# System Administration Guide: Basic Administration

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

Part No: 817–6958–10
September 2004


040607@9061

# 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 5.9 operating system
- Set up all the networking software that you plan to use

The SunOS 5.9 operating system is part of the Solaris product family, which also includes many features, including the Solaris Common Desktop Environment (CDE). The SunOS 5.9 operating system is compliant with AT&T's System V, Release 4 operating system.

For the Solaris 9 release, new features interesting to system administrators are covered in sections called *What's New in ... ?* in the appropriate chapters.

---

**Note –** The Solaris operating system runs on two types of hardware, or platforms, SPARC and x86. The Solaris operating system runs on both 64–bit and 32–bit address spaces. The information in this document pertains to both platforms and address spaces unless called out in a special chapter, section, note, bullet, figure, table, example, or code example.

---

**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.

---

# Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the Solaris 9 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, 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 networks, IPv4 and IPv6, DHCP, IP Security, Mobile IP, and IP Network Multipathing |
| *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)* | DNS, NIS, and LDAP naming and directory services |
| *System Administration Guide: Naming and Directory Services (FNS and NIS+)* | FNS and NIS+ naming and directory services |
| *System Administration Guide: Resource Management and Network Services* | Resource management, remote file systems, mail, SLP, and PPP |
| *System Administration Guide: Security Services* | Auditing, PAM, RBAC, and SEAM |

To view license terms, attribution, and copyright statements for open source software included in this Solaris release, the default path is `/usr/share/src/`*freeware-name* or `/usr/sfw/share/src/`*freeware-name*. If the Solaris operating system has been installed anywhere other than the default location, modify the given path to access the file at the installed location.

# Accessing Sun Documentation Online

The docs.sun.com[SM] 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.<br>Use `ls -a` to list all files.<br>`machine_name% you have mail.` |
| **`AaBbCc123`** | What you type, contrasted with on-screen computer output | `machine_name%` **`su`**<br>`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*.<br>These are called *class* options.<br>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 (Roadmap)

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 9 release:

- Diskless client support
- Solaris DHCP
- Resource Management
- Solaris Management Console (referred to as the console) tools suite
- Solaris Volume Manager (previously Solstice™ DiskSuite)
- Solaris Patch Manager
- Product Registry

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

**TABLE 1–1** New or Changed Solaris Management Tools in the Solaris 9 Release

| Solaris Administration Tool | Description | For More Information |
|---|---|---|
| Diskless Client Support | Provides a command-line interface for managing diskless client systems. | Chapter 6 |
| Resource Management | Enables you to control how applications use available system resources. | *System Administration Guide: Resource Management and Network Services* |
| Solaris DHCP | Provides improved performance, capacity, and flexibility in managing DHCP in your network. | "About Solaris DHCP (Overview)" in *System Administration Guide: IP Services* |
| Solaris Management Console [1] | Serves as a launching point for a variety of GUI-based system management tools. | This guide and the console online help |
| Solaris Volume Manager (previously Solstice™ DiskSuite) | Provides robust storage management and is launched from the Solaris Management Console. The command-line interface is also available. | *Solaris Volume Manager Administration Guide* |
| Solaris Patch Manager | You can use this tool to add signed and unsigned patches to your system. | Chapter 15 |
| Solaris Product Registry | The `prodreg` command includes `browse`, `info`, `unregister`, and `uninstall` subcommands that are similar to the Solaris Product Registry graphical user interface. | "Managing Software With the Solaris Product Registry Command-Line Interface (Task Map)" on page 224 |

[1] Do not confuse this tool with Sun Management Center (SunMC). For information about the Sun Management Center product, see `http://www.sun.com/solaris/sunmanagementcenter/docs`.

# Matrix of Solaris Management Tools Support

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 2.6 and Earlier Releases | Solaris 7 | Solaris 8 | Solaris 9 |
|---|---|---|---|---|
| admintool | Supported | Supported | Supported | Supported |
| Solstice AdminSuite 2.3 | Supported | Supported | Not Supported | Not Supported |
| Solstice AdminSuite 3.0 | Supported (Solaris 2.6 release only) | Supported | Supported | Not Supported |
| Solaris Management Tools 1.0 | Supported | Supported | Supported | Not Supported |
| Solaris Management Tools 2.0 | Not Supported | Not Supported | Supported (Solaris 8 01/01, 4/01, 7/01, 10/01, 2/02 releases only) | Not Supported |
| Solaris Management Tools 2.1 | Not Supported | Not Supported | Not 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 9 Management Tools

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

**TABLE 1–3** 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 CLI is available |
| Disks Tool | No | Yes |
| Enhanced Disk Tool (Solaris Volume Manager) | No | Yes |
| Job Scheduler | No | Yes |
| Log Viewer | No | Yes |
| Mail Alias Support | No | Yes |
| Mounts and Shares Tool | No | Yes |
| Name Service Support | No | For users, groups, and network information only |
| Patch Tool | No | Yes |
| Performance Tool | No | Yes |
| Printer Support | Yes | Solaris Print Manager is available separately |
| 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

This table lists the tools that are available in the Solaris 8 release and various Solaris 8 update releases.

**TABLE 1–4** 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 CLI is available separately) |
| Disks Tool | No | No | No | Yes |
| Job Scheduler | No | No | No | Yes |
| Log Viewer | 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 | Solaris Print Manager is available | Yes | No, but Solaris Print Manager is available |
| Software Package Tool | Yes | No | Yes | No |
| RBAC Support | No | Yes (rights support only) | No | Yes |
| RBAC Tool | No | RBAC CLI is available separately | No | Yes |
| Serial Port Tool | Yes | Yes | Yes | Yes |

**TABLE 1–4** 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) |
|---|---|---|---|---|
| User/Group Tool | Yes | Yes | Yes | Yes |

# Feature Descriptions for Previous Solaris Management Tools

This table describes the tools that are available in releases prior to the Solaris 8 release.

**TABLE 1–5** Feature Descriptions for Previous Solaris Management Tools

| Feature or Tool | Supported in admintool? | Supported in Solstice AdminSuite 2.3? | Supported in Solstice AdminSuite 3.0? (Solaris 2.6 only) |
|---|---|---|---|
| Diskless Client Support | No | Yes | No |
| Disks Tool | No | Yes | No |
| Log Viewer | No | No | Yes |
| Mail Alias Support | No | Yes | Yes |
| Mounts and Shares Tool | No | Yes | Yes |
| Name Service Support | No | Yes | Yes |
| Printer Support | Yes | Yes | Solaris Print Manager is available |
| RBAC Support | No | No | Yes (rights support only) |
| RBAC Tool | No | No | RBAC CLI is available separately |
| Serial Port Tool | Yes | Yes | Yes |
| User/Group Tool | 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 6.

## Solaris 9 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 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) |

**TABLE 1–6** Descriptions for Solaris Management Commands *(Continued)*

| Command | Description | Man Page |
|---|---|---|
| `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 listed Table 1–7 and Table 1–8 are available in the Solaris 8 release.

## Descriptions for Previous Solaris Management Commands

This table describes the commands that provide equivalent functionality to the Solstice AdminSuite™ 2.3 GUI tool. You must be superuser or be a member of the sysadmin group to use these commands.

---

**Note –** The Solstice AdminSuite 2.3 command man pages are not available online. You must have access to the Solstice AdminSuite 2.3 software to view these man pages.

---

All of the commands listed in Table 1–8 are also available in previous Solaris releases.

**TABLE 1–9** Descriptions for Solstice AdminSuite™ 2.3 Commands

| Command | Description | For More Information |
|---|---|---|
| `admhostadd`, `admhostmod`, `admhostdel`, `admhostls` | Adds, modifies, removes, and lists support for client and server systems set up with the AdminSuite software | *Solstice AdminSuite 2.3 Administration Guide* |
| `admuseradd`, `admusermod`, `admuserdel`, `admuserls`, `admgroupadd`, `admgroupmod`, `admgroupdel`, `admgroupls` | Adds, modifies, removes, and lists users and groups | *Solstice AdminSuite 2.3 Administration Guide* |

# For More Information About Solaris Management Tools

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

**TABLE 1–10** For More Information About Solaris Management Tools

| Tool | Availability | For More Information |
|---|---|---|
| Solaris Management Console 2.1 suite of tools | Solaris 9 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 | The Solaris Management Console online help |

**TABLE 1–10** For More Information About Solaris Management Tools *(Continued)*

| Tool | Availability | For More Information |
|---|---|---|
| Solaris Management Console 1.0 suite of tools | Solaris 2.6, Solaris 7, and Solaris 8 releases | *Solaris Easy Access Server 3.0 Installation Guide* |
| admintool | Solaris 9, Solaris 8, and previous Solaris releases | `admintool`(1M) |
| AdminSuite 2.3 | Solaris 2.4, Solaris 2.5, Solaris 2.5.1, Solaris 2.6, and Solaris 7 releases | *Solstice AdminSuite 2.3 Administration Guide* |
| AdminSuite 3.0 | Solaris 8, Solaris 8 6/00, and Solaris 8 10/00 releases | *Solaris Easy Access Server 3.0 Installation Guide* |
| | Solaris 9 releases – Consider using the Web Start Flash installation feature | *Solaris 9 9/04 Installation Guide* |
| Diskless Client CLI | Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02, and Solaris 9 releases | Chapter 6 |

CHAPTER 2

# Working With the Solaris™ Management Console (Tasks)

This chapter provides an overview of the Solaris management tools 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 with the Solaris Management Console, see:

- "Using the Solaris Management Tools With RBAC (Task Map)" on page 36
- "Using the Solaris Management Tools in a Name Service Environment (Task Map)" on page 42

For information on troubleshooting Solaris Management Console problems, see "Troubleshooting the Solaris Management Console" on page 49.

## Solaris Management Console (Overview)

The following sections provide information about the Solaris Management 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 users, projects, and cron jobs; for mounting and sharing file systems; and for managing disks and serial ports. For a brief description of each Solaris management tool, see Table 2–1.

You can always 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 *console*, this 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, and 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 would 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 and provides cross-references to background information for each tool.

**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 timezone. | "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. | "Troubleshooting Software Problems (Overview)" in *System Administration Guide: Advanced Administration* |
| | Processes | Monitors and manages system processes. | "Processes and System Performance" in *System Administration Guide: Advanced Administration* |

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

| Category | Tool | Description | For More Information |
|---|---|---|---|
| | Performance | Monitors system performance. | "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 53 and "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services* |
| | Projects | Creates and manages entries in the `/etc/project` database. | "Projects and Tasks" in *System Administration Guide: Resource Management and Network Services* |
| | Computers and Networks | Creates and monitors computer and network information. | Solaris Management Console online help |
| | Patches | Manages patches. | Chapter 15 |
| **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. | "Managing File Systems (Overview)" in *System Administration Guide: Devices and File Systems* |
| | Disks | Creates and manages disk partitions. | "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* |
| **Devices and Hardware** | Serial Ports | Sets up terminals and modems. | "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, which you can reach 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 interface, which includes dialog boxes, wizards, and context help. Experienced administrators will find that the console provides a convenient, secure alternative to using `vi` to manage hundreds of configuration parameters spread across tens 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* that can be assigned to users, typically administrators, who are permitted to make specific system changes.

