove software from your system by using the `uninstall` subcommand.	"How to Uninstall Software (`prodreg`)" on page 278
Uninstall damaged software.	Uninstalling a damaged software component might fail if the uninstaller program for the software component has been removed from the system.	"How to Uninstall Damaged Software (`prodreg`)" on page 282
Reinstall damaged software components.	If other software depends on a damaged software component, you might want to reinstall the damaged component, rather than uninstall the component and the other dependent software.	"How to Reinstall Damaged Software Components (`prodreg`)" on page 285

Managing Software With the Solaris Product Registry Command-Line Interface

The `prodreg` command is the command-line interface (CLI) to the Solaris Product Registry. The `prodreg` command supports several subcommands that enable you to manage the software on your system.

You can use the `prodreg` command in a terminal window to perform the following tasks:

- View a list of installed and registered software and software attributes.

- View all Solaris system products that you installed in their localized version in the System Software Localizations directory.
- Identify damaged software.
- Remove software entries from the Solaris Product Registry.
- Uninstall software and individual software packages.

For more information on how to manage the Solaris Product Registry by using the command-line interface, see the `prodreg`(1M) man page.

▼ How to View Installed or Uninstalled Software Information (`prodreg`)

You can view information about software in the Solaris Product Registry in a terminal window by using the `browse` subcommand to the `prodreg` command.

Steps

1. **Open a terminal window.**
2. **Browse the Solaris Product Registry.**

```
% prodreg browse
   BROWSE # +/-/.  UUID                                  #  NAME
   ======== =====  ====================================  =  ============
   1        -      root                                  1  System
                                                             Registry
   2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                             System
                                                             Software
   3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software
```

The `browse` subcommand to the `prodreg` command displays the following information about registered software.

`BROWSE #` When you use the `prodreg browse` command, the Solaris Product Registry generates a *browse number* for each registered software component. This number can be used as an argument to either the `prodreg browse` command or the `info` subcommand to descend the hierarchy of specific registered components.

Note – Browse numbers might change when you reboot or reinstall your system. Do not store browse numbers in scripts or attempt to reuse them between separate login sessions.

`+/-/.` This field indicates if a software component has additional software component children registered in the Solaris Product Registry. The following characters are displayed in this field:

- `+` indicates that the software component has additional children components that are not currently displayed.
- `-` indicates that the software component has additional children components that are currently displayed.
- `.` indicates that the software component does not have children components.

`UUID` This field lists the software's unique identifier in the Solaris Product Registry.

`#` This field indicates the *instance number* of the software component on the system. If the system contains multiple instances of a software component, the Solaris Product Registry assigns a separate instance number to each instance of the component.

`NAME` This field lists the localized name of the software. The name of the Solaris OS in this sample output is the Solaris 10 system software.

3. **Browse the information for one of the software components that are listed in the Solaris Product Registry.**

   `% prodreg browse -m "`*`name`***`"`**

   The `-m` "*name*" command displays information on the software component with the name *name*.

4. **If the system contains multiple instances of** *name* **software, type the following command to browse the Solaris Product Registry:**

   `% prodreg browse -u` *`name-UUID`* **`-i`** *`instance`* **`-n`** *`number`*

   `-u` *name-UUID* Displays information on the *name* software component with the unique identifier *name-UUID*.

   `-i` *instance* Displays information on the *name* software component with the instance number *instance*.

   `-n` *number* Displays software information by referencing the component's browse number *number*.

5. **Repeat Step 3 and Step 4 for each software component that you want to browse.**

Example 16–1 Viewing Software Information by Component Name (`prodreg`)

The following example shows how to view software information by referencing the component's name.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
```

```
    1        -      root                                    1  System
                                                               Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b    1  Solaris 10
                                                               System
                                                               Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b    1  Unclassified
                                                               Software

% prodreg browse -m "Solaris 10  System Software"
```

Example 16–2 Viewing Software Information by Component Browse Number (`prodreg`)

The following example shows how to use the `-n` option with the `prodreg browse` command to view software information by referencing the component's browse number.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                    #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                    1  System
                                                               Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b    1  Solaris 10
                                                               System
                                                               Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b    1  Unclassified
                                                               Software

% prodreg browse -n 2
```

Example 16–3 Viewing Software Information by Component UUID (`prodreg`)

The following example shows how to use the `-u` option with the `prodreg browse` command to view software information by referencing the component's UUID. The UUID is the software's unique identifier in the Solaris Product Registry.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                    #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                    1  System
                                                               Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b    1  Solaris 10
                                                               System
                                                               Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b    1  Unclassified
                                                               Software

% prodreg browse -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
```

▼ How to View Software Attributes (`prodreg`)

You can view specific software attributes by using the `info` subcommand of the `prodreg` command. The `prodreg info` command displays a variety of information about registered software, including the following items:

- Software component name
- Software component description
- Required components of the software
- Other components that require the software
- Base directory of the software
- Path to the software component

Steps

1. **Open a terminal window.**
2. **Browse the Solaris Product Registry.**

```
% prodreg browse
   BROWSE # +/-/.  UUID                                  #  NAME
   ======== =====  ====================================  =  ============
   1        -      root                                  1  System
                                                             Registry
   2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                             System
                                                             Software
   3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software
```

3. **View the attributes for one of the listed software components.**

```
% prodreg info  -m "name"
```

   The `-m` "*name*" command displays the attributes of the software component with the name *name*.

4. **Repeat Step 3 for each software component you want to view.**

Example 16–4 Viewing Software Attributes by Component Name (`prodreg`)

The following example shows how to view software attributes by referencing the component's name.

```
% prodreg browse
   BROWSE # +/-/.  UUID                                  #  NAME
   ======== =====  ====================================  =  ============
   1        -      root                                  1  System
                                                             Registry
   2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                             System
                                                             Software
   3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software
```

```
% prodreg info -m "Solaris 10  System Software"
```

Example 16–5 Viewing Software Attributes by Component Browse Number (`prodreg`)

The following example shows how to use the `-n` option with the `prodreg info` command to view software attributes by referencing the component's browse number.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg info -n 2
```

Example 16–6 Viewing Software Attributes by Component UUID (`prodreg`)

The following example shows how to use the `-u` option with the `prodreg info` command to view software attributes by referencing the component's UUID. The UUID is the software's unique identifier in the Solaris Product Registry.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg info -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
```

▼ How to Check for Software Dependencies (`prodreg`)

You can use the `prodreg info` command to view components that depend on a specific software component. You might want to check dependencies between software products before you uninstall specific components.

Steps

1. **Open a terminal window.**
2. **Browse the Solaris Product Registry.**

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software
```

   Repeat the `prodreg browse` command until the software component you want to check is displayed. See "How to View Installed or Uninstalled Software Information (`prodreg`)" on page 270 for more information on browsing the Solaris Product Registry by using the `prodreg browse` command.

3. **View the dependencies of a specific software component.**

```
% prodreg info -m "name" -a "Dependent Components"
```

`-m` "*name*"	Displays the attributes of the software component with the name *name*.
`-a` "Dependent Components"	Displays components that depend on *name* software by displaying the values of the Dependent Components attribute.

   This command output lists the software components that depend on *name* software.

Example 16–7 Viewing Components That Depend on Other Software Products (`prodreg`)

The following example shows how to view the components that depend on the software product that is named `ExampleSoft`.

```
% prodreg -m "ExampleSoft" -a "Dependent Components"
Dependent Components:
Name                         UUID                                  #
---------------------------  ------------------------------------  -
ExampleSoftA                 7f49ecvb-1ii2-11b2-a3f1-0800119u7e8e  1
```

▼ How to Identify Damaged Software Products (`prodreg`)

If you remove installed software files or packages without using the appropriate uninstaller, you can damage the software on your system. If software is damaged, the software might not function properly. You can use the `info` subcommand of the `prodreg` command to help you determine if a software product is damaged.

Steps

1. **View the Solaris Product Registry information on the software you want to check.**

```
% prodreg browse -m name
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
4          -     name-UUID                              1  name
233         .    component-a-pkg                        1  component-a
234         .    component-b-pkg                        1
```

`-m` "*name*"	Displays information on the software component with the name *name*.
name-UUID	Specifies the UUID of the *name* software component.
component-a-pkg	Specifies the package name of the *component-a* component that depends on *name* software.
component-a	Specifies the name of a component that depends on *name* software.
component-b-pkg	Specifies the package name of the *component-b* component that depends on *name* software.

In the previous sample output, the *component-b-pkg* entry does not have an associated name in the `Name` field. If a software component name is not displayed in the Solaris Product Registry, the component might be damaged.

2. **Verify that the software component is damaged.**

```
% prodreg info -u name-UUID -i 1 -d
isDamaged=TRUE
```

`-u` *name-UUID*	Displays information on the *name* software component.
`-i 1`	Displays information on the first instance of the *name* software component.
`-d`	Displays the value of the `isDamaged` attribute of the *name* software component.

The output `isDamaged=TRUE` indicates that the *name* software component is damaged.

3. **Identify the packages that form the *name-UUID* software component.**

```
% prodreg info -u name-UUID -i 1 -a PKGS
pkgs:
component-a-pkg  component-b-pkg
```

4. **Verify that these packages are installed on the system.**

```
% pkginfo component-a-pkg
application component-a-pkg  component-a

% pkginfo component-b-pkg
ERROR: information on "component-b-pkg" was not found
```

The error message output of the `pkginfo` *component-b-pkg* command indicates that the *component-b-pkg* package has been removed from the system. The *name* software component might not work without the *component-b-pkg* package.

Example 16–8 Identifying Damaged Software Components (`prodreg`)

The following example shows how to determine if the ExampleSoft software component is damaged.

```
% prodreg browse -m Examplesoft
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
4          -     95842091-725a-8501-ef29-0472985982be  1  ExampleSoft
233         .    90209809-9785-b89e-c821-0472985982be  1  Example Doc
234         .    EXSOzzt                               1
235         .    EXSOblob                              1  Example Data
```

The ExampleSoft child component EXSOzzt does not have an entry in the NAME field. The ExampleSoft software might be damaged. You would use the prodreg info command with the -u, -i, and -d options to determine if the ExampleSoft software is damaged.

```
% prodreg info -u 95842091-725a-8501-ef29-0472985982be -i 1 -d
isDamaged=TRUE
```

The output isDamaged=TRUE indicates that the ExampleSoft software is damaged. You would use the -a PKGS option of the prodreg info command to identify the ExampleSoft software packages.

```
% prodreg info
     -u 95842091-725a-8501-ef29-0472985982be
     -i 1 -a PKGS
pkgs:
EXSOzzt EXSOblob
```

To verify that the EXSOzzt and EXSOblob packages are installed on the system, you would use the pkginfo command.

```
% pkginfo EXSOzzt
ERROR: information for "EXSOzzt" was not found

% pkginfo EXSOblob
application EXSOblob       Example Data
```

The output of the pkginfo command indicates that the EXSOzzt package is not installed on the system. Thus, the ExampleSoft software is damaged.

▼ How to Uninstall Software (prodreg)

You can use the uninstall subcommand of the prodreg command to remove software from your system. When you uninstall software by using the prodreg uninstall command, you remove a specified software and all the child components associated with that software. Before you remove software, verify that other software does not depend on the software you want to uninstall. See "How to Check for Software Dependencies (prodreg)" on page 274.

After you uninstall software, you can remove that software and all the child components of that software from the Solaris Product Registry by using the prodreg unregister -r command.

Steps 1. **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **View the information on the software you want to uninstall.**

```
# prodreg browse -u name-UUID
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
1423       -     name-UUID                              1  name
1436        .    component-a-UUID                       1  component-a
1437        -    component-b-UUID                       1  component-b
1462          .  component-c-UUID                       1  component-c
```

`-u` *name-UUID*	Displays information on the software component with the unique identifier *name-UUID*.
name	Specifies the name of the software component you want to uninstall with the unique identifier *name-UUID*.
. *component-a-UUID*	Specifies the unique identifier of the *component-a* software component that is required by *name* software.
component-a	Specifies the name of a component that is required by *name* software.
- *component-b-UUID*	Specifies the unique identifier of the *component-b* component that is required by *name* software. The - symbol indicates that *component-b* requires an additional software component.
component-b	Specifies the name of a software component that is required by *name* software.
. *component-c-UUID*	Specifies the unique identifier of the *component-b* software component that is required by *component-b* software.
component-c	Specifies the name of a software component that is required by *component-b* software.

3. **Uninstall the software.**

```
# prodreg uninstall -u name-UUID
```

4. **Check the dependencies for the software that you want to uninstall.**

```
# prodreg info -u name-UUID
Title: name
.
.
.
Child Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
component-a                  component-a-UUID                     1
component-b                  component-b-UUID                     1

Required Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
component-a                  component-a-UUID                     1
component-b                  component-b-UUID                     1
```

Check the following information in the output of the `prodreg info` command.

- Child Components – Lists the software components that are associated with the *name* software component. When you unregister the *name* software, you also unregister the child components of *name* software. If the output of the previous `prodreg info` command lists any child components, verify that you want to unregister these child components.
- Required Components – Lists the software components that are required by the *name* software component. Software components might require other components that are not child components. When you uninstall and unregister a component, only child components are unregistered and uninstalled.
- Dependent Components – Lists the components that require *name* software to run. When you unregister the *name* software, you also unregister the dependent components of *name* software. If the output of the `prodreg info` command lists any dependent components, verify that you want to unregister these dependent components.

In the previous sample output, *name* software does not have any dependent components.

5. **Check the dependencies of *name* software's child components.**

```
# prodreg info -u component-a-UUID -i 1 -a "Dependent Components"
Dependent Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
name                         name-UUID                            1

# prodreg info -u component-b-UUID -i 1 -a "Dependent Components"
Dependent Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
name                         name-UUID                            1
```

```
# prodreg info -u component-c-UUID -i 1 -a "Dependent Components"
Dependent Components:
Name                          UUID                                  #
---------------------------  ------------------------------------  -
component-b                   component-b-UUID                      1
```

The sample output shows that no other software depends on the child components of *name* software.

6. **Unregister the software and its child components.**

```
# prodreg unregister -r -u name-UUID -i 1
```

`-r`	Recursively unregisters software with the unique identifier *name-UUID* and all the child components of this software.
`-u` *name-UUID*	Specifies the unique identifier of the software you want to unregister.
`-i 1`	Specifies the instance of the software you want to unregister.

Example 16–9 Example—Uninstalling Software Components (`prodreg`)

The following example shows how to uninstall ExampleSoft software and all the child components of ExampleSoft software.

```
# prodreg browse -m "ExampleSoft"
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
1423       -     95842091-725a-8501-ef29-0472985982be  1  ExampleSoft
1436        .    90209809-9785-b89e-c821-0472985982be  1  Example Doc
1437        -    EXSOzzt                               1  Example Data
1462         .   EXSOblob                              1  Example Data

# prodreg uninstall -u 95842091-725a-8501-ef29-0472985982be -i 1

# prodreg info -u 95842091-725a-8501-ef29-0472985982be
Title: ExampleSoft Software
.
.
.
Child Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
Example Doc                 90209809-9785-b89e-c821-0472985982be  1
```

```
Example Data                EXSOzzt                                 1

Required Components:
Name                        UUID                                    #
--------------------------  ------------------------------------  -
Example Doc                 90209809-9785-b89e-c821-0472985982be  1
Example Data                EXSOzzt                                 1

# prodreg info -u 90209809-9785-b89e-c821-0472985982be -i 1
    -a "Dependent Components"
Dependent Components:
Name                         UUID                                    #
---------------------------  ------------------------------------  -
ExampleSoft                  95842091-725a-8501-ef29-0472985982be  1

# prodreg info -u EXSOzzt -i 1 -a "Dependent Components"
Dependent Components:
Name                         UUID                                    #
---------------------------  ------------------------------------  -
ExampleSoft                  95842091-725a-8501-ef29-0472985982be  1

# prodreg info -u EXSOblob -i 1 -a "Dependent Components"
Dependent Components:
Name                         UUID                                    #
---------------------------  ------------------------------------  -
Example Data                 EXSOzzt                                 1

# prodreg unregister -r -u 95842091-725a-8501-ef29-0472985982be -i 1
```

▼ How to Uninstall Damaged Software (`prodreg`)

If you try to uninstall a damaged software component by using the `prodreg uninstall` command, the command might fail. This failure can occur if the uninstaller program for the software component has been removed from the system.