  The key benefit of RBAC is that roles can be limited 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, with the names and brief descriptions of each command. There is also a man page for each command.

  For those Solaris management tools that have no special commands, such as the Mounts and Shares tools, 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 36.

# Organization of the Solaris Management Console

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

The main part of the console consists of three panes:

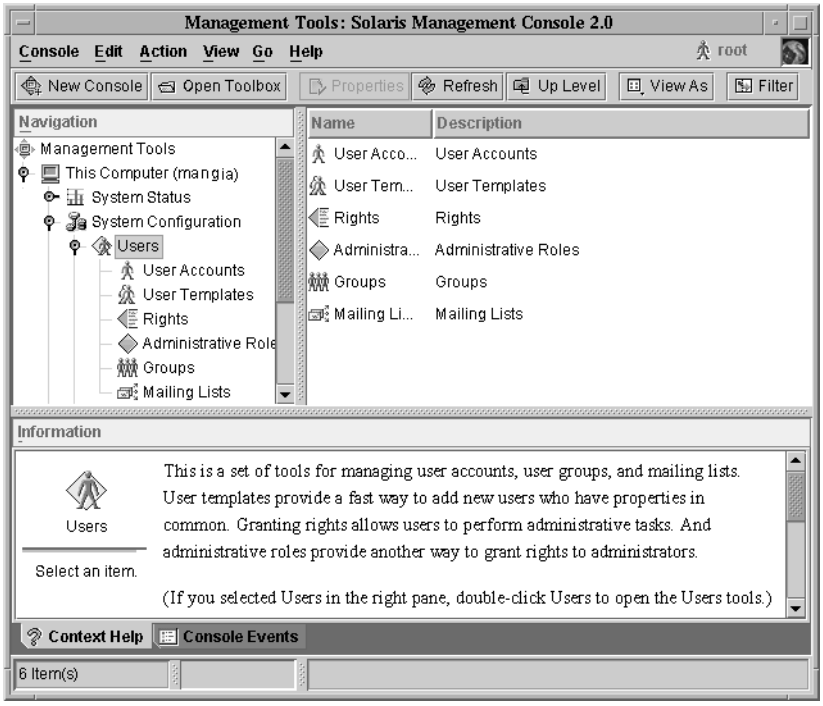



**FIGURE 2–1** Solaris Management Console – Users Tool

- 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, shows either the contents of the selected folder, subordinate tools, or 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 30, 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, as follows:

| 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 enter root as the user and the root password. | "How to Become Superuser (root) or Assume a Role" on page 34 |
| 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.<br><br>Note that you can apply this technique whether or not you are using other roles. | "Making a Role" 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 37 |
| 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* |
| Custom Roles | You can add your own roles, depending on your organization's security needs. | "Planning for RBAC" 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 into 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 into the console as yourself and then supply the root user name and password.
- Log into 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 becoming superuser.

For step-by-step instructions on creating the Primary Administrator role, see "How to Create the First Role (Primary Administrator)" on page 39. For an overview on configuring RBAC to use roles, see "Configuring RBAC (Task Map)" 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.

1. **Become Superuser – Select one of the following to become superuser.**

   a. **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 47.

**b. 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.

**c. 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.

**d. 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 "Securing Machines (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 to assume a role.**

**a. 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 the role has access to.

**b. 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 40.

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 Role-Based Access Control (RBAC) security features rather than use 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 setup RBAC with the console.

For detailed information on RBAC and using 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, and 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 40 |
| 2. Add a user account for yourself | Add a user account for yourself if you do not have one already. | Solaris Management Console online help |
| 3. Create the Primary Administrator role | Create the Primary Administrator role and add yourself to this role. | "How to Create the First Role (Primary Administrator)" on page 39 |
| 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 39 |
| 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. | "Making a Role" in *System Administration Guide: Security Services* |

| Task | Description | For Instructions |
|---|---|---|
| 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. | "How to Create a Role by Using the Administrative Roles Tool" 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), and then log in as superuser. This method gives you complete access to all the console tools.

Here are the general steps, depending on whether or not 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 one.
  - Create the role called Primary Administrator.
  - Assign the Primary Administrator right to the role 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 39.

    For an overview on configuring RBAC to use roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

# Creating the Primary Administrator Role

An administrative role is a special user account. Users who assume a role are permitted to perform a pre-defined 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 or for performing specific functions within an application, and other rights, whose use 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** Item Descriptions for Adding a Role by Using the Console

| Item | 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 | 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 into 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 and 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 they can assume the role to perform specific tasks. |

For detailed information about Role-Based Access Control, and 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.

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 40.

   See the console online help if you need to create a user account for yourself.

2. **Click This Computer icon in the Navigation pane.**

3. **Click 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 items.

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.

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 40.

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.

## ▼ 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 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, equivalent to that of superuser.

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:

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

     It might take a minute or two for the console to come up the first time.
   - From the Tools menu of the CDE front panel.
   - By double-clicking a 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 start-up 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 49.

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 45.

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 popup Log-In window is displayed.

7. **Decide if you want to the tool as superuser or as a role.**

   - If you are logging in as superuser and will be working as superuser, select step 8.
   - If you are logging in as yourself and will be assuming the Primary Administrator role, select steps 9 and 10.

8. **If you are logging in as superuser, enter the root password.**

9. **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.

10. **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 39.

    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 a 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 44 |
| 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 45 |
| 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 46 |
| 4. Select the toolbox 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 47 |

## 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 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 49.

The RBAC security files in the Solaris 9 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 release.

This table briefly describes the pre-defined security files that are installed on a Solaris 9 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 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, or
- 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 release. | *Solaris 9 9/04 Installation Guide* |
| 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 44 |
| Make sure your `/etc/nsswitch.conf` file is configured so that you can access your name service data. | "The `/etc/nsswitch.conf` File" on page 44 |

## 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 and Computers and Networks tools 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.

## The `/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 using the Solaris managements 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, and 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, 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`.

---

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, “NIS tools” for the NIS environment.
   b. **Provide a description in the Description window. Click Next.**
      For example, “tools for NIS environment.”
   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.

## Where to Go From Here

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 46.

# ▼ 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, and then add tools to the new toolbox.

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

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 45.

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

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

1. **Verify that the following prerequisites are met.**

   a. **Be sure the system you are logged into is configured to work in a name service environment.**

   b. **Verify that the `/etc/nsswitch.conf` file is configured to match your name service environment.**

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

   For more information, see "How to Start the Console as Superuser or as a Role" on page 40.

3. **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 45.

# 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 "Securing Legacy Applications" 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, X applications, command-line interface, and HTML, to a console toolbox. Each tool you add to a toolbox can then be launched from the Solaris Management Console.

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 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 console.

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 40.

# Troubleshooting the Solaris Management Console

Before using this troubleshooting procedure, make sure 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 is not running. Or, the server might be in a problem state that can be resolved by stopping and restarting it.

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 like 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 like the following:

```
SMC server is ready.
```

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

   a. **Stop the console server.**

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

   You should see a message like the following:

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

   b. **Start the console server.**

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

CHAPTER 3

# 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 4.

## What's New in Managing Users and Groups?

This section describes new features for managing users and groups in the Solaris 9 release.

## Solaris Management Console Tools Suite

The Solaris Management tools suite, available from the Solaris Management Console, enable you to manage all user and group features. For information on using the Solaris Management Console, see Chapter 2. For information on performing specific user and group management tasks, see "What You Can Do With Solaris User Management Tools" on page 62.

## Solaris Directory Services

You can manage user and group information in a LDAP (Lightweight Directory Access Protocol) directory service with the iPlanet™ Directory Server, as well as other LDAP directory servers. Managing user and group information is also available in the NIS, NIS+, or files environment.

For information on setting up LDAP, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

For information on managing users and groups with iPlanet Directory Server, see *iPlanet Directory Server 5.1 Administrator's Guide* at this `http://docs.sun.com/db/doc/816-2670?` location.

## Managing Users and Resources With Projects

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 "Projects and Tasks" in *System Administration Guide: Resource Management and Network Services*.

# 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. User account information has the following components:

| Component | Description |
|---|---|
| User name | A name that a user uses to log in to a system. This name is also known as a login name. |
| Password | A secret combination of characters that a user must enter with a user name to gain access to a system. |
| User's home directory | A directory that is usually the user's current directory at login. The user's home directory typically contains most of the user's files. |
| User initialization files | Shell scripts that control how the user's working environment is set up when a user logs in to a system. |

Also, 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*.

A user or group can be a member of one or more projects. A project is an identifier that is used to chargeback system resources. For information on using projects, see "Projects and Tasks" in *System Administration Guide: Resource Management and Network Services*.

# Guidelines for Managing User Accounts

The following sections describe some guidelines and planning information for creating user accounts.

## 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 using 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 (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 you create.

Keep the following guidelines in mind when creating user or role names:

- Be unique within your organization, which might span multiple domains
- Contain from two to eight letters and numerals. The first character should be a letter and at least one character should be a lowercase letter.

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.

Consider establishing a standard way of assigning user names so 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:

- The first initial, middle initial, first five characters of the user's last name,
- and 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 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.

# User ID Numbers

Associated with each user name is a user identification (UID) number. The user UID identifies the user name to any system on which the user attempts to log in. And, the user UID 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 user ID. In that way, the user can easily move files between systems without ownership problems.

UID numbers must be a whole number 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 reserved for user accounts and system accounts.

**TABLE 3–1** Reserved UID Numbers

| User ID Numbers | Use 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 |

Although UID numbers 0 through 99 are reserved, you can add a user with one of these numbers. However, do not use 0 through 99 for 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, like `who`, `tty`, and `ttytype`, low UIDs so they fall at the beginning of the `passwd` file.

As with user (login) names, you should adopt a scheme to assign unique UIDs. Some companies assign unique employee numbers, and 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 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, but that attribute might not be appropriate for the new user.

## Using Large User IDs and Group IDs

UIDs and 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 3–2** Interoperability Issues for UIDs or GIDs Over 60000

| Category | Product or Command | Issues or Cautions |
|---|---|---|
| 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, but their UIDs and GIDs will be set to 60001 (nobody). |
| | 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 3–3** Large UID or GID Limitation Summary

| UID or GID | Limitations |
|---|---|
| 60003 or greater | ■ Users in this category logging into systems running Solaris 2.5 and compatible releases and the NIS or files name service get a UID and GID of `nobody`. |

**TABLE 3–3** Large UID or GID Limitation Summary *(Continued)*

| UID or GID | Limitations |
|---|---|
| 65535 or greater | ■ Systems running Solaris 2.5 and compatible releases with the NFS version 2 software see UIDs in this category truncated to 16 bits, creating possible security problems.<br>■ Users in this category using 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.<br>■ SPARC based systems: Users in this category running SunOS 4.0 and compatible applications see `EOVERFLOW` returns from some system calls, and their UIDs and GIDs are mapped to `nobody`.<br>■ x86 based systems: Users in this category running SVR3-compatible applications will probably see `EOVERFLOW` return codes from system calls.<br>■ x86 based systems: If users in this category 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 they include a UID or GID larger than 99999. |
| 262144 or greater | ■ Users in this category using 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 in this category using the `ar` command have their UIDs and GIDs set to `nobody` in the archive. |
| 2097152 or greater | ■ Users in this category using the `tar` command, the `cpio -H ustar` command, or the `pax -x tar` command have their UIDs and GIDs set to `nobody`. |

# 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.

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

To make your computer systems more secure, ask users to 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, forwards, backwards, or jumbled
- Names of family members or pets
- Car license numbers
- Telephone numbers
- Social Security numbers
- Employee numbers
- Names related to a hobby or interest
- Seasonal themes, such as Santa in December
- Any word in the dictionary

# 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.

## 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, large or small, and the number of files 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: Resource Management and 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 the links will be valid no matter where the home directory is mounted.