Follow these steps to uninstall a software component with no associated uninstaller program on the system.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **View the information on the software you want to uninstall.**

   ```
   # prodreg browse -m "name"
   BROWSE #  +/-/.  UUID                                  #  NAME
   ========  =====  ====================================  =  ============
   1         -      root                                  1  System
   ```

```
                                                                  Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1        Solaris 10
                                                                  System
                                                                  Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1        Unclassified
                                                                  Software
4          -     UUID                                  1        name
1436       .     component-a-UUID                      1        component-a
1437       .     component-b-UUID                      1
```

`-m` "*name*"	Displays information on the *name* software component you want to uninstall.
UUID	Specifies the UUID of the software component you want to uninstall.
. *component-a-UUID*	Specifies the UUID of the *component-a* software component.
component-a	Specifies the name of a child software component of *name* software.
. *component-b-UUID*	Specifies the UUID of a child software component of *name* software.

The *component-b-UUID* entry does not have an associated component name. The missing name value might indicate that this component is damaged.

3. **Uninstall the software.**

```
# prodreg uninstall -u UUID -i 1
The install program requested could not be found
```

`-u` *UUID*	Specifies the UUID of the software component you want to uninstall.
`-i` 1	Specifies the instance of the software you want to uninstall.

The error message indicates that the uninstaller program is not on the system.

4. **Identify the uninstaller program for the software component.**

```
# prodreg info -m "name" -a uninstallprogram
uninstallprogram: /usr/bin/java -mx64m -classpath
uninstaller-location uninstall_name
```

`-m` "*name*"	Displays information on the *name* software component.
`-a uninstallprogram`	Displays information on the uninstaller program that is associated with the *name* software component.

uninstaller-location	Specifies the registered location of the uninstaller program for the *name* software component.

5. **Determine if the uninstaller is in the registered location.**

```
# ls uninstaller-location
uninstaller-location:
No such file or directory
```

The output of the `ls` command indicates that the uninstaller program is not in the registered location.

6. **Remove the software from the system in one of the following ways:**

- If you have a system backup available, follow these steps:
  a. Load the uninstaller program from the backup.
  b. Run the uninstaller program from a shell command-line interface such as a terminal window.
- If you do not have access to the uninstaller program on a backup, follow these steps:
  a. Unregister the software component.

```
# prodreg unregister -u UUID -i 1
```

  b. Remove any remaining registered components that are required by the software you want to remove.

```
# pkgrm component-a-UUID
```

Example 16–10 Uninstalling Damaged Software (`prodreg`)

The following example shows how to uninstall the damaged ExampleSoft software. In this example, the uninstaller program is not readily available on a system backup.

```
# prodreg browse -m Examplesoft
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
4          -     95842091-725a-8501-ef29-0472985982be  1  ExampleSoft
233         .    90209809-9785-b89e-c821-0472985982be  1  Example Doc
234         .    EXSOzzt                               1
235         .    EXSOblob                              1  Example Data

# prodreg uninstall -u 95842091-725a-8501-ef29-0472985982be -i 1
The install program requested could not be found
```

```
# prodreg info -m "ExampleSoft" -a uninstallprogram
uninstallprogram: /usr/bin/java -mx64m -classpath
/var/sadm/prod/org.example.ExampleSoft/987573587 uninstall_ExampleSoft

# ls /var/sadm/prod/org.example.ExampleSoft/987573587
/var/sadm/prod/org.example.ExampleSoft/987573587:
No such file or directory

# prodreg unregister -u 95842091-725a-8501-ef29-0472985982be -i 1

# pkgrm EXSOblob
```

▼ How to Reinstall Damaged Software Components (`prodreg`)

If other software depends on a damaged software component, you might want to reinstall the damaged component, rather than uninstall the component and the other dependent software. You can use the `-f` option with the `prodreg unregister` command to forcibly the unregister the damaged component. Then, you can reinstall the component.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **View the information on the software you want to reinstall.**

```
# prodreg browse -m "name"
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
4           .    UUID                                  1  name
```

`-m` "*name*"	Displays information on the *name* software component you want to reinstall.
UUID	Specifies the UUID of the software component you want to reinstall.

3. **Identify the software that depends on the software you want to reinstall.**

```
# prodreg info -m "name" -a "Dependent Components"
Dependent Components:
Name                          UUID                                  #
---------------------------  ------------------------------------  -
component-a                   component-a-UUID                      1
```

`-m` "*name*"	Specifies the name of the software component you want to reinstall.
`-a "Dependent Components"`	Displays the components that depend on *name* software.
component-a	Specifies the name of a software component that depends on *name* software.
component-a-UUID	Specifies the UUID of the *component-a* software component.

The *component-a* software component depends on the software you want to reinstall. To reinstall *name* software and not unregister *component-a*, you must forcibly unregister the *name* software, then reinstall *name* software.

4. **Unregister the software component you want to reinstall.**

```
# prodreg unregister -f -u UUID
```

5. **Reinstall the software component.**

```
# /usr/bin/java -cp /usr/installers/installer
```

The *installer* option specifies the name of the installer program for *name* software.

Example 16–11 Reinstalling Damaged Software Components (`prodreg`)

The following example shows how to reinstall the damaged software component `ComponentSoft` without unregistering or uninstalling the dependent component ExampleSoft.

```
# prodreg browse -m "ComponentSoft"
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 10
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
4           .    86758449-554a-6531-fe90-4352678362fe  1  ComponentSoft

# prodreg info -m "ComponentSoft" -a "Dependent Components"
```

```
Dependent Components:
Name                         UUID                                  #
---------------------------  ------------------------------------  -
ExampleSoft                  95842091-725a-8501-ef29-0472985982be  1

# prodreg unregister -f -u 86758449-554a-6531-fe90-4352678362fe -i 1

# /usr/bin/java -cp /usr/installers/org.example.componentsoft
```

CHAPTER 17

Managing Software by Using Package Commands (Tasks)

This chapter describes how to add, verify, and remove software packages by using the package commands.

For information on the procedures associated with performing these tasks, see:

- "Adding and Removing Signed Packages by Using the `pkgadd` Command (Task Map)" on page 289
- "Managing Software Packages by Using Package Commands (Task Map)" on page 296

Adding and Removing Signed Packages by Using the `pkgadd` Command (Task Map)

The following task map describes software management tasks that you can perform with signed package commands.

Task	Description	For Instructions
Import a certificate.	You can import a trusted certificate by using the `pkgadm addcert` command.	"How to Import a Trusted Certificate From the Java Keystore (`pkgadm addcert`)" on page 290

Task	Description	For Instructions
Print the details of one or more certificates.	You can print the details of a certificate by using the `pkgadm listcert` command.	"How to Display Certificate Information (`pkgadm listcert`)" on page 292
Remove a certificate.	You can remove a certificate by using the `pkgadm removecert` command.	"How to Remove a Certificate (`pkgadm removecert`)" on page 293
Set up a proxy server.	Use this procedures for systems that are set up behind a firewall with a proxy.	"How to Set Up a Proxy Server (`pkgadd`)" on page 293
Add a signed package.	After the root certificate is imported, you can add a signed package by using he `pkgadd` command.	"How to Add a Signed Package (`pkgadd`)" on page 294

Adding and Removing Signed Packages by Using the `pkgadd` Command

The following procedures explain how to add and remove signed packages by using the `pkgadd` command.

▼ How to Import a Trusted Certificate From the Java Keystore (`pkgadm addcert`)

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Verify that the root certificate authority (CA) certificate exists in the Java ™ keystore.**

   `# keytool -storepass` *storepass* `-list -keystore` *certfile*

   `keytool` Manages a Java keystore (database) of private keys and their associated X.509 certificate chains that authenticate

	the corresponding public keys. Also manages certificates from trusted entities. For more information on the `keytool` utility, see keytool-Key and Certificate Management Tool.
`-storepass` *storepass*	Specifies the password that protects the integrity of the keystore.
`-list`	By default, prints the MD5 fingerprint of a certificate.
`-keystore` *certfile*	Specifies the name and location of the persistent keystore file.

3. **Export the root CA certificate from the Java keystore to a temporary file.**

   # **keytool -export -storepass** *storepass* **-alias gtecybertrustca -keystore gtecybertrustca -keystore** *certfile* **-file** *filename*

`-export`	Exports the trusted certificate.
`-storepass` *storepass*	Specifies the password that protects the integrity of the Java keystore.
`-alias gtecybertrustca`	Identifies the alias of the trusted certificate.
`-keystore` *certfile*	Specifies the name and location of the keystore file.
`-file` *filename*	Identifies the file to hold the exported certificate.

4. **Import a trusted certificate to the package keystore.**

   # **pkgadm addcert -t -f** *format certfile*

`-t`	Indicates that the certificate is a trusted CA certificate. The output includes the details of the certificate, which the user is asked to verify.
`-f` *format*	Specifies the format of certificates and private keys. When you import a certificate, it must be encoded using PEM or binary DER format.
certfile	Specifies the file that contains the certificate.

5. **Remove the temporary file.**

   # **rm /tmp/***file-name*

   For more information, see the `pkgadm`(1M) man page.

Example 17–1 Importing a Trusted Certificate From the Java Keystore

The following example shows how to import a trusted certificate. In this example, Sun's root CA certificate is imported from the Java keystore into the package keystore by using the `keytool` command.

```
# keytool -export -storepass changeit -alias gtecybertrustca -keystore
gtecybertrustca -keystore /usr/j2se/jre/lib/security/cacerts -file \
 /tmp/root.crt
Certificate stored in file </tmp/root.crt>
# pkgadm addcert -t -f der /tmp/root.crt
    Keystore Alias: GTE CyberTrust Root
       Common Name: GTE CyberTrust Root
  Certificate Type: Trusted Certificate
Issuer Common Name: GTE CyberTrust Root
    Validity Dates: <Feb 23 23:01:00 1996 GMT> - <Feb 23 23:59:00 2006 GMT>
   MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
  SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:
D3:91:BC:65:A6:89:64

Are you sure you want to trust this certificate? yes
Trusting certificate <GTE CyberTrust Root>
Type a Keystore protection Password.
Press ENTER for no protection password (not recommended):
For Verification: Type a Keystore protection Password.
Press ENTER for no protection password (not recommended):
Certificate(s) from </tmp/root.crt> are now trusted
# rm /tmp/root.crt
```

▼ How to Display Certificate Information (`pkgadm listcert`)

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Display the contents of the package keystore.**

   # **pkgadm listcert -p** *passarg*

Example 17–2 Displaying Certificate Information

The following example shows how to display the details of a locally stored certificate.

```
# pkgadm listcert -P pass:test123
    Keystore Alias: GTE CyberTrust Root
       Common Name: GTE CyberTrust Root
  Certificate Type: Trusted Certificate
Issuer Common Name: GTE CyberTrust Root
    Validity Dates: <Feb 23 23:01:00 1996 GMT> - <Feb 23 23:59:00 2006 GMT>
   MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
  SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91:
BC:65:A6:89:64
```

▼ How to Remove a Certificate (`pkgadm removecert`)

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Remove the trusted certificate from the package keystore.**

   ```
   # pkgadm removecert -n "certfile"
   ```

   The `removecert -n` "*certfile*" option specifies the alias of the user certificate/key pair or the alias of the trusted certificate.

   Note – View the alias names for certificates by using the `pkgadm listcert` command.

Example 17–3 Removing a Certificate

The following example shows how to remove a certificate.

```
# pkgadm listcert
    Keystore Alias: GTE CyberTrust Root
       Common Name: GTE CyberTrust Root
  Certificate Type: Trusted Certificate
Issuer Common Name: GTE CyberTrust Root
    Validity Dates: <Feb 23 23:01:00 1996 GMT> - <Feb 23 23:59:00 2006 GMT>
   MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
  SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91:BC:
65:A6:89:64
# pkgadm removecert -n "GTE CyberTrust Root"
Enter Keystore Password: storepass
Successfully removed Certificate(s) with alias <GTE CyberTrust Root>
```

▼ How to Set Up a Proxy Server (`pkgadd`)

If your system is behind a firewall with a proxy, you will need to set up a proxy server before you can add a package from an HTTP server by using the `pkgadd` command.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Select one of the following methods to specify a proxy server.**

   a. **Specify the proxy server by using the `http_proxy`, `HTTPPROXY`, or `HTTPPROXYPORT` environment variable.**

      For example:

      ```
      # setenv http_proxy http://mycache.domain:8080
      ```

      Or, specify one of the following:

      ```
      # setenv HTTPPROXY mycache.domain
      # setenv HTTPPROXYPORT 8080
      ```

   b. **Specify the proxy server on the `pkgadd` command line.**

      For example:

      ```
      # pkgadd -x mycache.domain:8080 -d http://myserver.com/pkg SUNWpkg
      ```

   c. **Create an administration file that includes proxy server information.**

      For example:

      ```
      # cat /tmp/admin
      mail=
      instance=unique
      partial=ask
      runlevel=ask
      idepend=ask
      rdepend=ask
      space=ask
      setuid=ask
      conflict=ask
      action=ask
      networktimeout=60
      networkretries=3
      authentication=quit
      keystore=/var/sadm/security
      basedir=default
      proxy=mycache.domain:8080
      ```

      Then, identify the administration file by using the `pkgadd -a` command. For example:

      ```
      # pkgadd -a /tmp/admin -d http://myserver.com/pkg SUNWpkg
      ```

▼ How to Add a Signed Package (`pkgadd`)

This procedure assumes that you have imported Sun's root CA certificate. For more information, see "How to Import a Trusted Certificate From the Java Keystore (`pkgadm addcert`)" on page 290.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Add a signed package.**

   # **pkgadd -d** */pathname/device-name*

   The `-d` *device-name* option specifies the device from which the package is installed. The device can be a directory, tape, diskette, or removable disk. The device can also be a data stream created by the `pkgtrans` command.

Example 17–4 Adding a Signed Package

The following example shows how to add a signed package that is stored on the system.

```
# # pkgadd -d /tmp/signed_pppd
The following packages are available:
  1  SUNWpppd     Solaris PPP Device Drivers
                  (sparc) 11.10.0,REV=2003.05.08.12.24

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: all
Enter keystore password:
## Verifying signature for signer <User Cert 0>

.
.
.
```

The following example shows how to install a signed package using an `HTTP` URL as the device name. The URL must point to a stream-formatted package.

```
# pkgadd -d http://install/signed-video.pkg

## Downloading...
..............25%..............50%..............75%..............100%
## Download Complete
.
.
.
```

Managing Software Packages by Using Package Commands (Task Map)

The following task map describes the software management tasks that you can perform with the package commands for both signed and unsigned packages.

Task	Description	For Instructions
Add software packages to the local system.	You can add software packages to the local system by using the `pkgadd` command.	"How to Add Software Packages (`pkgadd`)" on page 297
Add software packages to a spool directory.	You can add software packages to a spool directory without actually installing the software.	"Adding a Software Package to a Spool Directory" on page 299
List information about all installed software packages.	You can list information about installed packages by using the `pkginfo` command.	"How to List Information About All Installed Packages (`pkginfo`)" on page 301
Check the integrity of installed software packages.	You can verify the integrity of installed software packages by using the `pkgchk` command.	"How to Check the Integrity of Installed Software Packages (`pkgchk`)" on page 302
Check the integrity of an installed object.	You can verify the integrity of an installed object by using the `pkchk` command with the `-p` and `-P` options. The `-p` option specifies the full path name. The new `-P` option specifies a partial path name.	"How to Check the Integrity of Installed Objects (`pkgchk -p, pkgchk -P`)" on page 304
Remove software packages.	You can remove unneeded software packages by using the `pkgrm` command.	"How to Remove Software Packages (`pkgrm`)" on page 306

Using Package Commands to Manage Software Packages

The following procedures explain how to manage software packages by using package commands.