## 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 that 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. For detailed information about customizing user initialization files for users, see "Customizing a User's Work Environment" on page 72. After you create the customized user initialization files, you can add them to a user's home directory when you create a new user account.

A recommended one-time task is to set up *skeleton* directories on a server. You can use the same server where the user's home directories are stored. The skeleton directories enable you to store customized user initialization files for different types of users.

**Note –** Do not use system initialization files, such as `/etc/profile` or `/etc/.login`, to manage a user's work environment, because they 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 when a user moved from system to system.

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

# Guidelines for Managing Groups

A *group* is a collection of users who can share files and other system resources. For example, a set of users that are 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 identifies the group internally to the system. The two types of groups that a user can belong to are:

- Primary group – Specifies a group that the operating system assigns to files 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.

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 memberships. For example, a user has to be a member of the sysadmin group (group 14) to use the Admintool software, but 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, 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 can be managed through a name service. To simplify group administration, you should use a name service like NIS or a directory service like LDAP, which enables you to centrally manage group memberships.

# Tools for Managing User Accounts and Groups

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

**TABLE 3–4** Tools for Managing Users and Groups

| Solaris Management Tool | Is Used To | Task Information |
|---|---|---|
| Users | Manage users. | Solaris Management Console Online Help |
| User Templates | Create a set of attributes for a specific kind of user like students, engineers, or instructors. | Solaris Management Console Online Help |
| Rights | Manage RBAC rights. | Solaris Management Console Online Help |
| Administrative Roles | Manage RBAC administrative roles. | Solaris Management Console Online Help |
| Groups | Manage group information. | Solaris Management Console Online Help |
| Projects | Manage project information. | Solaris Management Console Online Help |
| Mailing Lists | Manage mailing lists. | Solaris Management Console Online Help |

For information on the Solaris management commands that can be used to manage user accounts and groups if you are not using the Solaris Management Console, see Table 1–6. These commands provide the same functionality as the Solaris management tools, including authentication and name service support.

# What You Can Do With Solaris User Management Tools

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

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

**TABLE 3–5** User Account Management Tasks

| Task | Description | Background Information |
|---|---|---|
| Add a user | You can add a user to the local system or name service. | "What Are User Accounts and Groups?" on page 53 and "Guidelines for Managing User Accounts" on page 54 |
| Create a user Template | You can create a template of pre-defined user attributes for creating users of the same group, such a users, contractors, or engineers. | Same as above |
| Add a user with a user template | You can add a user with a template so that user attributes are pre-defined. | Same as above |
| Clone a user template | Clone a user template if you would like to use a similar set of pre-defined user attributes. Then, change only some of the attributes as needed. | Same as above |
| Set up user properties | You can set up user properties in advance of adding users such as 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. | Same as above |

**TABLE 3–5** User Account Management Tasks *(Continued)*

| Task | Description | Background Information |
|---|---|---|
| Add multiple users | You can add 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. | Same as above |
| View or change user properties | You can view or change user properties like login shell, password, or password options. | Same as above |
| Assign rights to users | You can assign rights to users that will allow them to perform specific administration tasks. | Same as above |
| Remove a user | You can remove the user from the local system or the name service and optionally specify whether the user's home directory or mail is removed. The user is also removed from any groups or roles. | Same as above |

**TABLE 3–6** User Rights Management Tasks

| Task | Description | Background Information |
|---|---|---|
| Grant a right | You can grant a user a right to run a specific command or application that was previously only available to an administrator. | "RBAC Rights Profiles" in *System Administration Guide: Security Services* |
| View or change existing rights Properties | You can view or change existing rights. | Same as above |
| Add an authorization | You can add an authorization, which is a discrete right granted to a role or a user. | "RBAC Authorizations" in *System Administration Guide: Security Services* |
| View or change an authorization | You can view or change existing authorizations. | Same as above |

**TABLE 3–7** User Role Management Tasks

| Task | Description | Background Information |
|---|---|---|
| Add an administrative role | You can add a role that someone would use to perform a specific administrative task. | "RBAC Roles" in *System Administration Guide: Security Services* |

**TABLE 3–7** User Role Management Tasks *(Continued)*

| Task | Description | Background Information |
|---|---|---|
| Assign rights to an administrative role | You can assign specific rights to a role that enable someone to perform a task. | Same as above |
| Change an administrative role | You can add or remove rights from a role. | Same as above |

**TABLE 3–8** Group Management Tasks

| Task | Description | |
|---|---|---|
| Add a group | Add a group to the local system or name service so that the group name is available before you add the user. | "Guidelines for Managing Groups" on page 60 |
| Add a user to a group | Add a user to a group if the user needs access to group-owned files. | Same as above |
| Remove a user from a group | You can remove a user from a group if the user no longer requires group file access. | Same as above |

**TABLE 3–9** Project Management Tasks

| Task | Description | Background Information |
|---|---|---|
| Create or clone a project | You can create a new project or clone an existing project if it has attributes similar to what you need for the new project. | Solaris Management Console online help |
| Modify or view project attributes | You can view or change existing project attributes. | Solaris Management Console online help |
| Delete a project | You can remove a project if it is no longer used. | Solaris Management Console online help |

**TABLE 3–10** Mailing List Management Tasks

| Task | Description | Background Information |
|---|---|---|
| Create a mailing list | You can create a mailing list, which is a list of names for sending email messages. | Solaris Management Console online help |
| Change a mailing list name | You can make changes to the mailing list after it is created. | Solaris Management Console online help |

**TABLE 3–10** Mailing List Management Tasks *(Continued)*

| Task | Description | Background Information |
|---|---|---|
| Remove a mailing list | You can remove a mailing list if it is no longer used. | Solaris Management Console online help |

# Managing 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 setup to be automounted, and the following entry is added to the `passwd` file:

  ```
  /home/username
  ```

- The only way you can use Users Tool to set up a user account that does not automount the home directory is to set up a user account template that disables this feature. Then, you can add users with this template. There is no way to 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.

# Modify 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 login 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 re-add it 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 Properties 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

## Delete 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, you can delete the files in the user's home directory and mail directory.

## Add Customized User Initialization Files

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.

You can customize the user initialization templates in the `/etc/skel` directory and then copy them to users' home directories.

## Administer Passwords

You can use Users Tool for password administration, which 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.

---

## Disable User Accounts

Occasionally, you might need to temporarily or permanently disable a login 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 to set a limit on how long the account is active.

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

# Where User Account and Group Information Is Stored

Depending on your site policy, you can store user account and group information in a name service or a local system's `/etc` files. In the NIS+ name service, information is stored in tables, in the NIS name service, information is stored in maps, and in 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 of the user account information is stored in the `passwd` file. However, password encryption and password aging is stored in the `passwd` file when using NIS or NIS+ and in the `/etc/shadow` file when using `/etc` files. Password aging is not available when using NIS.

Group information is stored in the `group` file.

## 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 3–11** 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. 60001 and 60002 are assigned to `nobody` and `noaccess`. 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 3–18 contains a description of shell features. |

## Default `passwd` File

The default Solaris `passwd` file contains entries for standard daemons, processes usually started at boot time to perform some system-wide task, such as printing, network administration, and 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:
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
```

**TABLE 3–12** 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` | Daemon associated with `uucp` functions. |
| `smmsp` | `25` | Sendmail message submission program daemon. |
| `listen` | `37` | Network listener daemon. |
| `nobody` | `60001` | Assigned to users or software processes that do not need nor should have any special permissions. |
| `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 3–13** Fields in the `shadow` File

| Field Name | Description |
|---|---|
| *username* | Contains the user or login name. |
| *password* | Might contain the following entries: a 13-character encrypted user password; the string `*LK*`, which indicates an inaccessible account; or 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. |
| *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 3–14** 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. |
| *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. |

**TABLE 3–14** Fields in the `group` File *(Continued)*

| Field Name | Description |
|---|---|
| *gid* | Contains the group's GID number. It 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. 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, and electronic mail. Many of these groups having corresponding entries in the `passwd` file.

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

**TABLE 3–15** Default `group` File Entries

| Group Name | Group ID | Description |
|---|---|---|
| `root` | `0` | Superuser group. |
| `other` | `1` | Optional group. |
| `bin` | `2` | Administrative group associated with running system binaries. |
| `sys` | `3` | Administrative group associated with system logging or temporary directories. |

**TABLE 3–15** Default `group` File Entries *(Continued)*

| Group Name | Group ID | Description |
|---|---|---|
| `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 Admintool and Solstice AdminSuite tools. |
| `smmsp` | 25 | Sendmail message submission program daemon. |
| `nobody` | `60001` | Group assigned to users or software processes that do not need nor should have any special permissions. |
| `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 not a member of a known group. |

# 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, its 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 3–16** User Initialization Files for Bourne, C, and Korn Shells

| Shell | User Initialization File | Purpose |
|---|---|---|
| Bourne | `$HOME/.profile` | Defines user's environment at login |
| C | `$HOME/.cshrc` | Defines user's environment for all C shells and is invoked after login shell |
| | `$HOME/.login` | Defines user's environment at login |
| Korn | `$HOME/.profile` | Defines 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 3–17** 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 them to provide the working environment for different types of users. 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 87.

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:

| 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`. |
| 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*` and `/etc/skel/.profile` files are copied into the user's home directory. At this point, you will 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- 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*

## Avoid Local System References

You should not add specific references to the local system in the user's initialization file. You want the instructions in a user initialization file to be valid regardless of the system to which the user logs in. 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`. `$HOME` 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, like `/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 3–18** 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 |
| 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` | 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, like `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 and 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 3–19** Setting Environment Variables in a User Initialization File

| Set a User's Environment Variables for The 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 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 `sh`(1), `ksh`(1), or `csh`(1).

**TABLE 3–20** 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 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. |
| 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 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.<br><br>The default PATH is automatically defined and set as specified in .profile (Bourne or Korn shell) or .cshrc (C shell) as part of the login process.<br><br>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. |

**TABLE 3–20** Shell and Environment Variable Descriptions *(Continued)*

| Variable | Description |
|---|---|
| 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 /etc/profile or /etc/.login.<br><br>When the TERMINFO environment variable is set, the system first checks the TERMINFO path defined by the user. If it 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 /etc/profile or /etc/.login. When the user invokes an editor, the system looks for a file with the same name as the definition of this environment variable. The system searches the directory referenced by TERMINFO to determine the terminal characteristics. |
| TZ | Sets the time zone, which 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, but most users modify it to add other command directories. Many user problems related to setting up the environment and accessing the right 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 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 and to reduce 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- 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, like 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 set whether the terminal session will support multibyte characters.

`LANG` sets all possible conversions and conventions for the given locale. If you have special needs, 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 3–21** Values for `LANG` and `LC` Variables

| Value | Locale |
|---|---|
| `de_DE.ISO8859-1` | `German` |
| `en_US.UTF-8` | `American English` (UTF-8) |

**TABLE 3–21** Values for LANG and LC Variables *(Continued)*

| Value | Locale |
|---|---|
| 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- 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 sets permissions for group
- The third sets permissions for other, also referred to as “world”

Note that if the first digit is zero, it is not displayed. For example, if umask 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, which shows the file and directory permissions that are created for each of the octal values of umask.

**TABLE 3–22** Permissions for umask Values

| umask Octal Value | File Permissions | Directory Permissions |
|---|---|---|
| 0 | rw- | rwx |
| 1 | rw- | rw- |
| 2 | r-- | r-x |
| 3 | r-- | r-- |
| 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. Many of the examples use system names and paths that you need to change for your particular site.

## Example—.profile File

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

```
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—`.cshrc` File

```
1 set path=($PATH $HOME/bin /usr/local/bin /usr/ccs/bin)
2 setenv MAIL /var/mail/$LOGNAME
3 setenv NNTPSERVER server1
4 setenv PRINTER printer1
5 alias h history
6 umask 022
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 will need 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 4

# Managing User Accounts and Groups (Tasks)

This chapter describes how to set up and maintain user accounts and groups by using the Solaris Management Console.

For information on the procedures associated with setting up and maintaining user accounts and groups with the Solaris Management Console, see "Setting Up User Accounts (Task Map)" on page 85 and "Maintaining User Accounts (Task Map)" on page 94.

For background information about managing user accounts and groups, see Chapter 3.

## Setting Up User Accounts (Task Map)

| Task | Description | For Instructions |
|---|---|---|
| (Optional) 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 86 |
| (Optional) Customize user initialization files | You can set up user initialization files (`.cshrc`, `.profile`, `.login`), so you can provide new users with consistent environments. | "How to Customize User Initialization Files" on page 87 |
| (Optional) Add a group | You can add a group with the following tools: | |

| Task | Description | For Instructions |
|---|---|---|
| | Solaris Management Console's Groups tool | "How to Add a Group with the Solaris Management Console's Groups Tool" on page 89 |
| | Solaris command line interface tools | "How to Add Groups and Users With CLI Tools" on page 91 |
| Add a user | You can add a user with the following tools: | |
| | Solaris Management Console's Users Tool | "How to Add a User With the Solaris Management Console's Users Tool" on page 90 |
| | Solaris command line interface tools | "How to Add Groups and Users With CLI Tools" on page 91 |
| (Optional) Set up a user template | You can create a user template so you don't have to manually add all similar user properties. | See Solaris Management Console online help |
| (Optional) Add rights or a role to a user | You can add rights or a role to a user so 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 the directory can be remotely mounted from the user's system. | "How to Share a User's Home Directory" on page 91 |
| 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 93 |

# How to Gather User Information

You can create a form like the one that follows to gather information about users before adding their accounts.

| Item | Description |
|---|---|
| User Name: | |

| | |
|---|---|
| Role Name: | |
| Profiles or Authorizations: | |
| User Name: | |
| UID: | |
| Primary Group: | |
| Secondary Groups: | |
| Comment: | |
| Default Shell: | |
| Password Status and Aging: | |
| Home Directory Server Name: | |
| 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

1. **Become superuser or assume an equivalent role on the system where the users' home directories are created and shared.**

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 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).

---

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 72.

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—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—.cshrc File" on page 82.

```
# 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

Use this procedure to add a group to the system.

1. **Become superuser or assume an equivalent role.**
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 40 or "How to Start the Solaris Management Console in a Name Service Environment" on page 47.
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 for your name service environment.**
5. **Double-click the System Configuration icon.**
6. **Double-click the User Accounts icon.**
7. **Provide the superuser password or the role password.**
8. **Double-click the Groups icon.**

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

### Example—Adding a Group With the Solaris Management Console's Groups Tool

The following example identifies the steps to add the group `mechanoids` (group ID 101) to the system `starbug`. This example assumes that the launcher has been started and Users tool is open.

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.

- Select Add Group from the Action menu.
- Identify the group name, `mechanoids`, at the Group Name prompt under Group Identification.
- Identify the group number, 101, at the Group ID number prompt.
- Click on OK.

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

Use the following procedure to add a user to the system.

1. **Become superuser or assume an equivalent role.**
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 40 or "How to Start the Solaris Management Console in a Name Service Environment" on page 47.
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 for your name service environment.**
5. **Double-click the System Configuration icon.**
6. **Double-click the User Accounts icon.**
7. **Provide the superuser password or the role password.**
8. **Double-click the Users icon.**

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

## Example—Adding a User With the Solaris Management Console's Groups Tool

The following example identifies the steps to add the user `kryten` (user ID 1001) to the system `starbug`. This example assumes that the launcher has been started and Users Tool is open.

Click Next between the steps below.

- Select Add User—>With Wizard from the Action menu.
- Step 1 – Identify the user's name or login name, `kryten`, at the User Name prompt under Group Identification.
- (Optional) Identify the user's full name, `kryten series 3000`, at the Full Name prompt.
- (Optional) Provide a further description of this user at the Description prompt.
- Step 2 – Provide the user ID, 1001, at the User ID Number prompt.
- Step 3 – 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.

- Step 4 – Select the user's primary group, `mechanoids`.
- Step 5 – Create the user's home directory by accepting the defaults at the Server and Path prompts.
- Step 6 – Specify the mail server.
- Step 7 – Review the information you provided and go back to correct the information, if necessary. Otherwise, click on Finish.

## How to Add Groups and Users With CLI Tools

This section provides examples of adding users and groups with CLI 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 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 `groupadd`(1M) and `useradd`(1M).

### 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 `starbug`.

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

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

## ▼ How to Share a User's Home Directory

1. **Become superuser or assume an equivalent role on the system that contains the home directory.**

2. **Verify that the `mountd` daemon is running.**

   ```
   # ps -ef | grep mountd
   root   176     1  0   May 02 ?        0:19 /usr/lib/nfs/mountd
   ```

   The `/usr/lib/nfs/mountd` line shows whether the `mountd` daemon is running.

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

   ```
   # /etc/init.d/nfs.server start
   ```

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

   ```
   # share
   ```

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

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

   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 containing 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 you do not have to wait to reboot the system.

8. **Verify that a user's home directory is shared, as follows:**

   ```
   # share
   ```

## Where to Go From Here

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 93.

## Example—Sharing a User's Home Directory

```
# ps -ef | grep mountd
# /etc/init.d/nfs.server start
# 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   ""
```

## ▼ 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: Resource Management and Network Services*.

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 91.

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/`*user-name* `nfs - yes rw`

   | | |
   |---|---|
   | *system-name* | The name of the system where the home directory is located. |
   | `/export/home/`*user-name* | The name of the user's home directory that will be shared. By convention, `/export/home`*user-name* contains user's home directories. However, this could be a different file system. |
   | - | Required placeholders in the entry. |
   | `/export/home/`*user-name* | 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 and Unmounting File Systems (Tasks)" in *System Administration Guide: Devices and File Systems*.