▼ How to Add Software Packages (pkgadd)

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Remove any already installed packages with the same names as the packages you are adding.**

   This step ensures that the system keeps a proper record of software that has been added and removed. Sometimes, you might want to maintain multiple versions of the same application on the system. For strategies on maintaining multiple software copies, see "Guidelines for Removing Packages (pkgrm)" on page 258. For task information, see "How to Remove Software Packages (pkgrm)" on page 306.

3. **Add a software package to the system.**

   # **pkgadd -a** *admin-file* **-d** *device-name pkgid* . . .

-a *admin-file*	(Optional) Specifies an administration file that the pkgadd command should check during the installation. For details about using an administration file, see "Using an Administration File" on page 259.
-d *device-name*	Specifies the absolute path to the software packages. *device-name* can be the path to a device, a directory, or a spool directory. If you do not specify the path where the package resides, the pkgadd command checks the default spool directory (/var/spool/pkg). If the package is not there, the package installation fails.
pkgid	(Optional) Is the name of one or more packages, separated by spaces, to be installed. If omitted, the pkgadd command installs all available packages from the specified device, directory, or spool directory.

   If the pkgadd command encounters a problem during installation of the package, it displays a message related to the problem, followed by this prompt:

   ```
   Do you want to continue with this installation?
   ```

   Respond with yes, no, or quit. If more than one package has been specified, type no to stop the installation of the package being installed. The pkgadd command continues to install the other packages. Type quit to stop the installation.

4. **Verify that the package has been installed successfully.**

   # **pkgchk -v** *pkgid*

   If no errors occur, a list of installed files is returned. Otherwise, the pkgchk command reports the error.

Example 17–5 Adding Software Packages From a Mounted CD

The following example shows how install the `SUNWpl5u` package from a mounted Solaris 10 CD. The example also shows how to verify that the package files were installed properly.

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_10/Product SUNWpl5u
    .
    .
    .
Installation of <SUNWpl5u> was successful.
# pkgchk -v SUNWpl5u
/usr
/usr/bin
/usr/bin/perl
/usr/perl5
/usr/perl5/5.8.4
.
.
.
```

Example 17–6 Installing Software Packages From a Remote Package Server

If the packages you want to install are available from a remote system, you can manually mount the directory that contains the packages (in package format) and install packages on the local system.

The following example shows how to install software packages from a remote system. In this example, assume that the remote system named `package-server` has software packages in the `/latest-packages` directory. The `mount` command mounts the packages locally on `/mnt`. The `pkgadd` command installs the `SUNWpl5u` package.

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt SUNWpl5u
    .
    .
    .
Installation of <SUNWpl5u> was successful.
```

If the automounter is running at your site, you do not need to mount the remote package server manually. Instead, use the automounter path, in this case, `/net/package-server/latest-packages`, as the argument to the `-d` option.

```
# pkgadd -d /net/package-server/latest-packages SUNWpl5u
    .
    .
    .
Installation of <SUNWpl5u> was successful.
```

Example 17–7 Installing Software Packages From a Remote Package Server by Specifying an Administration File

This example is similar to the previous example, except that it uses the `-a` option and specifies an administration file named `noask-pkgadd`, which is illustrated in "Avoiding User Interaction When Adding Packages (pkgadd)" on page 259. In this example, assume that the `noask-pkgadd` administration file is in the default location, `/var/sadm/install/admin`.

```
# pkgadd -a noask-pkgadd -d /net/package-server/latest-packages SUNWpl5u
    .
    .
    .
Installation of <SUNWpl5u> was successful.
```

Example 17–8 Installing Software Packages From an HTTP URL

The following example shows how to install a package using an `HTTP` URL as the device name. The URL must point to a stream-formatted package.

```
# pkgadd -d http://install/xf86-4.3.0-video.pkg

## Downloading...
..............25%.............50%.............75%.............100%
## Download Complete


The following packages are available:
  1  SUNWxf86r     XFree86 Driver Porting Kit (Root)
                   (i386) 4.3.0,REV=0.2003.02.28
  2  SUNWxf86u     XFree86 Driver Porting Kit (User)
                   (i386) 4.3.0,REV=0.2003.02.28

.
.
.
```

Adding a Software Package to a Spool Directory

For convenience, you can copy frequently installed packages to a spool directory. If you copy packages to the default spool directory, `/var/spool/pkg`, you do not need to specify the source location of the package (`-d` *device-name* argument) when you use the `pkgadd` command. The `pkgadd` command, by default, checks the `/var/spool/pkg` directory for any packages that are specified on the command line. Note that copying packages to a spool directory is not the same as installing the packages on a system.

▼ How to Add Software Packages to a Spool Directory (pkgadd)

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Remove any already spooled packages with the same names as the packages you are adding.**

   For information on removing spooled packages, see Example 17–20.

3. **Add a software package to a spool directory.**

   `# pkgadd -d` *device-name* `-s` *spooldir pkgid* `...`

`-d` *device-name*	Specifies the absolute path to the software packages. *device-name* can be the path to a device, a directory, or a spool directory.
`-s` *spooldir*	Specifies the name of the spool directory where the package will be spooled. You must specify a *spooldir*.
pkgid	(Optional) Is the name of one or more packages, separated by spaces, to be added to the spool directory. If omitted, the `pkgadd` command copies all available packages.

4. **Verify that the package has been copied successfully to the spool directory.**

   `$ pkginfo -d` *spooldir* `| grep` *pkgid*

   If *pkgid* was copied correctly, the `pkginfo` command returns a line of information about the *pkgid*. Otherwise, the `pkginfo` command returns the system prompt.

Example 17–9 Setting Up a Spool Directory From a Mounted CD

The following example shows how to transfer the `SUNWman` package from a mounted SPARC based Solaris 10 CD to the default spool directory (`/var/spool/pkg`).

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_10/Product -s /var/spool/pkg SUNWman
Transferring <SUNWman> package instance
```

Example 17–10 Setting Up a Spool Directory From a Remote Software Package Server

If packages you want to copy are available from a remote system, you can manually mount the directory that contains the packages, in package format, and copy them to a local spool directory.

The following example shows the commands for this scenario. In this example, assume that the remote system named `package-server` has software packages in the `/latest-packages` directory. The `mount` command mounts the package directory locally on `/mnt`. The `pkgadd` command copies the `SUNWpl5p` package from `/mnt` to the default spool directory (`/var/spool/pkg`).

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt -s /var/spool/pkg SUNWpl5p
Transferring <SUNWpl5p> package instance
```

If the automounter is running at your site, you do not have to mount the remote package server manually. Instead, use the automounter path, in this case, `/net/package-server/latest-packages`, as the argument to the `-d` option.

```
# pkgadd -d /net/package-server/latest-packages -s /var/spool/pkg SUNWpl5p
Transferring <SUNWpl5p> package instance
```

Example 17–11 Installing Software Packages From the Default Spool Directory

The following example shows how to install the `SUNWpl5p` package from the default spool directory. When no options are used, the `pkgadd` command searches the `/var/spool/pkg` directory for the named packages.

```
# pkgadd SUNWpl5p
    .
    .
    .
Installation of <SUNWpl5p> was successful.
```

▼ How to List Information About All Installed Packages (`pkginfo`)

Step

- **List information about installed packages by using the `pkginfo` command.**

```
$ pkginfo
```

Example 17–12 Listing Installed Packages

This example shows how to list all packages installed on a local system, whether that system is a stand-alone system or a server. The output shows the primary category, package name, and the description of the package.

```
$ pkginfo
system      SUNWaccr       System Accounting, (Root)
system      SUNWaccu       System Accounting, (Usr)
system      SUNWadmap      System administration applications
system      SUNWadmc       System administration core libraries
```

.
.
.

Example 17–13 Displaying Detailed Information About Software Packages

This example shows how to list all packages installed on a system by specifying the long format, which includes all available information about the designated packages.

```
$ pkginfo -l SUNWcar
   PKGINST:  SUNWcar
      NAME:  Core Architecture, (Root)
  CATEGORY:  system
      ARCH:  sparc.sun4u
   VERSION:  11.9.0,REV=2002.04.06.15.27
   BASEDIR:  /
    VENDOR:  Sun Microsystems, Inc.
      DESC:  core software for a specific hardware platform group
    PSTAMP:  leo20031003183400
  INSTDATE:  Feb 20 2004 16:57
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:      114 installed pathnames
                  36 shared pathnames
                  40 directories
                  57 executables
                21469 blocks used (approx)
```

▼ How to Check the Integrity of Installed Software Packages (pkgchk)

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Check the status of an installed package.**

   - To check the file attributes and contents, type the following:

     # **pkgchk -a| -c -v** *pkgid* . . .

   - To specify the absolute path of the spool directory, type the following:

     # **pkgchk -d** *spooldir pkgid* . . .

   -a Specifies to audit only the file attributes (the permissions), rather than the file attributes and the contents, which is the default.

`-c`	Specifies to audit only the file contents, rather than the file contents and attributes, which is the default.
`-v`	Specifies verbose mode, which displays file names as they are processed.
`-d` *spooldir*	Specifies the absolute path of the spool directory.
pkgid	(Optional) Is the name of one or more packages, separated by spaces. If you do not specify a *pkgid*, all the software packages installed on the system are checked.

Example 17–14 Checking the Contents of Installed Software Packages

The following example shows how to check the contents of a package.

```
# pkgchk -c SUNWbash
```

If no errors occur, the system prompt is returned. Otherwise, the `pkgck` command reports the error.

Example 17–15 Checking the File Attributes of Installed Software Packages

The following example shows how to check the file attributes of a package.

```
# pkgchk -a SUNWbash
```

If no errors occur, the system prompt is returned. Otherwise, the `pkgck` command reports the error.

Example 17–16 Checking Software Packages Installed in a Spool Directory

The following example shows how to check a software package that was copied to a spool directory (`/export/install/packages`).

```
# pkgchk -d /export/install/packages
## checking spooled package <SUNWadmap>
## checking spooled package <SUNWadmfw>
## checking spooled package <SUNWadmc>
## checking spooled package <SUNWsadml>
```

The checks made on a spooled package are limited because not all information can be audited until a package is installed.

▼ How to Check the Integrity of Installed Objects (`pkgchk -p, pkgchk -P`)

This procedure explains how to use the `pkgchk` command to check the integrity of installed objects. The new `-P` option enables you to specify a partial path. This option has been added to assist you in mapping files to packages. Use this option with the `-l` option to list the information about the files that contain the partial path. Use the `-p` option to check the integrity of installed objects by specifying the full path. For more information, see the `pkgchk`(1M) man page.

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Check the integrity of an installed object.**

   - To verify the integrity of an installed object for a full path name or path names, type the following:

     ```
     # pkgchk -lp path-name
     ```

   - To verify the integrity of an installed object for a partial-path name or path names, type the following:

     ```
     # pkgchk -lP partial-path-name
     ```

   `-p` *path* — Checks the accuracy only of the path name or path names that are listed. Path can be one or more path names separated by commas. Specifies to audit only the file attributes (the permissions), rather than the file attributes and the contents, which is the default.

   `-P` *partial-path* — Checks the accuracy of only the partial path name or path names that are listed. The partial-path can be one or more partial path names separated by commas. Matches any path name that contains the string contained in the partial path. Specifies to audit only the file contents, rather than the file contents and attributes, which is the default.

   `-l` — Lists information about the selected files that make up a package. This option is not compatible with the `-a`, `-c`, `-f`, `-g`, and `-v` options. Specifies verbose mode, which displays file names as they are processed.

Example 17–17 Checking the Integrity of an Installed Object by Specifying a Full Path Name

This example shows you how to use the `pkgchk -lp` command to check the contents/attributes of an object on a file system by a specifying the full path name. The `-l` option lists information on the selected files that make up a package.

```
# pkgchk -lp /usr/sbin/pkgadd
Pathname: /usr/sbin/pkgadd
Type: regular file
Expected mode: 0555
Expected owner: root
Expected group: sys
Expected file size (bytes): 867152
Expected sum(1) of contents: 45580
Expected last modification: Jul 02 02:20:34 2004
Referenced by the following packages:
        SUNWpkgcmdsu
Current status: installed
```

Example 17–18 Checking the Integrity of an Installed Object by Specifying a Partial Path Name

This example shows you how to use the `pkgchk -lP` command to check the contents/attributes of an object on a file system by a specifying a partial path name, such as a file or directory name. The `-l` option lists information on the selected files that make up a package.

```
# pkgchk -lP /sbin/pkgadd
Pathname: /usr/sbin/pkgadd
Type: regular file
Expected mode: 0555
Expected owner: root
Expected group: sys
Expected file size (bytes): 867152
Expected sum(1) of contents: 45580
Expected last modification: Jul 02 02:20:34 2004
Referenced by the following packages:
        SUNWpkgcmdsu
Current status: installed

Pathname: /usr/sbin/pkgask
Type: linked file
Source of link: ../../usr/sbin/pkgadd
Referenced by the following packages:
        SUNWpkgcmdsu
Current status: installed
```

Removing Software Packages

To remove or uninstall a software package, use the associated tool that you used to add or install a software package. For example, if you used the Solaris installation GUI to install software, use the Solaris installation GUI to uninstall software.

Caution – Do no use the `rm` command to remove software packages. Doing so will result in inaccuracies in the database that keeps track of all installed packages on the system.

▼ How to Remove Software Packages (`pkgrm`)

Steps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Remove an installed package.**

   # **pkgrm** *pkgid* ...

   pkgid identifies the name of one or more packages, separated by spaces, to be removed. If omitted, the `pkgrm`command removes all available packages.

Example 17–19 Removing Software Packages

This example shows how to remove a package.

```
# pkgrm SUNWctu

The following package is currently installed:
   SUNWctu         Netra ct usr/platform links (64-bit)
                   (sparc.sun4u) 11.9.0,REV=2001.07.24.15.53

Do you want to remove this package? y

## Removing installed package instance <SUNWctu>
## Verifying package dependencies.
## Processing package information.
## Removing pathnames in class <none>
.
.
.
```

Example 17–20 Removing a Spooled Software Package

This example shows how to remove a spooled package.

```
# pkgrm -s /export/pkg SUNWaudh
The following package is currently spooled:
   SUNWaudh        Audio Header Files
                   (sparc) 11.10.0,REV=2003.08.08.00.03
Do you want to remove this package? y
Removing spooled package instance <SUNWaudh>
```

CHAPTER 18

Managing Solaris Patches (Overview)

Patch management involves *applying* Solaris patches to a system. Patch management might also involve removing unwanted or faulty patches. Removing patches is also called *backing out* patches.

The following overview information is in this chapter:

For information about the Sun Patch Manager tool (Patch Manager) and for step-by-step instructions on using Patch Manager to manage patches, see Chapter 19.

For step-by-step instructions on using the `patchadd` command to manage patches, see Chapter 20.

For information about applying patches to diskless client systems, see "Patching Diskless Client OS Services" on page 138.

For information about recommended strategies and practices for using Solaris patches, go to `http://docs.sun.com/doc/817-0574/`.

Types of Patches

A *patch* is a collection of files and directories that replaces or updates existing files and directories that are preventing proper execution of the existing software. The existing software is derived from a specified *package* format, which conforms to the Application Binary Interface (ABI). For details about packages, see Chapter 15.