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

   `# mkdir -p /export/home/`*user-name*

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 user-name
```

### Example—Mounting a User'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= ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/onerror= ...
/proc on /proc read/write/setuid/dev=3dc0000 on Fri Apr  2 13:26:30 2004
/etc/mnttab on mnttab read/write/setuid/dev=3e80000 on Fri Apr  2 13:26:30 2004
/dev/fd on fd read/write/setuid/dev=3ec0000 on Fri Apr  2 13:26:33 2004
/var/run on swap read/write/setuid/xattr/dev=1 on Fri Apr  2 13:26:34 2004
/tmp on swap read/write/setuid/xattr/dev=2 on Fri Apr  2 13:26:38 2004
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr...
/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. | See Solaris Management Console online help |
| Delete a Group | You can delete a group if its no longer needed. | See Solaris Management Console online help |
| Modify a User Account | *Disable a User Account*<br>You can temporarily disable a user account if it will be needed in the future.<br>*Change a User's Password*<br>You might need to change a user's password if the user forgets it. | See Solaris Management Console online help<br>See Solaris Management Console online help |

| Task | Description | Instructions |
|---|---|---|
| | *Change Password Aging*<br>You can force users to change their passwords periodically with User Account tool's Password Options menu. | See Solaris Management Console online help |
| Delete a User Account | You can delete a user account if it is no longer needed. | See Solaris Management Console online help |

# Solaris User Registration

Solaris User Registration is a tool for getting information about new Solaris releases, upgrade offers, and promotions. This graphical user interface (GUI) automatically starts when you first log into your desktop. The GUI lets you register now, later, or never. The registration process also provides Sun with the user's Solaris version, survey type, platform, hardware, and locale.

## Accessing Solaris Solve

Completing the Solaris User Registration process provides access to Solaris Solve$^{SM}$, an exclusive web site that offers valuable Solaris product information and solutions—all in one convenient location. It provides a quick and easy method for getting the most recent information on what's happening around the latest Solaris release. Solaris Solve also provides a preview to additional Sun contract and service opportunities.

Basically, the steps for completing Solaris User Registration and accessing Solaris Solve are:

1. Fill in the electronic Solaris User Registration profile.
2. Submit the profile by email or print the profile to fax or mail.
3. Create your login ID and password to access the Solaris Solve site.

   Even if you do not access the Solaris Solve site immediately, we recommend that you create your Solaris Solve login ID and password during the Solaris User Registration process. A Solaris Solve login ID and password should contain 6 to 8 alphanumeric characters without spaces and colons.
4. Access the Solaris Solve site.

**Note –** Solaris User Registration is not invoked if the system administrator or user is logged in as superuser.

If you choose to register, a copy of the completed form is stored in `$HOME/.solregis/uprops`. If you choose to never register and change your mind later, you can start User Registration by:

- Typing `/usr/dt/bin/solregis` at any command line prompt, or
- Clicking the Registration icon in the Application Manager's desktop tools folder (Common Desktop Environment desktop only)

For more information, see `solregis`(1).

# Troubleshooting Solaris User Registration Problems

This section provides troubleshooting tips for solving Solaris User Registration problems.

The following table describes problems that may occur when you try to register, and actions required to resolve these conflicts.

**TABLE 4–1** Registration Problem Descriptions and Suggested Resolutions

| Problem Description | How to Resolve the Problem |
|---|---|
| The registration form failed to initialize: Web page window displays and requests user see system administrator to resolve problem that prevents registration setup. | Check for missing registration files. |
| The form could not be emailed: Dialog box displays and requests user see system administrator to resolve problem. | Check to see if email is configured correctly. Also check if CDE is on user's system since it must be present to email completed registration form. Alternatively, users can print the form and fax or mail it. |
| The form could not be printed: Dialog box displays and requests user to see system administrator to resolve problem. | Check to see if the printer is configured correctly. Alternatively, the user can email form. |

**TABLE 4–1** Registration Problem Descriptions and Suggested Resolutions *(Continued)*

| Problem Description | How to Resolve the Problem |
|---|---|
| The form could not be saved: Dialog box displays and verifies that registration succeeded; however, the registration information cannot be recalled when updating registration in the future. | Check the user's home directory. Required action depends on the system's configuration. |
| You forgot your Solaris Solve login ID and password. | Send a mail message describing the problem to `SolarisSolve@sun.com` or see "How to Restart Solaris User Registration" on page 97. |
| You want to restart the registration process. | "How to Restart Solaris User Registration" on page 97. |

## ▼ How to Restart Solaris User Registration

Use the following procedure to restart the Solaris User Registration process.

1. **Change to the `$HOME/.solregis` directory.**

   ```
   % cd $HOME/.solregis
   ```

2. **Remove the `uprops` file.**

   ```
   % rm uprops
   ```

3. **Restart the registration process.**

   ```
   % /usr/dt/bin/solregis &
   ```

## ▼ How To Disable User Registration

You can disable User Registration before or after installing Solaris software. Before disabling Solaris User Registration, Sun recommends that system administrators register for their organization.

1. **To disable user registration before installing the Solaris release, select one of the following:**

   - Deselect the `SUNWsregu` package (interactive installation).
   - Modify a custom JumpStart profile to not install the `SUNWsregu` package.
   - Create and run a finish script that creates a file named `solregis` in the `/etc/default` directory on one or more systems with the following line in the script:

     ```
     DISABLE=1
     ```

   For more information see *Solaris 9 9/04 Installation Guide* or `solregis`(1).

2. **To disable user registration after installing the Solaris release, select one of the following:**

   - Remove the `SUNWsregu` package
   - Add the `solregis` file to the `/etc/default` directory.

CHAPTER 5

# Managing Server and Client Support (Overview)

This chapter describes the management of server and client support on a network, and it provides overview information about each system configuration (referred to as a *system type*) that is supported in the Solaris environment. 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 6.

## What's New in Server and Client Management?

This section describes new server and client management features in the Solaris 9 release.

### Diskless Client Support

In this Solaris release, you can manage diskless clients with the `smosservice` and `smdiskless` commands. Diskless clients are systems with no disks that depend on servers for all their services.

These commands are part of the Solaris Management Console tool suite. You cannot use the Solaris Management Console to manage diskless clients. You can only use the `smosservice` and `smdiskless` commands to manage diskless clients.

For more information on managing diskless clients, see "Diskless Client Management Overview" on page 104 and Chapter 6.

# Where to Find Server and Client Tasks

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

| Server/Client Services | For More Information |
|---|---|
| Install or JumpStart clients | *Solaris 9 9/04 Installation Guide* |
| Diskless client systems in the Solaris 9 environment | "Diskless Client Management Overview" on page 104 and Chapter 6 |
| 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 following:

| 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. |

| System Type | Description |
|---|---|
| Client | A system that uses remote services from a server. Some clients have no disk storage capacity and they have to rely on remote file systems from a server to function. Diskless systems and appliance systems are examples of this type of client.<br><br>Other clients might use remote services (such as installation software) from a server, but they don't rely on a server to function. A standalone system, which has its own hard disk containing the root (/), /usr, and /export/home file systems and swap space, is a good example of this type of client. |
| Appliance | A network appliance such as the Sun Ray appliance provides access to applications and the Solaris environment. An appliance gives you centralized server administration and no client administration or upgrades. Sun Ray appliances also provide *hot desking*, which is the ability 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 operating system (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, standalone systems and server systems mount these file systems from a local disk, while 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 5–1** Characteristics of General System Types

| System Type | Local File Systems | Local Swap? | Remote File Systems | Network Use | Relative Performance |
|---|---|---|---|---|---|
| Server | root (/)<br>/usr<br>/home<br>/opt<br>/export/home<br>/export/root | Yes | – *None* – | High | High |
| Standalone System | root (/)<br>/usr<br>/export/home | Yes | – *None* – | Low | High |
| Diskless Client | – *None* – | No | root (/)<br>swap<br>/usr<br>/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:

- Operating system (OS) services for diskless systems systems that run a different release or clients that are 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

## Standalone Systems

A *networked standalone system* can share information with other systems in the network, but it can continue to function if detached from the network.

A standalone system can function autonomously because it has its own hard disk that contains the root (/), `/usr`, and `/export/home` file systems and swap space. The standalone system thus has local access to operating system software, executables, virtual memory space, and user-created files.

---

**Note –** A standalone system requires sufficient disk space to hold its necessary file systems.

---

A *non-networked standalone system* is a standalone system with all the characteristics listed above, 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 operating system 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 104.

## 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, but the appliance user's desktop session is run on a server and displayed back to the user. The X environment is setup 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 from a centralized server, quickly and easily, 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 category. A ranking of 1 is most efficient. A ranking of 4 is least efficient.

**TABLE 5–2** Comparison of System Types

| System Type | Centralized Administration | Performance | Disk Usage |
|---|---|---|---|
| Standalone System | 4 | 1 | 4 |
| Diskless Client | 1 | 4 | 1 |
| Appliance | 1 | 1 | 1 |

# Diskless Client Management Overview

The following sections and Chapter 6 describe how to manage diskless client support in the Solaris 9 release.

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 management tools. In the Solaris 9 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 2.6 | Solaris 7 | Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02 | Solaris 9 |
|---|---|---|---|---|
| SPARC Servers | Supported | Supported | Supported | Supported |
| x86 Servers | Supported | Supported | Supported | Supported |
| SPARC Clients | Supported | Supported | Supported | Supported |
| x86 Clients | Not Supported | Not Supported | Not Supported | Supported |

This table describes the combination of OS server-client configurations that are supported by the `smosservice` and `smdiskless` commands.

| | Solaris 2.6 Release Support | Solaris 7 Release Support | Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02 Support | Solaris 9 Support |
|---|---|---|---|---|
| **OS Server-Client OS Release** | Solaris 2.6–Solaris 2.6 | Solaris 7–Solaris 2.6, or 7 | Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02–Solaris 2.6, 7, or 8 1/01, 4/01, 7/01, 10/01, 2/02 | Solaris 9–Solaris 2.6, 7, 8 1/01, 4/01, 7/01, 10/01, 2/02 |

# 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 on every system in the network.

You can do 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 environment

You can only use the diskless client commands to set up diskless client booting. You cannot use them 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 9 9/04 Installation Guide*.

# 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 5–3** 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:

  ```
  % /usr/sadm/bin/smdiskless add -p my-password -u my-user-name -- -h
  ```

- View the `smdiskless`(1M) or `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 of either a subset 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 5–4** Required Rights For Diskless Client Management

| RBAC Right | Command | Task |
|---|---|---|
| Basic Solaris User, Network Management | `smosservice list` | List OS services |
| | `smosservice patch` | List OS services patches |
| | `smdiskless list` | List diskless clients |
| 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 standalone system to an OS server with 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 Sun4m systems running the Solaris 8 release, you must add Sun4m/Solaris 8 OS services to the OS server. You would also still need to add OS services to support SPARC Sun4c systems or x86 based systems that runs the Solaris 8 release, because they are different platform groups.

You must have access to the appropriate Solaris 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 error messages saying that you have inconsistent versions of the OS running on the server and the OS that you are trying to add. This 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 may have a server that is running the Solaris 7 release. You may also have additional OS services loaded on this server, including the Solaris 2.6 SPARC sun4m OS services that have been patched. If you try to add the Solaris 2.6 SPARC sun4c 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, make sure 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 9 release, 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 5–5** Disk Space Requirements for OS Servers

| Directory | Required Space in Mbytes |
|---|---|
| `/export/Solaris_`*version* | 10 |
| `/export/exec` | 800 |
| `/export/share` | 5 |
| `/export/swap/`*diskless_client* | 32 (default size) |
| `/export/dump/`*diskless_client* | 32 (default size) |
| `/export/root/templates/Solaris_`*version* | 30 |
| `/export/root/clone/Solaris_`*version*`/`*machine_class* | 30 through 60 (depends on machine class) |
| `/export/root/`*diskless_client* (clone of above) | 30 through 60 (depends on machine class) |

**TABLE 5–5** Disk Space Requirements for OS Servers *(Continued)*

| Directory | Required Space in Mbytes |
|---|---|
| `/tftpboot/inetboot.`*machine_class*`.Solaris_`*version* | 200 Kbytes per *machine_class*`.Solaris_`*version* |

CHAPTER 6

# Managing Diskless Clients (Tasks)

This chapter describes how to manage diskless clients in the Solaris environment.

For information on the procedures associated with managing diskless clients, see "Managing Diskless Clients (Task Map)" on page 111.

For overview information on managing diskless clients, see Chapter 5.

For information about managing clients with Solstice AdminSuite™ software, see *Solstice AdminSuite 2.3 Administration Guide*.

## Managing Diskless Clients (Task Map)

The following table identifies the procedures needed to manage diskless clients.

| Task | Description | For Instructions |
|---|---|---|
| 1. (Optional) Remove existing diskless client support | If you have existing diskless clients that were added with the Solstice AdminSuite product, remove the diskless client support and OS services with the `admhostdel` and `admhostmod` commands before installing the Solaris release. | *Solstice AdminSuite 2.3 Administration Guide* |

| Task | Description | For Instructions |
|---|---|---|
| 3. (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 40 |
| 4. Prepare for adding a diskless client | Verify supported releases and identify the *platform*, *mediapath*, and *cluster* (or software group) of each diskless client. | "How to Prepare for Adding Diskless Clients" on page 114 |
| 5. Add required OS services to an OS server | Add the OS services for the diskless clients you want to support with 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 115 |
| 6. Add a diskless client | Add diskless client support by specifying all required information with the `smdiskless` command. | "How to Add a Diskless Client" on page 117 |
| 7. Boot the diskless client | Verify that the diskless client support is successfully added by booting the diskless client. | "How to Boot a Diskless Client" on page 118 |
| 8. (Optional) Delete diskless client support | Delete support for a diskless client if it is no longer required. | "How to Delete Diskless Client Support" on page 119 |
| 9. (Optional) Delete OS services for a diskless client | Delete OS services for a diskless client if they are no longer needed. | "How to Delete OS Services for Diskless Clients" on page 119 |
| 10. (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 121 |

# Managing Diskless Clients

These sections describe the procedures needed to manage 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 contains 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 108.
- 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`.

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

Make sure that the system 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.

When you use the `smosservice add` command to add OS services, you must specify the *platform*, *mediapath*, and *cluster* (or software group) of each diskless client platform that you want to support.

1. **Verify that the intended OS server and diskless client will be running a combination of Solaris releases that are supported.**

   For more information, see "OS Server and Diskless Client Support Information" on page 105.

2. **Identify the diskless client platform by using this format:**

   *instruction_set.machine_class.Solaris_version*

   For example:

   `sparc.sun4u.Solaris_9`

   The following are the possible platform options:

| *instruction_set* | *machine_class* | *Solaris_version* |
|---|---|---|
| sparc | sun4d*, sun4c*, sun4m*, sun4u, | Solaris_9, Solaris_8, Solaris_2.7, Solaris_2.6 |
| i386 | i86pc | Solaris_9, Solaris_8, Solaris_2.7, Solaris_2.6 |

   * The sun4c architecture is not supported in the Solaris 8 or Solaris 9 releases. The sun4d architecture is not supported in the Solaris 9 releases.

3. **Identify the media path, which is the full path to the disk image that contains the operating system that you want to install for the diskless client.**

   The Solaris operating system 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:

   - Create an install image on a server. For information on setting up an install server, refer to *Solaris 9 9/04 Installation Guide*.
   - Load the required OS services from the CD image using one of the following scripts:
     - CD 1 of 2 – `/cdrom/cdrom0/s0/Solaris_9/Tools/setup_install_server`

- CD 2 of 2 – `/cdrom/cdrom0/s0/Solaris_9/Tools/add_to_install_server`
- Language CD – `/cdrom/cdrom0/s0/Solaris_9/Tools/add_to_install_server`

For example, if you are using the `setup_install_server` script from the Solaris 9 Software 1 of 2 SPARC Platform Edition CD on a locally connected CD-ROM device, the syntax looks something like this:

```
# mkdir /export/install/sparc_9
# cd /cd_mount_point/Solaris_9/Tools
# ./setup_install_server /export/install/sparc_9
```

- After the Solaris CD image is installed on the disk, specify the disk image path. For example:

```
/net/export/install/sparc_9
```

4. **Identify the `SUNWCXall` cluster when you add OS services.**

   You must use *the same cluster* for diskless clients that run the same operating system on the same system (SPARC or x86).

   For example, consider the following 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 situation, 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.

1. **Become superuser or assume an equivalent role on the server.**

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 34.

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 starbug:898 --
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from ...
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
was successful.
Platform
--------------------------------------------------------------------------
```

3. **Add the OS services.**

   # **/usr/sadm/bin/smosservice add -H** *hostname*:**898 -- -o** *hostname*
   **-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` *hostname: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 approximately 45 minutes, depending on the server speed and the OS service configuration you choose.

For more information, see `smosservice`(1M).

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** *hostname*:**898 --**

## Example—Adding an OS Service for Diskless Client Support

This example describes how to add Solaris 8 OS services on the server `starbug`. The server `starbug` is running the Solaris 9 release.

```
# /usr/sadm/bin/smosservice add -H starbug:898 -- -o starbug
-x mediapath=/net/install/export/sparc_8 -x platform=sparc.sun4u.Solaris_8
-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 :: xxx
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
was successful.
```

# ▼ How to Add a Diskless Client

Use this procedure to add a diskless client after you have added OS services.

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

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 34.

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=timezone -x locale=locale-name
   ```

| | |
|---|---|
| `add` | Adds the specified diskless client. |
| `--` | Identifies that the subcommand arguments start after this pont. |
| `-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=`*timezone* | Specifies the timezone for the diskless client. |
| `-x locale=`*locale-name* | Specifies the locale to install for the diskless client. |

For more information, see `smdiskless`(1M).

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** *hostname*:898 --

## Examples—Adding a Diskless Client

This example shows how to add a Solaris 8 client, `holoship`, from the server `starbug`.

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.103 -e 8:0:20:92:4e:f3
-n holoship -x os=sparc.sun4u.Solaris_8 -x root=/export/root/holoship
-x swap=/export/swap/holoship -x swapsize=128 -x tz=US/Mountain
-x locale=en_US
```

This example shows how to add a Solaris 7 client, `inquisitor`, from the server `starbug`.

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.102 -e 8:0:20:1f:31:be
-n inquisitor -x os=sparc.sun4u.Solaris_2.7 -x root=/export/root/inquisitor
-x swap=/export/swap/inquisitor -x swapsize=64 -x tz=US/Mountain
```

# ▼ How to Boot a Diskless Client

1. **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.

2. **Boot the diskless client.**

   ok **boot net**

## ▼ How to Delete Diskless Client Support

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

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 34.

2. **Remove the diskless client support.**

   # **/usr/sadm/bin/smdiskless delete -- -o** *hostname*:**898 -n** *client-name*

3. **Verify that the diskless client support is removed.**

   # **/usr/sadm/bin/smosservice list -H** *hostname*:**898 --**

### Example—Deleting Diskless Client Support

This example shows how to delete the diskless client `holoship` from the OS server `starbug`.

```
# /usr/sadm/bin/smdiskless delete -- -o starbug -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.
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
was successful.
```

## ▼ How to Delete OS Services for Diskless Clients

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

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 34.

2. **Remove the OS services for the diskless clients.**

   # **/usr/sadm/bin/smosservice delete -H** *hostname*:**898 --**
   **-x rmplatform=***instruction-set.machine-class.Solaris-version*

3. **Verify that the OS services are removed.**

```
# /usr/sadm/bin/smosservice list -H hostname:898 --
```

## Example—Deleting OS Services for Diskless Clients

The following example shows how to delete the diskless client OS services (`sparc.all.Solaris_9`) from the server `starbug`.

```
# /usr/sadm/bin/smosservice delete -H starbug: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 ::
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug: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 an Unsigned Solaris Patch" on page 276.

## 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_8 -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

1. **Become superuser or assume an equivalent role on the server.**

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 34.

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
   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 starbug
   Login to starbug as user root was successful.
   Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
   was successful.
   ```

   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 the patch is spooled.**

   ```
   # /usr/sadm/bin/smosservice patch -- -P
   ```

5. **Push the spooled patch to the diskless client.**

   ```
   # /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 starbug
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
was successful.
```

---

**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
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 starbug
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
was successful.
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
```

## Example—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
.
.
.
# /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
```

```
.
.
.
# /usr/sadm/bin/smosservice patch -- -m -U
Authenticating as user: root
.
.
.
# /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

- OS server does not respond to client RARP requests
- OS server does not respond to client `bootparam` requests
- OS server cannot mount diskless client root file system

Solution

*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 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.

*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 and swap areas, as follows:

*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 that 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.
- Verify that 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, 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.
- Verify that 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*=2.6, 2.7, 8, 9, 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 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 you specified when you added the diskless client.

  Verify that the following entries exists 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 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
- Incorrect server responds to diskless client's RARP request

Solution
Restart the following daemons on the OS server:

```
# /usr/sbin/rpc.bootparamd
# /usr/sbin/in.rarpd -a
```

CHAPTER 7

# Shutting Down and Booting a System (Overview)

This chapter provides guidelines for shutting down and booting a system. The Solaris software environment is designed to run continuously so that electronic mail and network resources are available to users. Occasionally, it is necessary to shut down or reboot a system 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 new features that are related to shutting down and booting a system in the Solaris 9 release.

## PXE Network Boot

You can boot the Solaris x86 Platform Edition 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.

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, and the system will boot from the disk drive.

Some early versions of PXE firmware cannot boot the Solaris system. If you have one of these older versions, your system can read the PXE network bootstrap program from a boot server, but 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 `elxl`(7D) and`iprb`(7D).

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 183.

# Where to Find Shutting Down and Booting 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 9 |
| Boot a SPARC based system | Chapter 10 |

| Shut Down and Boot Task | For More Information |
|---|---|
| Boot an x86 based system | Chapter 11 |
| Manage a SPARC based system with the power management software | `power.conf`(4), `pmconfig`(1M) |

# Shutting Down and Booting 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 is used to perform transitions between run levels. System administrators use the `init` command to initiate a run-level transition. This book refers to init states as run levels.

  For more information about run levels, see "Run Levels" on page 135.
- **Boot types** – A *boot type* describes how a system is booted. Different boot types 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, because 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 – Means that 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. 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 – Means that 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 8.

# Guidelines for Booting a System

Keep the following in mind when you boot a system:

- After a system is shut down, it is booted by using the `boot` command at the PROM level on a SPARC based system or by using the `boot` command at the Primary Boot Subsystem Menu on an x86 based system.
- A system can be rebooted by turning the power off and then back on. This method is not a clean shutdown because system services and processes are terminated abruptly. However, turning a system's power off and back on is an alternative for emergency situations.
- SPARC based systems and x86 based systems use different hardware components for booting. These differences are described in Chapter 12.

# 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.

In addition, there are two network configuration boot strategies available:

- RARP (Reverse Address Resolution Protocol and ONC+ RPC Bootparams Protocol)
- DHCP (Dynamic Host Configuration Protocol)

The default network boot strategy is set to RARP.

Use this table if you need information on booting a system over the network.

| Network Boot Task | For More Information |
|---|---|
| Boot a SPARC system or a SPARC diskless client | Chapter 10 |
| Boot an x86 system or an x86 diskless client | Chapter 11 |
| Boot a DHCP client during installation | *Solaris 9 9/04 Installation Guide* |
| Configure a DHCP client with DHCP Manager | *System Administration Guide: IP Services* |

# When to Shut Down a System

The following table provides a list of system administration tasks and the type of shut down that is needed to initiate the task.

**TABLE 7–1** Shutting Down a System

| Reason for System Shut Down | 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 9 |
| To change kernel parameters in the `/etc/system` file | Run level 6 (reboot the system) | Chapter 9 |
| To perform file system maintenance, such as backing up or restoring system data | Run level S (single-user level) | Chapter 9 |
| To repair a system configuration file such as `/etc/system` | See "When to Boot a System" on page 132 | N/A |
| To add or remove hardware from the system | Reconfiguration boot (also to turn off power when adding or removing hardware) | "Managing Devices (Tasks)" in *System Administration Guide: Devices and File Systems* |

**TABLE 7–1** Shutting Down a System *(Continued)*

| Reason for System Shut Down | Appropriate Run Level | For More Information |
|---|---|---|
| To repair an important system file which is causing system boot failure | See "When to Boot a System" on page 132 | N/A |
| To boot the kernel debugger (`kadb`) to track down a system problem | Run level 0, if possible | Chapter 9 |
| To recover from a hung system and you want to force a crash dump | See "When to Boot a System" on page 132 | N/A |

For examples of shutting down a server or a standalone system, see Chapter 9.

# When to Boot a System

The following table provides a list of system administration tasks and the corresponding boot type that is used to complete the task.