You can manage patches on your Solaris system by using the Patch Manager software or by using the `patchadd` command.

Signed and Unsigned Patches

A *signed patch* is one that has a *digital signature* applied to it. A patch that has its digital signature verified has not been modified since the signature was applied. The digital signature of a signed patch is verified after the patch is *downloaded* to your system.

Patches for the Solaris 2.6, Solaris 7, Solaris 8, Solaris 9, and Solaris 10 releases are available as signed patches and as *unsigned patches*. Unsigned patches do not have a digital signature.

Signed patches are stored in Java archive format (JAR) files and are available from the SunSolve Online℠ web site. Unsigned patches are stored in directory format and are also available from the SunSolve Online web site as `.zip` files.

For information about applying patches to your system by using Patch Manager, see "Managing Patches by Using the Command-Line Interface (Task Map)" on page 338.

For information about applying patches to your system by using the `patchadd` command, see "Managing Solaris Patches by Using the `patchadd` Command (Task Map)" on page 357.

For additional overview information about signed patches, see "Signed Packages and Patches" on page 252.

Accessing Solaris Patches

Sun customers can access patches from the SunSolve Online web site whether or not they are in the SunSpectrum℠ program. These patches are updated nightly.

- **If you are in the SunSpectrum program –** You have access to the entire SunSolve℠ database of patches and all patch information.
- **If you are *not* in the SunSpectrum program –** As of this Solaris release, you have access to the entire SunSolve database of patches and all patch information except for patches that have third-party contract restrictions.

You can obtain Solaris patches in the following ways:

- From the `http://sunsolve.sun.com` web site

  To access patches from the Patch Portal of the SunSolve Online site, your system must be connected to the Internet and be capable of running a web browser, such as the Netscape™ software.

- By using anonymous `ftp` to download the patches to your system

  To obtain patches by using the anonymous `ftp` command, your system must be connected to the Internet and be capable of running the `ftp` command.
- By using the Patch Manager tools that are described in this book

You can access individual patches or a set of patches from a patch cluster, or refer to patch reports. You can also use Sun Patch Manager to *analyze* your system to determine the appropriate patches. Patch Manager also can download and apply the patches to your system. See Chapter 19.

Each patch is associated with a README file that has information about the patch.

Solaris Patch Numbering

Patches are identified by unique *patch IDs*. A patch ID is an alphanumeric string that is a patch base code and a number that represents the patch revision number joined with a hyphen. For example, patch 108528-10 is the patch ID for the SunOS 5.8 kernel update patch.

Tools for Managing Solaris Patches

You can use the following tools to apply patches to Solaris systems:

- Sun Patch Manager command-line interface (`smpatch`)
- `patchadd`
- Solaris Management Console Patches tool (GUI, starting with Solaris 9)

If you need to apply a patch to a diskless client system, see "Patching Diskless Client OS Services" on page 138.

The following table summarizes the availability of the Solaris patch management tools.

Tool Availability	`patchadd`/`patchrm` Commands	Solaris 2.6 and Solaris 7 Patch Management Tools	Sun Patch Manager 2.0	PatchPro Interactive or PatchPro Expert
How do I get this tool?	Included with the Solaris release	Download the tool from the Sun Download Center web site[1]	Included with the Solaris 10 release (`SUNWCprog`, `SUNWCall`, and `SUNWCXall`) Download the Solaris 8 or Solaris 9 version of the tool from the Sun Download Center web site[1]	Run tool from the PatchPro web site[2]
Solaris release availability	Solaris 2.6, Solaris 7, Solaris 8, Solaris 9, and Solaris 10 releases	Solaris 2.6 and Solaris 7 releases	Solaris 8, Solaris 9, and Solaris 10 releases	Solaris 2.6, Solaris 7, Solaris 8, Solaris 9, and Solaris 10 releases
Applies signed patches?	**Starting with the Solaris 9 12/03 release –** Yes, and automatically verifies the signed patch when it is downloaded	Yes, and automatically verifies the signed patch when it is downloaded	Yes, and automatically verifies the signed patch when it is downloaded	No, these tools do not apply patches
Applies unsigned patches?	Yes	No	Yes, but the patches must be unzipped first	No
GUI available?	No	No	Yes, for Solaris 9 and Solaris 10 systems only	Yes, these tools can only be run from the PatchPro web site[2]
Analyzes system to determine the appropriate patches and downloads signed or unsigned patches	No	Yes, signed patches only	Yes, signed patches only	Yes, unsigned patches only

[1] The Sun Download Center web site is `http://wwws.sun.com/software/download`.

[2] The PatchPro web site is `http://www.sun.com/PatchPro`.

Tool Availability	`patchadd`/`patchrm` Commands	Solaris 2.6 and Solaris 7 Patch Management Tools	Sun Patch Manager 2.0	PatchPro Interactive or PatchPro Expert
Local and remote system patch support	Local	Local	Local and remote **For Solaris 8 systems –** Local	No
RBAC support?	Yes	No	Yes	No

Note – Starting with the Solaris 9 release – A graphical user interface (GUI), the Patches tool in the Solaris Management Console (`smc`), is also available. The Patches tool enables you to analyze systems to determine the appropriate patches, view patch properties, download patches, apply patches to systems, and remove patches.

Managing Solaris Patches

When you apply a patch, the patch tools call the `pkgadd` command to apply the patch packages from the patch directory to a local system's disk.

Caution – Do *not* run the `pkgadd` command directly to apply patches.

More specifically, the patch tools do the following:

- Determine the Solaris version number of the managing host and the target host
- Update the patch package's `pkginfo` file with this information:
  - Patches that have been *obsoleted* by the patch being applied
  - Other patches that are required by this patch
  - Patches that are *incompatible* with this patch

While you apply patches, the `patchadd` command logs information in the `/var/sadm/patch/`*patch-id*`/log` file.

The `patchadd` command cannot apply a patch under the following conditions:

- The package is not fully installed on the system.
- The patch package's architecture differs from the system's architecture.
- The patch package's version does not match the installed package's version.
- A patch with the same base code and a higher revision number has already been applied.

- A patch that obsoletes this patch has already been applied.
- The patch is incompatible with a patch that has already been applied to the system. Each patch that has been applied keeps this information in its `pkginfo` file.
- The patch being applied depends on another patch that has not yet been applied.

Selecting the Best Method for Applying Patches

You can use several different methods to download or apply one or more patches to your system. Use the following table to determine which method is best for your needs.

Command or Tool	Description	For More Information
`smpatch update`	**Starting with the Solaris 8 release –** Use this command to analyze your system to determine the appropriate patches and to automatically download and apply the patches. Note that this command will not apply a patch that has the `interactive` property set. **Note –** For Solaris 8 systems, only the local mode `smpatch` is available.	"How to Update Your System With Patches (Command Line)" on page 341 `smpatch`(1M) man page
`smpatch analyze` and `smpatch update`	**Starting with the Solaris 8 release –** First, use `smpatch analyze` to analyze your system to determine the appropriate patches. Then, use `smpatch update` to download and apply one or more of the patches to your system. **Note –** For Solaris 8 systems, only the local mode `smpatch` is available.	"How to Analyze Your System to Obtain the List of Patches to Apply (Command Line)" on page 340 "How to Update Your System With Patches (Command Line)" on page 341 `smpatch`(1M) man page

Command or Tool	Description	For More Information
`smpatch analyze`, `smpatch download`, and `smpatch add`	**Starting with the Solaris 8 release –** First, use `smpatch analyze` to analyze your system to determine the appropriate patches. Then, use `smpatch download` to download them. This command also downloads any prerequisite patches. Then, use `smpatch add` to apply one or more of the patches to your system while the system is in single-user or multiuser mode. **Note –** For Solaris 8 systems, only the local mode `smpatch` is available.	"Managing Patches by Using the Command-Line Interface (Task Map)" on page 338 `smpatch`(1M) man page
`patchadd`	**Starting with the Solaris 2.6 release –** Apply unsigned patches to your system. **Starting with the Solaris 9 12/03 release –** Use this command to apply either signed or unsigned patches to your system. To apply signed patches, you must first set up your package keystore.	Chapter 20 `patchadd`(1M) man page
Solaris Management Console Patches tool	**Starting with the Solaris 9 release –** Use this tool when you want the convenience of a GUI tool to manage signed patches.	Solaris Management Console online help

If you choose to use the the `smpatch` command-line interface, see "Getting Started With Patch Manager" on page 330 for additional information that might affect which method you select to apply patches.

Managing Patches in the Solaris Operating System (Road Map)

Use this road map to identify all the tasks for managing Solaris patches. Each task points to a series of additional tasks such as managing signed or unsigned patches.

Task	Description	For Instructions
Determine whether to apply signed or unsigned patches.	Determine whether applying signed or unsigned patches is best for your environment.	"Determining Whether to Apply Signed or Unsigned Patches to Your System" on page 316
Apply a patch to your system.	You can apply patches in the following ways: ■ Use the `smpatch` command on Solaris 8, Solaris 9, or Solaris 10 systems to apply signed or unsigned patches. ■ Use the `patchadd` command on Solaris 2.6, Solaris 7, Solaris 8, Solaris 9, or Solaris 10 systems to apply unsigned Solaris patches. **Starting with the Solaris 9 12/03 release –** Use the `patchadd` command to apply either signed or unsigned patches.	"Managing Solaris Patches by Using the Sun Patch Manager Command-Line Interface (Task Map)" on page 331 "Managing Solaris Patches by Using the `patchadd` Command (Task Map)" on page 357

Determining Whether to Apply Signed or Unsigned Patches to Your System

The key factor when determining whether to apply signed or unsigned patches to your system is whether you trust of the source of patches.

If you trust the source of patches, for example, a patch CD from a known distributor or an HTTPS connection to a trusted web site, you can use unsigned patches. However, if you do not trust the source, use signed patches.

If you are unsure about whether to trust the source of patches, use signed patches.

Solaris Patch Management Terms and Definitions

The following terms are used throughout the patch management chapters.

analyze — To check a system to determine the list of patches that are appropriate for this system.

Patch Manager uses analysis modules and a list of available patches from the Sun patch server to generate a list of patches for your Solaris system.

apply — To install a patch on a system.

back out — To remove a patch from a system.

backout data — Data that is created when a patch is applied to enable the system to return to its previous state if the patch is removed (backed out).

backout directory — Directory in which backout data is stored. By default, this is the `save` directory of each package that was installed by the patch.

caching — The ability of a server in a chain of patch servers to store a patch that has been downloaded to it from another server.

dependency — See **patch dependency**.

digital signature — An electronic signature that can be used to ensure that a document has not been modified since the signature was applied.

download — To copy one or more patches from a source of patches, such as the Sun patch server, to the system where the patches are to be applied.

download directory — Directory in which patches are stored when they are downloaded from the patch source. This is also the directory from which patches are applied. The default location is `/var/sadm/spool`.

keystore — A repository of certificates and keys that is queried when you attempt to apply a signed patch.

local mode — A mode available for the `smpatch` command, which can only be run on the local system. This mode can be used to apply patches while the system is in single-user mode or in multiuser mode.

nonstandard patch — A patch that is associated with the `interactive` property, with one or more of the `rebootafter`, `rebootimmediate`, `reconfigafter`, `reconfigimmediate`, and `singleuser` properties, or a patch that cannot be applied by running the usual patch management tools.

order — To sort a set of patches in an order suitable for applying patches.

package	The form in which software products are delivered for installation on a system. The package contains a collection of files and directories in a defined format.
patch	An update to software that corrects an existing problem or that introduces a feature.
patch analysis	A method of checking a system to determine which patches are appropriate for the system.
patch dependency	An instance where a patch depends on the existence of another patch on a system. A patch that depends on one or more patches can only be applied to a system when those other patches have already been applied.
patch ID	A unique alphanumeric string, with the patch base code first, a hyphen, and a number that represents the patch revision number.
patch incompatibility	A rare situation where two patches cannot be on the same system. Each patch in the relationship is incompatible with the other. If you want to apply a patch that is incompatible with a patch already on the system, you must first remove the patch that is already on the system. Then, you can apply the new patch.
patch list	A file that contains a list of patches, one patch ID per line. Such a list can be used to perform patch operations. The list can be generated based on the analysis of a system or on user input. Each line in a patch list has two columns. The first column is the patch ID, and the second column is a synopsis of that patch.
patch management process	A process that involves analyzing a system to determine the appropriate patches, downloading the patches to that system, and applying the patches. Another part of the patch management process is the optional removal of patches.
patch obsolescence	An instance where a patch replaces another patch, even if it has not already been applied to a system. A patch that obsoletes one or more patches replaces those patches entirely and does not require that the obsolete patches be applied before the replacement patch is applied.
PatchPro	A product developed by Sun Network Storage to provide automated patch management technology, which is used by Sun Patch Manager.
patch server	A source of Solaris patches that can be used by your systems to perform patch analyses and from which to obtain the appropriate patches.
policy for applying patches	A user-configurable policy that specifies the types of patches that can be applied during an update of your system.

remote mode — A mode available for the `smpatch` command, which can be run on a local system to update another system with patches. This mode can only be used while the system is in multiuser mode.

resolve — To determine the patch dependencies required for a list of patches. Each patch in the list is checked to determine whether any other patches must be added to the list. If any patches are required, they are added to the ordered patch list.

signed patch — A patch that is signed with a valid digital signature. A signed patch offers greater security than an unsigned patch. The digital signature of the patch can be verified before the patch is applied to your system. A valid digital signature ensures that the signed patch has not been modified since the signature was applied. Signed patches are stored in Java Archive (JAR) format files.

standard patch — A patch that can be applied to a Solaris system that is running in multiuser mode without having to reboot. Such a patch is associated with the `standard` patch property.

Sun Alert — A notification to customers of a known product issue that might negatively impact customers' computing environments or productivity. A problem that warrants a Sun Alert notification meets the criteria for issues that are related to at least one of these concerns: availability, security, and data loss.

SunSolve Online — The Sun Microsystems web site that provides access to patch data. Patch Manager uses the data to perform patch analyses of your systems. See `http://sunsolve.sun.com`.

unsigned patch — A patch that is not signed with a digital signature.

update — To perform the steps necessary to apply patches to a system. The system is analyzed, and the patches are downloaded and then applied.

web proxy — A system that is used to connect your system to the Internet. Your system cannot connect directly to the Internet, but must use the web proxy to establish the connection.

CHAPTER 19

Managing Solaris Patches by Using Sun Patch Manager (Tasks)

This chapter describes the Sun Patch Manager tool (Patch Manager), which you can use to manage patches on your Solaris 8, Solaris 9, and Solaris 10 systems.

The following information is covered in this chapter:

You must install at least the Developer Solaris Software Group of Solaris 10 software to use the Sun Patch Manager tool. The Patch Manager software is included in the Solaris 10 release.

If you want to run Patch Manager on a Solaris 8 system, you must install at least the End User Solaris Software Group of Solaris 8 software. If you want to run Patch Manager on a Solaris 9 system, you must install at least the Entire Solaris Software Group of Solaris 9 software. You must also obtain the Patch Manager software from the Sun Download Center at `http://wwws.sun.com/software/download`.

For step-by-step instructions for managing Solaris patches by using the `patchadd` command, see Chapter 20.

Note – As of this Solaris release, not all Sun patches are available through Sun Patch Manager. Such patches include those that do not conform to PatchPro standards, and those that have third-party contract restrictions.

New Patch Manager Features

Patch Manager has been enhanced with these features:

- PatchPro analysis engine
- Local-mode command-line interface
- Patch list operations

PatchPro Analysis Engine

Patch Manager now incorporates *PatchPro* functionality to automate the patch management process. This process includes performing patch analyses on systems, then downloading and applying the resulting patches. This automation functionality was previously available for Solaris 2.6, Solaris 7, Solaris 8, and Solaris 9 as a separate PatchPro product and is now part of the standard Solaris 10 release.