**TABLE 7–2** Booting a System

| Reason for System Reboot | Appropriate Boot Type | Information for SPARC Procedure | Information for x86 Procedure |
|---|---|---|---|
| To turn off system power due to anticipated power outage | Turn system power back on | Chapter 9 | Chapter 9 |
| To 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 166 | "x86: How to Boot a System to Run Level 3 (Multiuser Level)" on page 179 |
| To perform file system maintenance, such as performing a backup or restoring system data | Use 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 167 | "x86: How to Boot a System to Run Level S (Single-User Level)" on page 180 |
| To repair a system configuration file such as `/etc/system` | Interactive boot | "SPARC: How to Boot a System Interactively" on page 168 | "x86: How to Boot a System Interactively" on page 181 |

**TABLE 7–2** Booting a System *(Continued)*

| Reason for System Reboot | Appropriate Boot Type | Information for SPARC Procedure | Information for x86 Procedure |
|---|---|---|---|
| To 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* |
| To boot the kernel debugger (`kadb`) to track down a system problem | Booting `kabd` | "SPARC: How to Boot the System With the Kernel Debugger (`kadb`)" on page 173 | "x86: How to Boot a System With the Kernel Debugger (`kadb`)" on page 189 |
| 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 171 | "x86: How to Boot a System for Recovery Purposes" on page 184 |
| To recover from a hung system and you want to force a crash dump | Recovery boot | See example on "SPARC: How to Force a Crash Dump and Reboot the System" on page 174 | See example on "x86: How to Force a Crash Dump and Reboot the System" on page 190 |

For examples of booting a system, see Chapter 10 or Chapter 11.

CHAPTER 8

# Run Levels and Boot Files (Tasks)

This chapter provides overview information and tasks that are related to run levels and boot files.

This is a list of the step-by-step instructions in this chapter.



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



## 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 environment 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 8–1** 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 environment. |
| 4 | Alternative multiuser state | | Currently unavailable. |
| 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). |

# How to Determine 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, except run level 0.

## Example—Determining a System's Run Level

This example displays information a system's current run level and information about previous run levels.

```
$ who -r
 .    run-level 3[1]  Dec 13 10:10[2]   3[3]   0[4] S[5]
$
```

1. Identifies the current run level
2. Identifies the date of last run level change
3. Also identifies the current run level
4. Identifies the number of times the system has been at this run level since the last reboot
5. Identifies the previous run level

# The /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 three important items for the `init` process:

- The system's default run level
- 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 8–2** 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. |

**TABLE 8–2** Fields Descriptions for the `inittab` File *(Continued)*

| Field | Description |
|---|---|
| *action* | Identifies how the process that is specified in the process field is to be run. Possible values include: `initdefault`, `sysinit`, `boot`, `bootwait`, `wait`, and `respawn`.<br><br>`initdefault` identifies the default run level. For a description of the other action keywords, see `inittab`(4). |
| *process* | Defines the command or script to execute. |

# Example—Default `inittab` File

The following example shows an annotated default `inittab` file that is installed with the Solaris release:

```
1 ap::sysinit:/sbin/autopush -f /etc/iu.ap
2 ap::sysinit:/sbin/soconfig -f /etc/sock2path
3 fs::sysinit:/sbin/rcS sysinit   >/dev/msglog 2<>/dev/msglog </dev/console
4 is:3:initdefault:
5 p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/...
6 sS:s:wait:/sbin/rcS              >/dev/msglog 2<>/dev/msglog </dev/console
7 s0:0:wait:/sbin/rc0              >/dev/msglog 2<>/dev/msglog </dev/console
8 s1:1:respawn:/sbin/rc1           >/dev/msglog 2<>/dev/msglog </dev/console
9 s2:23:wait:/sbin/rc2             >/dev/msglog 2<>/dev/msglog </dev/console
10 s3:3:wait:/sbin/rc3             >/dev/msglog 2<>/dev/msglog </dev/console
11 s5:5:wait:/sbin/rc5             >/dev/msglog 2<>/dev/msglog </dev/console
12 s6:6:wait:/sbin/rc6             >/dev/msglog 2<>/dev/msglog </dev/console
13 fw:0:wait:/sbin/uadmin 2 0      >/dev/msglog 2<>/dev/msglog </dev/console
14 of:5:wait:/sbin/uadmin 2 6      >/dev/msglog 2<>/dev/msglog </dev/console
15 rb:6:wait:/sbin/uadmin 2 1      >/dev/msglog 2<>/dev/msglog </dev/console
16 sc:234:respawn:/usr/lib/saf/sac -t 300
17 co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: "
   -T terminal-type -d /dev/console -l console
-m ldterm,ttcompat
```

1. Initializes STREAMS modules
2. Configures socket transport providers
3. Initializes file systems
4. Defines default run level
5. Describes a power fail shutdown
6. Defines single-user level
7. Defines run level 0
8. Defines run level 1
9. Defines run level 2
10. Defines run level 3
11. Defines run level 5
12. Defines run level 6
13. Defines an unused level, firmware

14. Defines an unused level, off
15. Defines an unused level, reboot
16. Initializes Service Access Controller
17. Initializes console and identifies the terminal type

## 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. Identifies the `initdefault` entry, which defines the default run level (3).
   b. Executes any process entries that have `sysinit` in the `action` field so that any special initializations can take place before users login.
   c. Executes any process entries that have a 3 in the `rstate` field, which matches the default run level, 3.

      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 8–3** 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 8–4** 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. |

**TABLE 8–4** Command Descriptions for Run Level 3 *(Continued)*

| Command or Script Name | Description |
|---|---|
| `/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. |
| `/usr/lib/saf/sac -t 30` | Starts the port monitors. This process is restarted if it fails. |
| `/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: " -T` *terminal_type* `-d /dev/console -l console` | Starts the `ttymon` process that monitors the console for login requests. This process is restarted if it fails.<br>The *terminal_type* on a SPARC based system is `sun`.<br>The *terminal_type* on an x86 based system is `AT386`. |

# Run Control Scripts

**Note –** The way system services are started and stopped in the Solaris environment might change in some future release.

The Solaris software environment 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*          S05RMTMPFILES*     S72inetsvc*          S88utmpd*
K03sshd*           S10lu*             S72slpd*             S89PRESERVE*
```

```
K05appserv*       S20sysetup*        S73cachefs.daemon*  S89bdconfig@
K05volmgt*        S21perf*           S73nfs.client*      S90wbem*
K06mipagent*      S30sysid.net*      S74autofs*          S91zuluinit*
K07dmi*           S40llc2*           S74syslog*          S93cacheos.finish*
K07snmpdx*        S42ncakmod*        S74xntpd*           S94ncalogd*
K15imq*           S47pppd*           S75cron*            S95IIim*
K16apache*        S69inet*           S75flashprom*       S95svm.sync*
K21dhcp*          S70sckm*           S75savecore*        S98efcode*
K27boot.server*   S70uucp*           S76nscd*            S99audit*
K28kdc*           S71ldap.client*    S77sf880dr*         S99dtlogin*
K28kdc.master*    S71rpc*            S80lp*              S99emul64*
K28nfs.server*    S71sysid.sys*      S80spc*             S99rcapd*
README            S72autoinstall*    S85power*
S01MOUNTFSYS*     S72directory*      S88sendmail*
```

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 also 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:

- Mounts all local file systems if the previous run level was S
- Enables disk quotas if at least one file system was mounted with the quota option
- Saves temporary editor files in the /usr/preserve directory
- Removes any files and subdirectories in the /tmp directory
- Starts system activity data collecting, system accounting, and system auditing, if configured
- Starts the system logging daemon (syslogd), sets the default dump device, and rotates the /var/adm/messages file
- Sets the default scheduling class if the /etc/dispadmin.conf file exists
- Starts LP print service (lpsched) if a local printer is configured and cleans up the print queue
- Configures power management, if appropriate
- Starts the utmpd daemon
- Starts the cron and vold daemons
- Configures serial device stream
- Configures WBEM services
- Syncs volumes, if required, and starts the mdmonitord daemon to monitor the physical components of the volumes
- Starts the CDE desktop login process, dtlogin, if appropriate

Network service or security-related tasks:

- Configures the network interfaces, sets ifconfig netmask, and configures network routing, if appropriate
- Starts network service (inetd and rpcbind) daemons
- Starts the logical link controller (llc2), if configured
- Sets the name service domain name, starts various name services daemons, depending on if the system is configured for a name service, and whether the system is a client or a server
- Starts the keyserv, statd, lockd, and xntpd daemons, if appropriate
- Mounts all NFS entries

- 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 LDAP cache manager (`ldap_cachemgr`), if configured
- Starts directory server (`slapd`) daemon, if configured
- Starts DNS (`in.named`) daemon, if configured
- Starts Service Location Protocol (`slpd`) daemon, if configured
- Configures system resource controls and system pools if the `/etc/rctladm.conf` and `/etc/pooladm.conf` files exist
- Starts the `cachefsd`, `automount`, and `sendmail` daemons, if appropriate
- 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
- Starts the Sun Fire 880 Dynamic Reconfiguration daemon (`sf880drd`), if appropriate
- Runs the flash PROM update script
- Configures any graphic frame buffers or graphic accelerators
- Runs the FCode interpreter daemon (`efdaemon`), if necessary

Transitions the following services between run-level changes:

- Apache (`tomcat`)
- Boot server (`in.rarpd`), (`rpc.bootparamd`), or (`rpld`)
- DHCP (`in.dhcpd`)
- Kerberos KDC (`krb5kdc` ) and Kerberos administration (`kadmind`)
- Mobile IP (`mipagent`)
- NFS server (`nfsd`), (`mountd`), (`nfslogd`)
- Samba (`smdb`) and (`nmdb`)
- Secure shell (`sshd`)
- Solstice Enterprise Agents (`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 the DHCP daemon (in.dhcpd), if appropriate
- Starts Kerberos KDC (krb5kdc) and Kerberos administration (kadmind) daemons, if configured
- Starts Mobile IP daemon (mipagent), if configured
- Starts the Samba daemons (smdb and nmdb), if configured
- Starts the secure shell daemon (sshd), if appropriate
- Starts the Solstice Enterprise Agents (dmispd and snmpXdmid)
- Cleans up the /etc/dfs/sharetab file
- Starts the NFS server daemons nfsd, mountd, and nfslogd, if appropriate
- If the system is a boot server, starts the rarpd, rpc.bootparamd, and rpld daemons

## 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:

- Establishes a minimal network
- Checks and mounts root (/), /usr, /tmp, /var, /var/adm, and /var/run file systems.
- Sets the system name
- Mounts pseudo file systems (/proc and /dev/fd)
- Rebuilds the device entries for reconfiguration boots

- Checks and mounts other file systems to be mounted in single-user level

## Using a Run Control Script to Stop or Start Services

**Note –** The way system services are started and stopped in Solaris environment might change in some future release.

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.

### ▼ How to Use a Run Control Script to Stop or Start a Service

1. **Become superuser.**
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—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
341
343
```

```
347
345
# pgrep -f nfs -d, | xargs ps -fp
 UID   PID  PPID  C    STIME TTY      TIME CMD
  daemon   341     1  0   Aug 21 ?        0:00 /usr/lib/nfs/statd
    root   343     1  0   Aug 21 ?        0:00 /usr/lib/nfs/lockd
    root   347     1  0   Aug 21 ?        0:41 /usr/lib/nfs/nfsd
    root   345     1  0   Aug 21 ?        0:02 /usr/lib/nfs/mountd
```

# Adding a Run Control Script

**Note –** The way system services are started and stopped in the Solaris environment might change in some future release.