PatchPro uses signed patches, which improves the security of Solaris patches by ensuring that they have not been modified.

Note – The `pprosetup` and `pprosvc` commands are included with Sun Patch Manager 2.0 for transition purposes. It is best *not* to use these commands and to use the `smpatch` command instead.

Local-Mode Command-Line Interface

Note – On Solaris 8 systems, you can only run `smpatch` in local mode.

Starting with Solaris 9, the `smpatch` command is available in two modes: local mode and remote mode. *Local mode* can only be run on the local system. This mode can be run while the system is in single-user or multiuser mode. *Remote mode* can be used to perform tasks on remote systems. Both local mode and remote mode can be used by users or roles that have the appropriate authorizations.

By default, local mode is run. In local mode, the Solaris WBEM services are not used, and none of the authentication options or those options referring to remote systems are available. The `smpatch` command in local mode runs faster than in remote mode.

If you specify any of the remote or authentication options (except for `-L`), remote mode is used.

Single-User Mode Operations in Local Mode

You can use the `smpatch add` command in local mode to apply patches while the system is in single-user mode. Apply patches in this way when the patches are associated with the `singleuser` patch property, or when you want to apply any patches to a quiet system.

Use only the `smpatch add`, `smpatch order`, and `smpatch remove` commands to manage patches when your system is running in single-user mode.

You can configure your patch management environment while the system is running in single-user mode by using the `smpatch get`, `smpatch set`, and `smpatch unset` commands.

Do not use the `smpatch analyze`, `smpatch download`, and `smpatch update` commands while the system is running in single-user mode. These commands depend on network services that are not available while the system is in single-user mode.

If you previously used the `smpatch update` command to update your system with patches, some of the patches might not have been applied. Such patches cannot be applied if they do not meet the policy for applying patches, and must be applied manually in single-user mode.

To apply the patches while the system is in single-user mode, use the `smpatch add` command with the `-x idlist=` option to specify the list of patches to apply.

You can use the `disallowed_patch_list` file as input to the `smpatch add` command to apply the `singleuser` patches. This file, stored in the *download directory*, lists any patch that could not be applied by `smpatch update` while the system was in multiuser mode. For example:

```
# smpatch add -x idlist=/var/sadm/spool/disallowed_patch_list
```

Patch List Operations

Patch Manager can create an *ordered* list of patches that you can save to a text file and use to perform patch operations.

You might use a patch list to apply the same set of patches to systems that have the same hardware and software configurations. Or, you might create a patch list file that contains all pertinent security patches and use the patch list to apply those security patches to one or more systems.

You can create a file that contains an ordered *patch list* by using the `smpatch` command in any of these ways:

- **Perform an analysis of a system –** Use the `smpatch analyze` command to *analyze* a system to generate an ordered list of patches and write it to a file. You can edit this file to remove unneeded patches.

- **Supply a specific list of patches –** Use the `smpatch analyze` command to generate an ordered list of patches based on a set of patches that you specify for a particular system. The patch list is *resolved* by augmenting the list with patches on which they depend.
- **Point to a collection of patches stored on a system –** Use the `smpatch order` command to produce an ordered list of patches based on a collection of patches stored on a system.

If you modify a patch list and the patches are available on your system, use the `smpatch order` command to put the list in an order suitable for applying patches. Otherwise, use the `smpatch analyze` command, which also produces an ordered list of patches.

You can use patch lists as input to the `smpatch add`, `smpatch analyze`, `smpatch download`, `smpatch order`, and `smpatch update` commands.

Caution – The `smpatch add` command attempts to apply all of the patches in the patch list, regardless of the policy for applying patches and *patch dependencies*.

Sun Patch Manager Concepts

Sun Patch Manager is the standard tool for managing *patches* on Solaris systems.

Patch Manager primarily operates on *signed patches*, which include a *digital signature* from Sun Microsystems. A signed patch offers greater security than an *unsigned patch*, which does not have a digital signature. The digital signature of the patch is verified before the patch is applied to your system. A valid digital signature ensures that the signed patch that you apply has not been modified since the signature was applied. You can use the `smpatch add` command to apply unsigned patches.

Patch Management Process

Patch Manager enables you to manually or automatically perform the *patch management process*, which includes the following tasks:

- Updating your system with some or all of the appropriate patches, which automatically analyzes the system to determine the appropriate patches, downloads the patches, and applies the patches to the system
- Analyzing the system to obtain a list of appropriate patches
- Downloading the appropriate patches to your system

- Applying the appropriate patches to your system
- Configuring the patch management environment for your system
- Tuning the patch management environment for your system
- Removing patches from your system

For information about recommended strategies and practices for using Solaris patches, go to `http://docs.sun.com/doc/817-0574/`.

Automatically Updating Your System With Patches

Patch Manager can automatically apply the set of appropriate patches to your system. An update performs these steps in the patch management process:

- Analyzes your system to determine which patches are appropriate
- Downloads those patches to your system
- Applies only the patches that meet the policy for applying patches

After a patch has been successfully applied, the downloaded patch is removed from the download directory.

Patches are applied to your system depending on the specified policy and the patch properties associated with the patches that are downloaded.

If a patch does not meet the *policy for applying patches*, the patch is not applied. Instead, a patch entry for that patch is written to the `disallowed_patch_list` file in the download directory. Sun Patch Manager continues trying to apply the other patches. Later, you can go to the download directory and use the `smpatch add` command to manually apply any disallowed patches that are listed in this file. For any of the patches that have the `interactive` property set, follow the instructions in the patch's README file to apply them.

For example, you can bring your system to single-user mode and apply the patches listed in the `disallowed_patch_list` file by typing the following:

```
# smpatch add -x idlist=/var/sadm/spool/disallowed_patch_list
```

Instead of performing an *update*, you can perform the analyze, download, and apply tasks manually by using the `smpatch` command. These tasks are described in the following sections.

Analyzing Your System

Before you can apply patches to your system, you can determine which patches are needed. You can use Patch Manager to perform a *patch analysis* of your system to obtain a list of appropriate patches.

Patch Manager uses analysis modules and a list of available patches from the source of patches, which is the SunSolve Online web site by default, to perform the analysis of your Solaris system. For information about the source of patches, see "Specifying the Source of Patches" on page 327.

Based on the result of the analysis, the patches can be downloaded and applied to your system.

Sometimes a patch depends on another patch, that is, the first patch cannot be applied to the system until the other patch is applied. The first patch is said to have a *dependency* on the second patch. When Patch Manager analyzes your system, it checks for patch dependencies and automatically includes all patches in the resulting list. If you request a system analysis based on particular patches, Patch Manager adds any patches to the list that are needed to resolve patch dependencies.

Note – The list of patches that is generated by the analysis is based on all of the available patches from the Sun patch server. No explicit information about your host system or its network configuration is transmitted to Sun. Only a request for the Sun patch set is transmitted. The patch set is scanned for patches that are appropriate for this host system, the results are displayed, and those patches are optionally downloaded.

Downloading Patches to Your System

Before you apply patches to your system, you must *download* the patches that you want from the Sun patch server to that system.

You can download patches from the Sun patch server based on an analysis of the system, or you can specify particular patches to download.

Applying Patches to Your System

Patch Manager can *apply* patches to your system.

If you use the `smpatch add` command to apply particular patches, it attempts to apply only those patches that you specified. The `smpatch add` command does not attempt to resolve patch dependencies. If you want to apply a patch that has a missing dependency, the patch is not applied. You can use the `smpatch analyze` command or the `smpatch update` command to resolve patch dependencies.

Removing Patches From Your System

You might want to remove (or *back out*) a patch that you previously applied to your system. Patch Manager enables you to remove patches.

When you remove a patch, the Solaris patch tools restore all of the files that have been modified by that patch, unless any of the following are true:

- The patch was applied by the `patchadd -d` command, which instructs `patchadd` *not* to save copies of files being updated or replaced.
- The patch was applied by the `patchadd` command without using the `-d` option and the backout files that were generated have since been removed.
- The patch has been *obsoleted* by a later patch.
- The patch is required by another patch.

The Solaris patch tools call the `pkgadd` command to restore *packages* that were saved when the patch was initially applied.

During the patch removal process, the `patchrm` command logs the backout process in the `/tmp/backoutlog.`*process-id* file. This log file is automatically removed if the patch is successfully removed.

Note that you can only remove *one* patch at a time when you use the `smpatch remove` command.

Note – If you attempt to remove a patch on which other patches depend, it is not removed. If you remove all of the patches that depend upon this patch, then you can remove it.

Specifying the Source of Patches

When you use Patch Manager, your client systems must have access to Solaris patches and patch data. Both client systems and local patch servers can obtain patches from these sources:

- **Patch server –** A server that provides access to Solaris patches and patch data.
- **Local collection of patches –** A collection of patches and patch data that is stored in a directory available to the local system. Such a directory might be a local directory, a shared network directory, or a CD mounted on your local system.

The default source of patches for client systems is the Sun patch server. As a result, any client system that obtains patches from the Sun patch server must be connected, either directly or through a *web proxy*, to the Internet.

You can use a combination of different patch sources to configure these patch management environments.

Clients access patches and patch data from the following sources:

- **Sun patch server –** Your client systems obtain patches from the Sun patch server.

  This configuration requires that your client systems are connected, directly or through a web proxy, to the Internet.

- **Local collection of patches –** Your client systems obtain patches and patch data from a collection of patches on your local system.

  This configuration does not require that the client systems be connected to the Internet.

For instructions on specifying the source of patches for your client system, see or "How to Specify the Source of Patches (Command Line)" on page 337.

Customizing the Policy for Applying Patches

Patch Manager enables you to customize a policy for applying patches to use when updating your system. The policy determines the types of patches that can be applied during an update operation.

Solaris patches are classified as being standard or nonstandard. A *standard patch* can be applied to your Solaris system when running in multiuser mode. A reboot is not required. Such a patch is associated with the `standard` patch property.

A *nonstandard patch* has one of the following characteristics:

- The patch is associated with one or more of the `rebootafter`, `rebootimmediate`, `reconfigafter`, `reconfigimmediate`, and `singleuser` properties. Such a nonstandard patch can be applied during an update operation if permitted by the policy.
- The patch is associated with the `interactive` property. Such a patch cannot be applied by using the `smpatch update` command. You can use the `smpatch add` command or the `patchadd` command to apply such a patch.

Note – As of this Solaris release, not all Sun patches are available through Sun Patch Manager. Such patches include those that do not conform to PatchPro standards, and those that have third-party contract restrictions.

You can specify the types of patches that Patch Manager can apply during an update. Such patches might include those that require a reboot or those that must be applied while the system is in single-user mode.

For descriptions of the following patch properties, see the `smpatch`(1M) man page.

- `interactive`
- `rebootafter`
- `reconfigafter`
- `rebootimmediate`
- `reconfigimmediate`
- `singleuser`
- `standard`

Setting Patch Manager Configuration Parameters

You can use the `smpatch` command to set the following Patch Manager parameters.

`patchpro.patchset`
: Name of the patch set to use. The default name is `patchdb`.

`patchpro.download.directory`
: Path of the directory where downloaded patches are stored and from which patches are applied. The default location is `/var/sadm/spool`.

`patchpro.backout.directory`
: Path of the directory where patch backout data is saved. When a patch is removed, the data is retrieved from this directory as well. By default, backout data is saved in the package directories.

`patchpro.patch.source`
: URL that points to the collection of patches. The default URL is that of the Sun patch server, `https://updateserver.sun.com/solaris/`.

`patchpro.sun.user`
: The Sun user name that you use to obtain patches. You obtain this user name by registering at `http://sunsolve.sun.com`. By default, you are not permitted to access contract patches.

`patchpro.sun.passwd`
: Password used with your Sun user name. No default password is set. If you specify your Sun user name, you must also specify your password.

`patchpro.proxy.host`
: Host name of your web proxy. By default, no web proxy is specified, and a direct connection to the Internet is assumed.

`patchpro.proxy.port`
: Port number used by your web proxy. By default, no web proxy is specified, and a direct connection to the Internet is assumed. The default port is `8080`.

`patchpro.proxy.user`
: Your user name used by your web proxy for authentication.

`patchpro.proxy.passwd`
: Password used by your web proxy for authentication.

`patchpro.install.types`
: Your policy for applying patches. The value is a list of zero or more colon-separated patch properties that are permitted to be applied by an update operation (`smpatch update`).

  By default, patches that have the `standard`, `rebootafter`, and `reconfigafter` properties can be applied. See "Customizing the Policy for Applying Patches" on page 328.

Getting Started With Patch Manager

To determine which method is best for downloading and applying patches to your system, see "Selecting the Best Method for Applying Patches" on page 314.

To get started using Patch Manager, find the situation that best describes your patch management environment.

- Your client system is directly connected to the Internet.

  You are ready to manage patches by using Patch Manager. See "Accessing the Sun Patch Manager Command-Line Interface" on page 332.

- Your client system is connected to the Internet by means of a web proxy.

  You must first specify the host name and port of the web proxy. If required, also specify the user name and password associated with the web proxy. See "How to Specify Your Web Proxy (Command Line)" on page 335.

  After you change your configuration, see "Accessing the Sun Patch Manager Command-Line Interface" on page 332.

- You need a user name and a password to access patches from the Sun patch server.

  If you need to obtain a user name and password, register at `http://sunsolve.sun.com`.

  Then, specify the user name and password for each client system on which you run Patch Manager. See "How to Specify a User Name and Password With Which to Obtain Patches (Command Line)" on page 336.

  After you change your configuration, see "Accessing the Sun Patch Manager Command-Line Interface" on page 332.

Tasks Supported by Sun Patch Manager

The following tasks are supported by Sun Patch Manager:

- Performing patch management operations on a remote system

  You can use the `smpatch` command in remote mode to perform patch management operations on a remote system. In local mode, the `smpatch` command can only be run on the local system.

- Analyzing a system for patches
- Performing scheduled patch analyses

  Use the `cron` command to run the `smpatch analyze` command.

- Downloading individual patches
- Resolving patch dependencies

Run `smpatch update` or `smpatch analyze -i` *patch-id* to resolve patch dependencies. Note that if you run `smpatch add`, patch dependencies are *not* resolved.

- Updating a system with patches.
- Running while the system is in single-user mode (limited operations of local mode with `smpatch` only).
- Operating on patch lists
- Configuring the patch management environment for your system

Managing Solaris Patches by Using the Sun Patch Manager Command-Line Interface (Task Map)

The following table identifies the tasks that you might perform when you use the Sun Patch Manager command-line interface.

Task	Description	For Instructions
Access the command-line interface.	If you want to perform patch management tasks on the command-line, use the `smpatch` command.	"Accessing the Sun Patch Manager Command-Line Interface" on page 332
Configure the patch management environment for your system.	By default, your system is assumed to be connected directly to the Internet and configured to obtain patches from the Sun patch server. If this is not true for your system, change the configuration settings to match your environment.	"Configuring Your Patch Management Environment by Using the Command-Line Interface (Task Map)" on page 334
Manage patches on your system.	You can use the command-line interface to perform an analysis of your system, apply one or more patches, find patch dependencies, order patch lists, and remove patches.	"Managing Patches by Using the Command-Line Interface (Task Map)" on page 338

Task	Description	For Instructions
(Optional) Tune the patch management environment for your system.	Change some optional configuration settings, such as the policy for applying patches.	"Tuning Your Patch Management Environment by Using the Command-Line Interface (Task Map)" on page 348

Accessing the Sun Patch Manager Command-Line Interface

Caution – Do *not* run simultaneous Patch Manager operations on your system because it might become unstable. Do not interrupt a patch operation once it has started. If a patch operation is running, you must wait for that operation to complete before starting another operation.

You can run either the local mode or remote mode `smpatch` command as a user with the appropriate authorizations, such as superuser, or by assuming a role that includes the appropriate profiles.

A user must have the `solaris.admin.patchmgr.*` authorization to run the `smpatch` command.

The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

Note – The `pprosetup` and `pprosvc` commands are included with Sun Patch Manager 2.0 for transition purposes. It is best *not* to use these commands and to use the `smpatch` command instead.

For more information about `smpatch` command-line options, see the `smpatch`(1M) man page.

▼ How to Access the Sun Patch Manager Command-Line Interface (Command Line)

By default, the `smpatch` command runs in local mode.

Steps

1. **Decide whether to manage patches on the local system or on a remote system.**
   - If you want to manage patches on the local system only, go to Step 2.
   - If you want to manage patches on a remote system, go to Step 4.

     The Solaris WBEM services must be running on the remote system.
2. **Log in to a system as a user with appropriate authorizations or assume a role with the appropriate authorizations.**

   Note that you must be an appropriately authorized user to assume an appropriate role. See "Configuring RBAC" in *System Administration Guide: Security Services*.
3. **Run the `smpatch` command you want.**

   For example:

   ```
   $ smpatch analyze
   ```
4. **Log in to a system as a user who is appropriately authorized or is permitted to assume a role that is appropriately authorized.**
5. **Run the `smpatch` command with the `-n` option to specify the name of the system on which to operate.**

   For example:

   ```
   $ smpatch analyze -n system-name
   ```

   To perform the operation with an assumed role, type:

   ```
   $ smpatch analyze -r role-name -n system-name
   ```

Example 19–1 Accessing the Sun Patch Manager Command-Line Interface

The following examples use the `smpatch get` command, which lists the configuration settings for your patch management environment.

This example shows how to run the `smpatch` command on the local system.

```
# smpatch get
```

This example shows how an authorized user can run the `smpatch` command on the remote system called `jupiter`.

```
# smpatch get -n jupiter
```

This example shows how you can run the `smpatch` command on the remote system called `jupiter` as the role `patcher`.

```
# smpatch get -r patcher -n jupiter
```

More Information

What to Do Next

You can use the `smpatch` command to configure the patch management environment for your system and manage patches. See the following:

- "Configuring Your Patch Management Environment by Using the Command-Line Interface (Task Map)" on page 334
- "Managing Patches by Using the Command-Line Interface (Task Map)" on page 338
- "Tuning Your Patch Management Environment by Using the Command-Line Interface (Task Map)" on page 348

Configuring Your Patch Management Environment by Using the Command-Line Interface (Task Map)

Use the `smpatch` command to perform the configuration tasks in this section. For the list of configuration parameters you can set, see "Setting Patch Manager Configuration Parameters" on page 329 and the `smpatch`(1M) man page.

By default, the patch management environment is configured to obtain patches directly from the Sun patch server. Therefore, you must customize your environment if your system does one or more of the following:

- Connects to the Internet by means of a web proxy
- Requires a user name and password to obtain patches
- Obtains patches from a patch source other than the Sun patch server

The following table identifies the tasks that you might perform when you configure the patch management environment for your system.

Task	Description	For Instructions
Specify the web proxy to use.	If your system is connected to the Internet through a web proxy, you must specify the web proxy that is used to access the Sun patch server. By default, no web proxy is specified.	"How to Specify Your Web Proxy (Command Line)" on page 335

Task	Description	For Instructions
Specify the user and password needed to provide authentication for the web proxy.	If your web proxy requires authentication, you must specify the web proxy user that is needed for authentication. By default, no web proxy user is specified.	"How to Specify Your Web Proxy (Command Line)" on page 335
Specify the user and password needed to obtain patches from the Sun patch server.	If you needed a user and password to obtain patches, you must specify the user name and password.	"How to Specify a User Name and Password With Which to Obtain Patches (Command Line)" on page 336
Specify the source of patches for your system.	Your system can obtain patches from one of the following sources: ■ Sun patch server ■ Local collection of patches The default source of patches for your system is the Sun patch server.	"How to Specify the Source of Patches (Command Line)" on page 337

Note – The following procedures and examples show how to run the local mode `smpatch` command, which is run by default. To run the remote mode version, use any of the authentication options (except for `-L`) or the remote options. See the `smpatch`(1M) man page.

▼ How to Specify Your Web Proxy (Command Line)

If your system connects to the Internet through a web proxy, you must provide information about the web proxy to Patch Manager.

Steps

1. **Obtain the host name and the port of the web proxy from your network administrator.**

2. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

3. **Specify the web proxy.**

   ```
   # smpatch set patchpro.proxy.host=web-proxy-server \
   patchpro.proxy.port=port
   ```

4. **(Optional) If the web proxy requires authentication, supply the user name and password.**

   Obtain this information from your network administrator.

   a. **Specify the user name to be used for authentication.**

      # **smpatch set patchpro.proxy.user=***web-proxy-user*

   b. **Specify the proxy user's password by having `smpatch` prompt you for the password.**

      # **smpatch set patchpro.proxy.passwd**
      Web Proxy User Password: *web-proxy-password*

      Setting the password in this way ensures that the password you type does not appear as clear text in the following:

      - Standard output
      - Output of the `ps` command
      - Your shell history file

▼ How to Specify a User Name and Password With Which to Obtain Patches (Command Line)

If you needed a user name and password to obtain patches from the Sun patch server, you must specify them for Patch Manager.

If you do not have an account on SunSolve, register for one at `http://sunsolve.sun.com`.

As of this Solaris release, a user name and password are not required to obtain patches from the Sun patch server.

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

2. **Specify your user name.**

   # **smpatch set patchpro.sun.user=***user-name*

3. **Specify the password for your user by having `smpatch` prompt you for the password.**

   # **smpatch set patchpro.sun.passwd**
   Sun User Password: *password*

Setting the password in this way ensures that the password you type does not appear as clear text in the following:

- Standard output
- Output of the `ps` command
- Your shell history file

▼ How to Specify the Source of Patches (Command Line)

Your system can obtain patches from the following sources:

- Sun patch server
- Local patch collection

By default, your system obtains patches from the Sun patch server.

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

2. **Specify the URL of the patch source.**

   - For the Sun patch server, type:

     ```
     # smpatch unset patchpro.patch.source
     ```

   - For a collection of patches in a directory, use this URL format:

     ```
     # smpatch set patchpro.patch.source=file:/directory-name
     ```

     Note that *directory-name* can be a local file system or a remotely mounted file system.

   See Example 19–2 for examples of using the `file:/` URL format.

Example 19–2 Specifying the Source of Patches

The following example shows how to configure a system to obtain patches from the `/export/patches` directory on the local system.

```
# smpatch set patchpro.patch.source=file:/export/patches
```

The following example shows how to configure a system to obtain patches from the `/export/patches` directory on the remote system called `jupiter`.

```
# smpatch set patchpro.patch.source=file:/net/jupiter/export/patches
```

The following example shows how to configure a system to obtain patches from a CD mounted from the first CD-ROM drive of the local system.

```
# smpatch set patchpro.patch.source=file:/cdrom/cdrom0
```

More Information

What to Do Next

After you specify a patch source, your client system is ready to manage patches. See "Managing Patches by Using the Command-Line Interface (Task Map)" on page 338.

Managing Patches by Using the Command-Line Interface (Task Map)

Use the `smpatch` command to perform most of the common patch management tasks described in the following table. See the `smpatch`(1M) man page.

Task	Description	For Instructions
Analyze your system to determine the list of patches.	You want to analyze your system to obtain the list of appropriate patches. Based on the analysis, you can update your system with one or more patches in the list.	"How to Analyze Your System to Obtain the List of Patches to Apply (Command Line)" on page 340
Automatically update your system with one or more patches in a single procedure.	You want to automatically download and apply the patches that are appropriate for your system. The list of patches is determined by having Patch Manager analyze your system.	"How to Update Your System With Patches (Command Line)" on page 341
Apply patches to your system.	■ After you have determined the patches to apply and have downloaded them to your system, you can apply them.	"How to Apply Patches to Your System (Command Line)" on page 342

Task	Description	For Instructions
	■ Some patches should be applied while the system is in single-user mode because they might cause the system to become unstable. Such patches are associated with the `singleuser` patch property. In single-user mode, you must use the `smpatch add` command to apply patches.	"How to Apply Patches to Your System (Command Line)" on page 342
	■ Some patches are nonstandard and must be applied manually.	"How to Apply a Nonstandard Patch (Command Line)" on page 344 "How to Download and Apply a Solaris Patch" on page 360
	■ (Optional) Determine whether the patches you want to apply depend on others being applied first.	"How to Resolve a List of Patches (Command Line)" on page 344
Remove patches from your system.	You want to remove, or back out, patches that you applied to your system.	"How to Remove Patches From Your System (Command Line)" on page 347
(Optional) View patch management tool log entries.	View Patch Manager log entries in the system log file to identify problems with installing patch management tools or applying patches.	"How to View Patch Manager Log Entries (Command Line)" on page 347
Apply patches to an inactive boot environment on your system by using `luupgrade`.	You want to use Solaris Live Upgrade to apply patches to a system that has more than one boot environment.	"How to Use `luupgrade` to Apply a List of Patches to an Inactive Boot Environment (Command Line)" on page 345

Note – The following procedures and examples show how to run the local mode `smpatch` command, which is run by default. To run the remote mode version, use any of the authentication options (except for `-L`) or the remote options. See the `smpatch`(1M) man page.

▼ How to Analyze Your System to Obtain the List of Patches to Apply (Command Line)

You can perform an analysis of your system to determine the list of appropriate patches. The list is in an order that can be used to apply patches. You can also supply a list of one or more patches as input to restrict the analysis to just those patches. In addition to performing the analysis, you can save the patch list for modification or later use.

The system analysis writes the list of patches to standard output, so you can save the contents of the patch list to a file by redirecting standard output to a file.

Each line in a patch list has two columns. The first column is the patch ID, and the second column is a synopsis of that patch.

If you supply a list of one or more patches to the `smpatch analyze` command, the list of patches is augmented with any patches that are required as dependencies.

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

2. **Perform a patch analysis of your system and optionally save the list of patches in a file.**

   - To create a list of all the appropriate patches for your system, type:

     ```
     # smpatch analyze
     ```

   - To create a list of particular patches for your system based on a patch list, type:

     ```
     # smpatch analyze -x idlist=patch-list-file
     ```

   - To create a list of particular patches for your system, type:

     ```
     # smpatch analyze -i patch-id...
     ```

Example 19–3 Analyzing Your System to Obtain the List of Patches to Apply

The following example shows how to analyze a system to create a list of all appropriate patches. The list is written to the `/tmp/patch.all` file.

```
# smpatch analyze > /tmp/patch.all
```

The following example shows how to create a list of patches, `plist`, modify it, and resolve the patch dependencies. The list is written to the `/tmp/patch.plist` file.

```
# smpatch analyze > plist
# vi plist
```

.
.
.
```
# smpatch analyze -x idlist=plist > /tmp/patch.plist
```

The following example shows how to resolve patch dependencies for patch 112785-28 and write the resulting patch list to a file called `/tmp/patch.out`. Patch 112785-28 depends on patch 113096-03. After running the `smpatch analyze` command, the `patch.out` file contains this ordered list: 113096-03 and 112785-28.

```
# smpatch analyze -i 112785-28 > /tmp/patch.out
```

▼ How to Update Your System With Patches (Command Line)

An update of a system performs the entire patch management process in one step. First, the analysis determines the appropriate patches for your system. Next, those patches are downloaded to your system. Finally, the patches are applied to your system.

All standard patches are applied by an update. You can configure your system to apply some nonstandard patches by changing the default policy for applying patches. To change the policy for your system, see "How to Change the Policy for Applying Patches (Command Line)" on page 351.

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

2. **Update the system with patches in one of the following ways:**

   - To update your system with all appropriate patches, type:

     ```
     # smpatch update
     ```

   - To update your system with all patches listed in a file, first create a patch list (see "How to Analyze Your System to Obtain the List of Patches to Apply (Command Line)" on page 340), then type:

     ```
     # smpatch update -x idlist=patch-list-file
     ```

   - To update your system with particular patches, type:

     ```
     # smpatch update -i patch-id -i patch-id ...
     ```

   If you specify particular patches by using the `-i` or `-x idlist=` options, the list is augmented with patches on which they depend before the update occurs.

Note – Any patches that cannot be applied to the system are listed in a patch list file called `disallowed_patch_list`, which is located in the download directory. You can use this file as input to the `smpatch add` command.

For example, you might bring your system to single-user mode and apply the patches listed in the `disallowed_patch_list` file by typing the following:

```
# init S
# smpatch add -x idlist=/var/sadm/spool/disallowed_patch_list
```

See "How to Apply Patches to Your System (Command Line)" on page 342 for more information.

Example 19–4 Updating Your System With Patches

The following example shows how to update a system with patch 112622-12 and 112771-17.

```
# smpatch update -i 112622-12 -i 112771-17
```

The following example shows how to update a system by using a list of patches, named `plist`, as input. It then shows how to create a patch list and modify it to contain only the patches that you want to apply to your system. Then, use the `smpatch update` command to apply the patches and update the system.

1. Create a list of patches by performing an analysis.
2. Edit the patch list to include only the patches that you want to apply.
3. Run the `smpatch update` command to apply the patches.

   For example:

```
# smpatch analyze > plist
.
.
.
# vi plist
.
.
.
# smpatch update -x idlist=plist
.
.
.
```

▼ How to Apply Patches to Your System (Command Line)

You can use the `smpatch add` command to apply one or more downloaded patches to your system.

You can use the local mode version of the `smpatch` command to apply one or more downloaded patches while your system is in single-user mode or in multiuser mode.

Caution – The `smpatch add` command ignores the policy for applying patches and does not resolve dependencies when applying patches.

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

2. **Apply the downloaded patches to your system.**

   - To apply all patches listed in a file, type:

     ```
     # smpatch add -x idlist=patch-list-file
     ```

   - To apply particular patches, type:

     ```
     # smpatch add -i patch-id -i patch-id ...
     ```

   - To apply particular patches that have the `singleuser` property, you must first bring the system to single-user mode. Type:

     ```
     # init S
     # smpatch add -i patch-id -i patch-id ...
     ```

   - To apply the list of patches that could not be applied by the `smpatch update` command, you must first bring the system to single-user mode. Type:

     ```
     # init S
     # smpatch add -x idlist=/var/sadm/spool/disallowed_patch_list
     ```

Example 19–5 Applying Patches to Your System

- The following example shows how to apply the patches listed in the file `plist` while the system is in single-user mode.

```
Requesting System Maintenance Mode
SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode Entering System Maintenance Mode
.
.
.
# smpatch add -x idlist=plist
```

- The following example shows how to apply patch 112662-12 while the system is in single-user mode.

```
Requesting System Maintenance Mode
SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxxx
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode Entering System Maintenance Mode
.
.
.
# smpatch add -i 112662-12
```

▼ How to Apply a Nonstandard Patch (Command Line)

You cannot use `smpatch` to apply nonstandard patches that have the `interactive` property set. To apply the patch, review the information in the Special Installation Instructions section of the patch's README file.

Steps

1. **Become superuser.**
2. **In the download directory, find the nonstandard patch that you want to apply.**

   ```
   # cd /var/sadm/spool; ls
   ```

3. **To access the patch README file, do one of the following:**
   - View the patch README file from the Sun patch server at `http://sunsolve.sun.com`.
   - To extract the patch README file from the JAR archive, do the following:
     a. Identify the name of the README file.
     b. Extract the README file.
     c. View the README file.
4. **Follow the instructions in the Special Installation Instructions section of the README file to apply the patch.**

▼ How to Resolve a List of Patches (Command Line)

Sometimes a patch depends on another patch, that is, the first patch cannot be applied to the system until the other patch is applied. The first patch is said to have a dependency on the second patch.

If you specify a list of patches to apply, you can resolve the list for patch dependencies. The resulting list might include additional patches that you must apply before applying the patches you specified.

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

2. **Resolve the list of patches.**

   - Resolve a list of patches specified one at a time on the command line.

     ```
     # smpatch analyze -i patch-id -i patch-id ...
     ```

   - Resolve a list of patches specified in a file.

     ```
     # smpatch analyze -x idlist=patch-list-file
     ```

Example 19–6 Resolving a List of Patches

The following example shows how to resolve patch dependencies for patch 112785-28 and write the resulting patch list to a file called `/tmp/patch.out`. Patch 112785-28 depends on patch 113096-03. After running the `smpatch analyze` command, the `patch.out` file contains this ordered list: 113096-03 and 112785-28.

```
# smpatch analyze -i 112785-28 > /tmp/patch.out
```

The following example shows how to take a modified list of patches, `plist`, and resolve the patch dependencies. The list is written to the `/tmp/patch.plist` file.

```
# smpatch analyze -x idlist=plist > /tmp/patch.plist
```

▼ How to Use `luupgrade` to Apply a List of Patches to an Inactive Boot Environment (Command Line)

A patch list that is created by the `smpatch` command can be used by `luupgrade` to apply patches to an inactive boot environment. You can also use the `luupgrade` command to remove patches from an inactive boot environment based on `showrev` information. See the `luupgrade`(1M) and `showrev`(1M) man pages.

Note – This procedure assumes that you have created a second boot environment that is a duplicate of the active boot environment. See the `lumake`(1M) man page for information about creating boot environments.

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

2. **Perform a patch analysis on the active boot environment to obtain the list of appropriate patches to apply to the inactive boot environment, and remove the synopsis for each patch entry.**

   # **smpatch analyze | sed 's/ .*//' >** *patch-list-file*

   The modified file will be a list of patches, one *patch ID* per line.

3. **Download the patches from a patch list to your system.**

   # **smpatch download -x idlist=***patch-list-file*

4. **Apply patches from a patch list to the inactive boot environment.**

   # **luupgrade -t -n** *BE-name* **-s** *dir-name* **`cat** *patch-list-file***`**

   You must specify the name of the inactive boot environment to update, *BE-name*, and the directory where the patches are stored, *dir-name*.

5. **(Optional) To remove a patch from the inactive boot environment, use the following command:**

   # **luupgrade -T -n** *BE-name patch-id*

   You must specify the name of the inactive boot environment to update, *BE-name*, and the patch to be removed, *patch-id*.

Example 19–7 Using `luupgrade` to Apply a List of Patches to an Inactive Boot Environment

- The following example shows how to use Patch Manager and Solaris Live Upgrade commands to apply a list of patches to an inactive boot environment. For this example, a duplicate boot environment, `be2`, of the active boot environment has been created.

  First, use the `smpatch analyze` and `sed` commands to analyze the active boot environment and create a patch list, `plist`, that includes one patch ID per line. The `sed` command removes the synopsis from each patch entry. Use the `smpatch download` command to download the patches in the list. Then, use the `luupgrade` command to apply the list of patches to the inactive boot environment of the system. The inactive boot environment is called `be2`, and the directory where the patches reside is `/var/sadm/spool` on the active boot environment.

```
# smpatch analyze | sed 's/ .*//' > plist
.
.
.
# smpatch download -x idlist=plist
.
.
```

```
.
# luupgrade -t -n be2 -s /var/sadm/spool `cat plist`
.
.
.
```

- The following example shows how to use Patch Manager and the Solaris Live Upgrade commands to remove a patch from an inactive boot environment. For this example, a duplicate boot environment, `be2`, of the active boot environment has been created.

  Use the `luupgrade` command to remove patch 107058-01 from the inactive boot environment of the system, `be2`.

```
# luupgrade -T -n be2 107058-01
.
.
.
```

▼ How to Remove Patches From Your System (Command Line)

You can remove only one patch at a time.

If your system has more than one boot environment, you can use the `luupgrade` command to remove a list of patches from an inactive boot environment. See "How to Use `luupgrade` to Apply a List of Patches to an Inactive Boot Environment (Command Line)" on page 345.

Steps

1. **Identify the patch that you want to remove.**
2. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.
3. **Remove the patch from your system.**

   # **smpatch remove -i** *patch-id*

▼ How to View Patch Manager Log Entries (Command Line)

Patch Manager writes to the system log file `/var/adm/messages`.

Steps

1. **Choose which method to use to see information about a failed installation of a patch.**
   - `/var/adm/messages` – Identifies problems that are found when applying a patch to a system by using Patch Manager.
   - **Solaris WBEM log –** To view this log from the command line, use the `smlog view` command. See the `smlog`(1M) man page.
2. **View log entries from the appropriate log file.**

Tuning Your Patch Management Environment by Using the Command-Line Interface (Task Map)

The following table identifies the optional tasks that you might perform when you tune the patch management environment for your system.

Use the `smpatch` command to tune your patch management environment. For the list of configuration parameters you can set, see "Setting Patch Manager Configuration Parameters" on page 329 and the `smpatch`(1M) man page.

The following are optional tasks that you can perform with Sun Patch Manager.

Task	Description	For Instructions
Obtain configuration information about your patch management environment.	View the configuration of your patch management environment, which might help you diagnose problems.	"How to View the Configuration Settings for Your Patch Management Environment (Command Line)" on page 349

Task	Description	For Instructions
Change the policy for applying patches for your system.	Patch Manager can update your system with standard patches automatically. If you want to update your system with some types of nonstandard patches, you must change your policy for applying patches. By default, only patches that are associated with the `standard`, `rebootafter`, or `reconfigafter` properties are applied by an update operation.	"How to Change the Policy for Applying Patches (Command Line)" on page 351
Change the patch set to use for system analysis.	Patch Manager bases analyses on all available Sun patches. If you want to apply only patches from a different patch set, such as the Recommended Patch Cluster, you must change the patch set.	"How to Change the Patch Set (Command Line)" on page 352
Set different directory locations.	You might want to specify a different location for the download directory or the backout directory if the default locations are not large enough.	"How to Change Directory Locations (Command Line)" on page 352
Reset configuration parameters to the default values.	You might want to reset configuration parameters to the default values. Note that some configuration parameters have an empty default value.	"How to Reset Configuration Parameters to the Default Values (Command Line)" on page 353

Note – The following procedures and examples show how to run the local mode `smpatch` command, which is run by default. To run the remote mode version, use any of the authentication options (except for `-L`) or the remote options. See the `smpatch`(1M) man page.

▼ How to View the Configuration Settings for Your Patch Management Environment (Command Line)

You can check the configuration settings of your patch management environment to help diagnose problems or to understand your system's patch-related settings.

The configuration settings output shows an entry for all configuration parameters. Each entry appears on a line by itself.

When you list all settings, each entry includes three fields: the parameter name, the value you have assigned, and its default value. The fields are separated by one or more tab characters.

The following values have special meaning:

- `-` means that no value is set
- `""` means that the value is the null string
- `\-` means that the value is `-`
- `\""` means that the value is `""` (two double quotes)

In addition to these special values, these special characters might appear in the output:

- `\t` for a tab
- `\n` for a newline
- `\\` for a backslash

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

2. **List the configuration settings for your patch management environment.**

   - To list all settings, type:

     ```
     # smpatch get
     ```

   - To list the values for one or more parameters, type:

     ```
     # smpatch get parameter-name...
     ```

Example 19–8 Viewing Configuration Settings for Your Patch Management Environment

The following example shows how to list all the configuration settings for your patch management environment.

```
# smpatch get
patchpro.backout.directory  -      ""
patchpro.download.directory -      /var/sadm/spool
patchpro.install.types      -      rebootafter:reconfigafter:standard
patchpro.patch.source       -      https://updateserver.sun.com/solaris/
patchpro.patchset           -      patchdb
patchpro.proxy.host         -      ""
patchpro.proxy.passwd       ****   ****
patchpro.proxy.port         -      8080
patchpro.proxy.user         -      ""
patchpro.sun.passwd         ****   ****
patchpro.sun.user           -      ""
```

The following example shows how to list the configuration settings for the `patchpro.download.directory` and `patchpro.patchset` parameters.

```
# smpatch get patchpro.download.directory patchpro.patchset
/var/sadm/spool
patchdb
```

▼ How to Change the Policy for Applying Patches (Command Line)

If you want to configure your system to apply some nonstandard patches during an update operation, you must change the policy for applying patches.

By default, only patches that are associated with the `standard`, `rebootafter`, or `reconfigafter` properties can be applied by an update operation.

Caution – If you change your policy from the default, Sun makes no guarantees that the patches apply correctly to your system or that your system will function properly.

For more information about the policy for applying patches, see "Customizing the Policy for Applying Patches" on page 328.

Steps

1. **Determine the types of nonstandard patch properties that you want to apply during an update.**
2. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.
3. **Specify the new policy.**

   ```
   # smpatch set patchpro.install.types=patch-property
   ```

   patch-property is a list of patch properties each separated by a colon (`:`). For the list of valid patch properties, see "Customizing the Policy for Applying Patches" on page 328.

Example 19–9 Changing the Policy for Applying Patches

This example shows how to set the policy for a system. The new policy also includes patches that require that the system be rebooted immediately for the patch to take effect.

```
# smpatch set \
patchpro.install.types=standard:rebootafter:reconfigafter:rebootimmediate
```

▼ How to Change the Patch Set (Command Line)

You can choose to analyze your system based on different sets of Sun patches, such as the Recommended Patch Cluster. By default, you use the patch set All Available Patches.

As of this Solaris release, the only patch sets available from Sun are All Available Patches and Recommended Patch Cluster.

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.

2. **Specify the patch set to use.**

   - To base your analysis on all patches, type:

     ```
     # smpatch set patchpro.patchset=patchdb
     ```

   - To base your analysis on recommended patches, type:

     ```
     # smpatch set patchpro.patchset=recommended
     ```

   - To base your analysis on another patch set, type:

     ```
     # smpatch set patchpro.patchset=patch-set
     ```

▼ How to Change Directory Locations (Command Line)

Patch Manager is configured to use these default locations for storing patch-related data:

- **Download directory –** Directory in which patches are stored when they are downloaded from the patch source. This is also the directory from which patches are applied. Patches remain in this directory until they are successfully applied. The default location is `/var/sadm/spool`.
- **Backout data directory –** Directory in which data that enables a patch to be backed out is stored. By default, *backout data* is stored in the default locations used by `patchadd`. This is the `save` directory of each package that was modified by the patch. For example, if a patch modifies the `SUNWcsr` package, the backout data for

that package is stored in the `/var/sadm/pkg/SUNWcsr/save` directory.

If you run out of available disk space in the default locations, specify different locations for these directories.

Note – If you specify a different directory, you must manually create that directory before performing any patch operations.

Steps

1. **Determine the new locations for the directories.**
2. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.
3. **Specify a new directory, *dir-name*, for any of the patch-related directories.**
   - To specify a different download directory, type:

     ```
     # smpatch set patchpro.download.directory=dir-name
     ```

     where *dir-name* is `/export/patches`, for example.
   - To specify a different *backout directory*, type:

     ```
     # smpatch set patchpro.backout.directory=dir-name
     ```

     where *dir-name* is `/export/patches/backout`, for example.

▼ How to Reset Configuration Parameters to the Default Values (Command Line)

You must reset parameter values explicitly. You cannot use the `smpatch` command to reset all parameter values at once.

Steps

1. **Become an appropriately authorized user or assume a role that includes the Software Installation profile or the `solaris.admin.patchmgr.*` authorization.**

   The System Administrator profile includes the appropriate profiles. To create the role and assign the role to a user, see "Configuring RBAC" in *System Administration Guide: Security Services*.
2. **Reset a configuration parameter for your patch management environment to its default value.**

   ```
   # smpatch unset parameter-name...
   ```

Example 19–10 Resetting Configuration Parameters to the Default Values

The following example shows how to configure a system to obtain patches from the Sun patch server instead of from a different patch source.

```
# smpatch unset patchpro.patch.source
```

The following example shows how to reset the patch download directory and the backout directory locations to the default values.

```
# smpatch unset patchpro.download.directory patchpro.backout.directory
```

Patch Manager Troubleshooting

This section describes common problems that you might encounter when using Patch Manager to perform the following tasks:

- Analyze systems to determine the list of appropriate patches
- Download the patches to the system
- Apply the patches to the system

Additional troubleshooting information about Sun Patch Manager 2.0 might appear in the *Solaris 10 Release Notes*.

Patch Manager General Errors

Cannot Update Patches Due to Network or Server Failures

Description: When running the `smpatch update` command, any of the following errors appear:

```
Cannot connect to retrieve patchdb: Connection refused
```

Or:

```
Cannot connect to retrieve patchdb: Connection timed out
```

Or:

```
Unknown host (host-name) connecting to http://host-name/
```

Cause: This problem might be caused by a network failure between the client and the patch server, or the patch server is down.

Workaround: Ensure that `patchpro.patch.source` points to a valid patch source.

Check the condition of the network.

- If the problem is between your system and Sun, wait for the issue to be resolved.

Solaris WBEM Services Are Unavailable When Using `smpatch` in Remote Mode

Description: When running `smpatch` in remote mode or trying to restart the Solaris WBEM services, the following error messages appear:

```
# smpatch analyze -u root
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default
denoted by [ ]
Please enter a string value for: password :: root-password
There is no Solaris Management Console Server running on
pserver2.
# /etc/init.d/init.wbem status
Solaris Management Console server not running on port 898.
# /etc/init.d/init.wbem start
# /etc/init.d/init.wbem status
Solaris Management Console server not running on port 898.
```

Workaround: Manually stop the Solaris WBEM services before restarting them.

```
# /etc/init.d/init.wbem stop
# /etc/init.d/init.wbem start
# /etc/init.d/init.wbem status
Solaris Management Console server version 2.1.0 running on
port 898.
```

Solaris 10: Java Virtual Machine Cannot Be Initialized

Description: When you run any `smpatch` subcommand, you see the following error message:

```
# smpatch analyze
Error occurred during initialization of VM
java.lang.Error: Properties init: Could not determine current
working directory.
```

Workaround: Change directories and retry the command.

```
# cd /
# smpatch analyze
```

CHAPTER 20

Managing Solaris Patches by Using the `patchadd` Command (Tasks)

This chapter provides step-by-step instructions on how to manage Solaris patches by using the `patchadd` command. For additional information, see the `pkgadd`(1M) man page.

For overview information about managing Solaris patches, see Chapter 18.

For information about the Sun Patch Manager tool (Patch Manager) and for step-by-step instructions on using Patch Manager to manage patches, see Chapter 19.

Managing Solaris Patches by Using the `patchadd` Command (Task Map)

Task	Description	For Instructions
1. (Optional) Set up the package keystore.	If you plan to apply signed patches to your system, you must first import Sun's Root CA certificate into your package keystore.	"How to Import a Trusted Certificate to Your Package Keystore" on page 358
2. (Optional) Specify a web proxy.	If your system is behind a firewall with a web proxy, you must specify the web proxy to obtain patches from the Sun patch server.	"How to Specify a Web Proxy" on page 359
3. Download and apply a patch.	You can download and apply a patch to your system by using the `patchadd` command.	"How to Download and Apply a Solaris Patch" on page 360

Task	Description	For Instructions
4. (Optional) Display information about patches that have been applied to your system.	If you want information about the patches that have already been applied to your system, use the `patchadd`, `showrev`, or `pkgparam` command.	"How to Display Information About Solaris Patches" on page 361
5. (Optional) Remove a patch from your system.	If necessary, remove a patch from your system by using the `patchrm` command.	"How to Remove a Solaris Patch by Using the `patchrm` Command" on page 362

▼ How to Import a Trusted Certificate to Your Package Keystore

To apply *signed patches* to your system by using the `patchadd` command, you must add Sun's Root CA certificate, at the very least, to verify the signature of your signed patch. You can import this certificate from the Java *keystore* to the package keystore.

Steps

1. **Become superuser or assume an equivalent role.**

2. **Export the Root CA certificate from the Java keystore to a temporary file.**

   For example:

```
# keytool -export -storepass changeit -alias gtecybertrustca \
-keystore gtecybertrustca -keystore /usr/j2se/jre/lib/security/cacerts \
-file /tmp/root.crt
Certificate stored in file </tmp/root.crt>
```

`-export`	Exports the trusted certificate.
`-storepass` *storepass*	Specifies the password that protects the integrity of the Java keystore.
`-alias gtecybertrustca`	Identifies the alias of the trusted certificate.
`-keystore` *certfile*	Specifies the name and location of the keystore file.
`-file` *filename*	Identifies the file in which to hold the exported certificate.

3. **Import the Root CA certificate from the temporary file to the package keystore.**

   For example:

```
# pkgadm addcert -t -f der /tmp/root.crt
Enter Keystore Password: storepass
      Keystore Alias: GTE CyberTrust Root
         Common Name: GTE CyberTrust Root
```

```
    Certificate Type: Trusted Certificate
  Issuer Common Name: GTE CyberTrust Root
      Validity Dates: <Feb 23 23:01:00 2004 GMT>-<Feb 23 23:59:00 ...
     MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
    SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91...

Are you sure you want to trust this certificate? yes
Trusting certificate <GTE CyberTrust Root>
Type a Keystore protection Password.
Press ENTER for no protection password (not recommended):
For Verification: Type a Keystore protection Password.
Press ENTER for no protection password (not recommended):
Certificate(s) from </tmp/root.crt> are now trusted
```

`-t` Indicates that the certificate is a trusted CA certificate. The command output includes the certificate details, which you are asked to verify.

`-f` *format* Specifies the format of the certificate or private key. When importing a certificate, it must be encoded using either the PEM (`pem`) or binary DER (`der`) format.

certfile Specifies the file that contains the certificate.

4. **Display the certificate information.**

```
# pkgadm listcert
Enter Keystore Password: storepass
    Keystore Alias: GTE CyberTrust Root
       Common Name: GTE CyberTrust Root
  Certificate Type: Trusted Certificate
Issuer Common Name: GTE CyberTrust Root
    Validity Dates: <Feb 23 23:01:00 2004 GMT>-<Feb 23 23:59:00 2006 GMT>
   MD5 Fingerprint: C4:D7:F0:B2:A3:C5:7D:61:67:F0:04:CD:43:D3:BA:58
  SHA1 Fingerprint: 90:DE:DE:9E:4C:4E:9F:6F:D8:86:17:57:9D:D3:91:
BC:65:A6:89:64
```

5. **Remove the temporary file.**

```
# rm /tmp/root.crt
```

▼ How to Specify a Web Proxy

If your system is behind a firewall with a web proxy, you must specify the web proxy to use `patchadd` to *apply* a patch.

Steps

1. **Become superuser or assume an equivalent role.**
2. **Use one of the following methods to specify a web proxy:**
   - Specify the web proxy by using the `http_proxy`, `HTTPPROXY`, or `HTTPPROXYPORT` environment variable.

For example:

```
# setenv http_proxy http://mycache.domain:8080
```

Or, specify one of the following:

```
# setenv HTTPPROXY mycache.domain
# setenv HTTPPROXYPORT 8080
```

- Specify the web proxy on the `patchadd` command line.

  For example:

```
# patchadd -x mycache.domain:8080 \
-M http://www.sun.com/solaris/patches/latest 101223-02 102323-02
```

▼ How to Download and Apply a Solaris Patch

Use this procedure to *download* either a signed or an *unsigned Solaris patch* and then apply it to your system.

If you want to apply signed patches, you must first set up the package keystore.

Steps

1. **Gain access to the system in one of these ways:**
   - Log in to the system where you want to apply the patch.
   - Download the patch and use the `ftp` command to copy the patch to the target system.
2. **Start a web browser and go to the Patch Portal at `http://sunsolve.Sun.COM`.**
3. **Determine whether to download a specific patch or a patch cluster, then do one of the following:**
   - Type the patch number (*patch-id*) in the Find Patch search field, then click Find Patch.

     Entering *patch-id* downloads the latest patch revision.

     If this patch is freely available, the patch README appears. If this patch is not freely available, an `ACCESS DENIED` message appears.

     Note that patch numbers for SPARC based and x86 based systems are different. The *patch IDs* are listed in the patch README. Ensure that you apply the patch that matches your system architecture.
   - Select the Recommended Patch Cluster that matches the Solaris release that is running on the system that you want to patch.
4. **Download the patch.**

- To download a copy of the signed patch, click the Download Signed Patch (*n* bytes) HTTPS button or the FTP button.
- To download an unsigned patch, click the Download Patch (*n* bytes) HTTP button or the FTP button.

When the patch or patches are successfully downloaded, close the web browser.

5. **Change to the directory that contains the downloaded patch.**
6. **Become superuser or assume an equivalent role.**
7. **(***Unsigned patch***) If you downloaded an unsigned patch, unzip the patch.**

   # **unzip** *patch-id*

8. **Apply the signed or unsigned patch.**
   - If you downloaded a signed patch, apply it.

     For example:

     ```
     # patchadd /tmp/111879-01.jar
     ```

   - If you downloaded an unsigned patch, apply it.

     For example:

     ```
     # patchadd /tmp/111879-01
     ```

9. **Verify that the patch has been successfully applied.**

   For example:

   ```
   # patchadd -p | grep 111879
   Patch: 111879-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWwsr
   ```

▼ How to Display Information About Solaris Patches

Before applying patches, you might want to know more about patches that have been previously applied. The following commands provide useful information about patches that are already applied to a system.

- `patchadd -p` or `showrev -p`

  Shows all patches that have been applied to the system.

- `pkgparam` *pkgid* `PATCHLIST`

  Shows all patches that have been applied to the package identified by *pkgid*, for example, `SUNWadmap`.

- `patchadd -S` *Solaris-OS* `-p`

  Shows all the `/usr` patches that have been applied to an OS server.

Step ● **Use one of the following `patchadd` command lines to display information about patches that have been applied to your system.**

- To obtain information about all patches that have been applied to your system, type:

  ```
  $ patchadd -p
  ```

- To verify whether a particular patch has been applied to your system, type, for example:

  ```
  $ patchadd -p | grep 111879
  ```

▼ How to Remove a Solaris Patch by Using the `patchrm` Command

Steps 1. **Become superuser.**

2. **Remove the patch.**

   ```
   # patchrm 111879-01
   Checking installed patches...

   Backing out patch 111879-01...

   Patch 111879-01 has been backed out.
   ```

3. **Verify that the patch was removed.**

   ```
   # patchadd -p | grep 111879
   #
   ```

APPENDIX **A**

SMF Services

The following table lists some of the services that have been converted to use SMF. Each service includes the daemon or service name, the FMRIs for that service, the run script that used to start the service, and whether the service is started by `inetd`.

TABLE A–1 SMF Services

Service Name	FMRI	Run Script	`inetd` Service
`automount`	`svc:/system/filesystem/autofs:default`	`autofs`	None
`consadmd`	`svc:/system/consadm:default`	`rootusr`	No
`coreadm`	`svc:/system/coreadm:default`	`coreadm`	No
`cron`	`svc:/system/cron:default`	`cron`	No
`cryptoadm`	`svc:/system/cryptosvc:default`	N/A	No
`cvcd`	`svc:/system/cvc:default`	`cvcd`	No
`dcs`	`svc:/platform/<arch>/dcs:default`	None	Yes
`dtspcd`	`svc:/network/dtspc/tcp:default`	None	Yes
`dumpadm`	`svc:/system/dumpadm:default`	`savecore`	None
`efdaemon`	`svc:/platform/<arch>/efdaemon:default`	`efcode`	No
`fmd`	`svc:/system/fmd:default`	N/A	No
`gssd`	`svc:/network/rpc/gss:default`	None	Yes
`imapd`	`svc:/network/imap/tcp:default`	None	Yes
	`svc:/network/imapnew/tcp:default`		
`in.chargend`	`svc:/network/chargen:dgram`	None	Yes
	`svc:/network/chargen:stream`		

TABLE A–1 SMF Services *(Continued)*

Service Name	FMRI	Run Script	inetd Service
in.comsat	svc:/network/comsat:default	None	Yes
in.daytimed	svc:/network/daytime:dgram	None	Yes
	svc:/network/daytime:stream		
in.dhcpd	svc:/network/dhcp-server:default	dhcp	No
in.discardd	svc:/network/discard:dgram	None	Yes
	svc:/network/discard:stream		
in.echod	svc:/network/echo:dgram	None	Yes
	svc:/network/echo:stream		
in.fingerd	svc:/network/finger:default	None	Yes
in.ftpd	svc:/network/ftp:default	None	Yes
in.named	svc:/network/dns/server:default	inetsvc	No
in.rarpd	svc:/network/rarp:default	boot.server	No
in.rdisc	svc:/network/initial:default	inetinit	No
in.rexecd	svc:/network/rexec:default	None	Yes
in.rlogind	svc:/network/login:rlogin	None	Yes
	svc:/network/login:eklogin		
	svc:/network/login:klogin		
in.routed	svc:/network/initial:default	inetinit	No
in.rshd	svc:/network/shell:default	None	Yes
	svc:/network/kshell		
in.talkd	svc:/network/talk:default	None	Yes
in.telnetd	svc:/network/telnet:default	None	Yes
in.tftpd	svc:/network/tftp/udp6:default	None	Yes
in.timed	svc:/network/time:dgram	None	Yes
	svc:/network/time:stream		
in.tnamed	svc:/network/tname:default	None	Yes
in.uucpd	svc:/network/uucp:default	None	Yes
inetd-upgrade	svc:/network/inetd-upgrade:default	N/A	No
inetd	svc:/network/inetd:default	inetsvc	No

TABLE A–1 SMF Services *(Continued)*

Service Name	FMRI	Run Script	`inetd` Service
`ipop3d`	`svc:/network/pop3/tcp:default`	None	Yes
`kadmind`	`svc:/network/security/kadmin:default`	`kdc.master`	No
`kbd`	`svc:/system/keymap:default`	`keymap`	No
`keyserv`	`svc:/network/rpc/keyserv:default`	`rpc`	No
`kpropd`	`svc:/network/security/krb5_prop:default`	None	Yes
`krb5kdc`	`svc:/network/security/krb5kdc:default`	`kdc`	No
`ktkt_warnd`	`svc:/network/security/ktkt_warn:default`	None	Yes
`ldap_cachemgr`	`svc:/network/ldap/client:default`	`ldap.client`	No
`loadkeys`	`svc:/system/keymap:default`	`keymap`	No
`lockd`	`svc:/network/nfs/client:default`	`nfs.server`	No
	`svc:/network/nfs/server:default`		
`lpsched` and `lpshut`	`svc:/application/print/server:default`	`lp`	No
`mdmonitord`	`svc:/system/mdmonitor:default`	`svm.sync`	No
`metainit`	`svc:/system/metainit:default`	`svm.init`	No
`metadevadm`	`svc:/platform/<arch>/mpxio-upgrade:default`	N/A	No
`mount`	`svc:/system/filesystem/local:default`	`nfs.client`, `rootusr`, `standardmounts`	None
	`svc:/system/filesystem/minimal:default`		
	`svc:/system/filesystem/root:default`		
	`svc:/system/filesystem/usr:default`		
`mountd`	`svc:/network/nfs/server:default`	`nfs.server`	No
`nfsd`	`svc:/network/nfs/server:default`	`nfs.server`	No
`nfsmapid`	`svc:/network/nfs/client:default`	`nfs.server`	No
	`svc:/network/nfs/server:default`		
`nis_cachemgr`	`svc:/network/rpc/nisplus:default`	`rpc`	No
`nscd`	`svc:/system/name-service-cache:default`	`nscd`	No
`ntpdate`	`svc:/network/ntp:default`	`xntpd`	No
`ocfserv`	`svc:/network/rpc/ocfserv:default`	`ocfserv`	No
`picld`	`svc:/system/picl:default`	`picld`	No

TABLE A–1 SMF Services *(Continued)*

Service Name	FMRI	Run Script	inetd Service
pmconfig	svc:/system/power:default	power	No
printd	svc:/application/print/cleanup:default	spc	No
quotaon	svc:/system/filesystem/local:default	ufs_quota	None
rcapd	svc:/system/rcap:default	rcapd	No
rpcbind	svc:/network/rpc/bind:default	rpc	No
rpc.bootparamd	svc:/network/rpc/bootparams:default	boot.server	No
rpc.mdcomm	svc:/network/rpc/mdcomm:default	None	Yes
rpc.metad	svc:/network/rpc/meta:default	None	Yes(?)
rpc.metamedd	svc:/network/rpc/metamed:default	None	Yes
rpc.metamhd	svc:/network/rpc/metamh:default	None	Yes
rpc.nisd	svc:/network/rpc/nisplus:default	rpc	No
rpc.nispasswdd	svc:/network/rpc/nisplus:default	rpc	No
rpc.rexd	svc:/network/rpc/rex:default	None	Yes
rpc.rstatd	svc:/network/rpc/rstat:default	None	Yes
rpc.rusersd	svc:/network/rpc/rusers:default	None	Yes
rpc.smserverd	svc:/network/rpc/smserver:default	None	Yes
rpc.sprayd	svc:/network/rpc/spray:default	None	Yes
rpc.ttdbserverd	svc:/network/rpc-100083_1/rpc_tcp:default	None	Yes
rpc.walld	svc:/network/rpc/wall:default	None	Yes
rpc.yppasswdd and rpc.ypupdated	svc:/network/nis/server:default	rpc	No
rquotad	svc:/network/nfs/rquota:default	None	Yes
sadc	svc:/system/sar:default	perf	No
savecore	svc:/system/dumpadm:default	savecore	None
sendmail	svc:/network/smtp:sendmail	sendmail	No
sf880drd	svc:/platform/<arch>/sf880drd:default	sf880dr	No
slpd	svc:/network/slp:default	slpd	No
sshd	svc:/network/ssh:default	sshd	No

TABLE A–1 SMF Services *(Continued)*

Service Name	FMRI	Run Script	`inetd` Service
`statd`	`svc:/network/nfs/client:default`	`nfs.server`	No
	`svc:/network/nfs/server:default`		
`svc.startd`	`svc:/system/svc/restarter:default`	N/A	No
`syseventd`	`svc:/system/sysevent:default`	`devfsadm`	No
`sysidpm`, `sysidns`, `sysidroot`, `sysidsys`	`svc:/system/sysidtool:system`	`sysid.sys`	No
`sysidnet`	`svc:/system/sysidtool:net`	`sysid.net`	No
`syslogd`	`svc:/system/system-log:default`	`syslog`	No
`ttymon`	`svc:/system/console-login:default`	`inittab`	No
`utmpd`	`svc:/system/utmp:default`	`utmpd`	No
`xntpd`	`svc:/network/ntp:default`	`xntpd`	No
`ypbind`	`svc:/network/nis/client:default`	`rpc`	No
`ypserv`	`svc:/network/nis/server:default`	`rpc`	No
`ypxfrd`	`svc:/network/nis/server:default`	`rpc`	No
`zoneadm`	`svc:/system/zones:default`	N/A	No
None	`svc:/network/loopback:default`	`network`	No
None	`svc:/network/physical:default`	`network`	No

Index

C

D

E

O

P

R

S

W

X