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.

## ▼ How to Add a Run Control Script

1. **Become superuser.**
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/ /etc/rc2.d/ /etc/rcn.d/
   ```

### Example—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/S100xyz
# ln xyz /etc/rc0.d/K100xyz
# ls /etc/init.d /etc/rc2.d /etc/rc0.d
```

## Disabling 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.

### ▼ How to Disable a Run Control Script

1. **Become superuser.**
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—Disabling a Run Control Script

The following example shows how to rename the `S100datainit` script.

```
# cd /etc/rc2.d
# mv S100datainit _S100datainit
# ls _*
# _S100datainit
```

# x86: Boot Files

In addition to the run control scripts and boot files described previously, there are additional boot files that are associated with booting a Solaris x86 system.

**TABLE 8–5** 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. |
| `/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 standalone `kadb`. In addition, this executable provides some boot firmware services. |
| `/boot/solaris/boot.rc` | Prints the Solaris x86 Platform Edition 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 9

# 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 8.

## 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 as 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 7.

For information on using your system's power management features, see *Solaris Common Desktop Environment: User's Guide*.

# 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, which means that 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, it is sometimes necessary to use these actions in emergency situations. For instructions on system recovery techniques, see Chapter 10 or Chapter 11.

The following table describes the various shutdown commands and provides recommendations for using them.

**TABLE 9–1** 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 running at run level 3 because users are notified of the impending shut down. 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 syncs the disks before changing run levels. | Recommended for standalone 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 syncs the disks and passes boot instructions to the `uadmin` system call, which, in turn, stops the processor. | Not recommended. Use the `init` command instead. |
| `halt` | An executable that syncs the disks and stops the processor. | Not recommended because it doesn't execute the `/etc/rc0` script. This script stops all processes, syncs the disks, and unmounts any remaining file systems. |

# 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 onto the system and all systems that are mounting resources from the affected system.

For this reason, the `shutdown` command is recommended over 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`(1) 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. See "How to Determine a System's Run Level" on page 136.

## ▼ How to Determine Who Is Logged in to a System

1. **Log into the system to be shut down.**
2. **Display logged-in users.**

   ```
   $ who
   ```

### Example—Determining Who Is Logged in to a System

The following example shows how to display who is logged in to the system.

```
$ who
holly [1]     console      May  7 07:30
kryten      pts/0  [2]     May  7 07:35      (starbug) [4]
lister      pts/1        May  7 07:40 [3] (bluemidget)
```

1. Identifies the user name of the logged-in user.
2. Identifies the terminal line of the logged-in user.
3. Identifies the date and time that the user logged in.
4. (Optional) Identifies the host name if a user is logged in from a remote system.

## ▼ How to Shut Down a Server

Use this procedure when you need to shut down a server.

1. **Become superuser.**
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
   ```

| | |
|---|---|
| -i*init-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. |
| -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 `shutdown`(1M).

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): xxx
```

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 System Prompt | x86 System Prompt |
|---|---|---|
| S (single-user level) | # | # |
| 0 (power-down level) | `ok` or `>` | `type any key to continue` |
| Run level 3 (multiuser level with remote resources shared) | *hostname* `console login:` | *hostname* `console login:` |

## SPARC: Example—Bringing a Server to Run Level S

In the following example, the `shutdown` is used to bring a SPARC based system to run level S (single-user level) in 3 minutes.

```
# who
root       console      Dec 13 14:30
# shutdown -g180 -y

Shutdown started.    Thu Dec 13 14:30:32 MST 2001
```

```
Broadcast Message from root (console) on earth Thu Dec 13 14:30:33...
The system earth will be shut down in 3 minutes
.
.
.
Broadcast Message from root (console) on earth Thu Dec 13 14:30:33...
The system earth 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.
Dec 13 14:34:00 earth syslogd: going down on signal 15
Killing user processes: done.

INIT: SINGLE USER MODE

Type control-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Entering System Maintenance Mode ...
#
```

## SPARC: Example—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      Dec 12 08:08
rimmer     pts/0        Dec 11 14:48    (starbug)
pmorph     pts/1        Dec 13 12:31    (bluemidget)
# shutdown -i0 -g300 -y
Shutdown started.    Thu Dec 13 14:51:39 MST 2001

Broadcast Message from root (console) on earth Thu Dec 13 14:51:39...
The system earth 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 156.

## SPARC: Example—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 without requiring additional confirmation.

```
# who
root       console      Dec 12 08:08
rimmer     pts/0        Dec 11 14:48    (starbug)
pmorph     pts/1        Dec 13 12:31    (bluemidget)
# shutdown -i6 -g120 -y
Shutdown started.    Thu Dec 13 15:56:30

Broadcast Message from root (console) on earth Thu Dec 13 15:56:30...
The system earth 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...
.
.
.
earth console login:
```

## Where to Go From Here

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 10 or Chapter 11.

# ▼ How to Shut Down a Standalone System

Use this procedure when you need to shut down a standalone system.

1. **Become superuser.**

2. **Shut down the system.**

   # **init** *run-level*

   *run-level* identifies the new run level.

   For more information, see `init`(1M).

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 System Prompt | x86 System Prompt |
|---|---|---|
| S (single-user level) | # | # |
| 2 (multiuser level) | # | # |
| 0 (power-down level) | ok or > | `type any key to continue` |
| 3 (multiuser level with NFS resources shared) | *hostname* `console login:` | *hostname* `console login:` |

## x86: Example—Bringing a Standalone System to Run Level 0

In the following example, the `init` command is used to bring an x86 based standalone 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
Type any key to continue
```

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 156.

## SPARC: Example—Bringing a Standalone System to Run Level S

In the following example, the `init` is used to bring a SPARC based standalone 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.
INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Entering System Maintenance Mode
#
```

### Where to Go From Here

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 10 or Chapter 11.

## Turning Off Power to All Devices

You need turn off power to all system devices is 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 like an approaching electrical storm

System devices to power down include the CPU, the monitor, and external devices such as disks, tapes, and printers.

Before you turn off power to all system devices, you should shutdown the system cleanly, as described in the preceding sections.

### ▼ How to Turn Off Power to All Devices

1. **Select one of the following to shut down the system.**
   a. **If shutting down a server, see "How to Shut Down a Server" on page 151.**
   b. **If shutting down a standalone system, see "How to Shut Down a Standalone System" on page 154.**

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 10

# 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 system, see "SPARC: Booting a System (Task Map)" on page 159.

For overview information about the boot process, see Chapter 7. 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 11.

# 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 do one of the following: | |
| | Identify the PROM revision number. | "SPARC: How to Find the PROM Revision for a System" on page 161 |
| | Identify devices on the system to boot from. | "SPARC: How to Identify Devices on a System" on page 161 |

| Task | Description | For Instructions |
| --- | --- | --- |
| | Change the default boot device when a new disk is added or when you need to change the system boot method. | "SPARC: How to Change the Default Boot Device" on page 163 |
| Boot the system | Select one of the following boot methods: | |
| | Boot to run level 3 - Used after shutting down the system or performing some system hardware maintenance task. | "SPARC: How to Boot a System to Run Level 3 (Multiuser Level)" on page 166 |
| | Boot to run level S - Used after performing some system maintenance task such as backing up a file system. At this level, only local file systems are mounted and users cannot log into the system. | "SPARC: How to Boot a System to Run Level S (Single-User Level)" on page 167 |
| | Boot interactively - Used after making temporary changes to a system file or the kernel for testing purposes. | "SPARC: How to Boot a System Interactively" on page 168 |
| | 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 From the Network" on page 169 |
| | 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: | |
| | First, stop the system to attempt recovery. | "SPARC: How to Stop the System for Recovery Purposes" on page 170 |
| | Boot to repair an important system file that is preventing the system from booting successfully. | "SPARC: How to Boot a System for Recovery Purposes" on page 171 |
| | Boot `kadb` - Used to troubleshoot system problems. | "SPARC: How to Stop the System for Recovery Purposes" on page 170 |

| Task | Description | For Instructions |
|---|---|---|
| | 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 the System" on page 174 |

# SPARC: Using the Boot PROM

System administrators typically use the PROM level to boot a system. Occasionally, however, 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.

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 standalone system from the network

For a complete list of PROM commands, see `monitor`(1M) or `eeprom`(1M).

## SPARC: How to Find the PROM Revision for a System

Display a system's PROM revision level with 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 figure out what are the appropriate devices to boot from.

Before you can safely use the `probe` commands to find out what devices are attached to the system, you need to do the following:

- Change the PROM `auto-boot?` parameter to false and

- Issue the `reset-all` command to clear system registers

You can the `probe` commands that are available on your system by using the `sifting probe` command, as follows:

```
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
```

1. **Change the PROM auto-boot? parameter to false.**

   ```
   ok setenv auto-boot? false
   ```

2. **Clear the system's registers.**

   ```
   ok reset-all
   ```

3. **Identify the devices on the system.**

   ```
   ok probe-device
   ```

4. **(Optional) If you want the system to reboot after a power failure or after using the `reset` command, then change the `auto-boot?` parameter back to true.**

   ```
   ok setenv auto-boot? true
   auto-boot? =          true
   ```

5. **Boot the system back to multiuser mode.**

   ```
   ok reset
   ```

## SPARC: Examples—Identifying the Devices on a System

The following example shows how to identify the devices connected to an Ultra10 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
```

You can use the `devalias` command to identify the device aliases and the associated paths of devices that *might* be connected to the system.

```
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 161.

1. **Become superuser.**
2. **Change to run level 0.**

   ```
   # init 0
   ```

   The `ok` PROM prompt is displayed.

   For more information, see `init`(1M).

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 with identifying the disk number.

4. **Verify that the default boot device is changed.**

   ```
   ok printenv boot-device
   ```

5. **Save the new `boot-device` value.**

   ```
   ok reset
   ```

   The new `boot-device` value is written to the PROM.

## SPARC: Examples—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 and 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 8.

## ▼ 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.

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 `boot`(1M).

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:`

## SPARC: Example—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: disk:a  File and args:
SunOS Release 5.9 Version Generic 64-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
configuring IPv4 interfaces: hme0.
Hostname: starbug
The system is coming up.  Please wait.
checking ufs filesystems
/dev/rdsk/c0t0d0s7: is clean.
/dev/rdsk/c0t0d0s4: is clean.
NIS domainname is Solar.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 starbug
syslog service starting.
Print services started.
volume management starting.
The system is ready.

starbug console login:
```

## ▼ 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.

1. **Boot the system to run level S.**

   ```
   ok boot -s
   ```

2. **Type the superuser password when the following message is displayed.**

   ```
   INIT: SINGLE USER MODE
   Type Ctrl-d to proceed with normal startup,

   (or give root password for system maintenance):  xxx
   ```

3. **Verify that the system is at run level S.**

   ```
   # who -r
   .       run-level S  Jun 10 15:27     3      0
   ```

4. **To bring the system up to multiuser state after you completed the system maintenance task, press Control-D.**

## SPARC: Example—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 5.9  Generic May 2002
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
configuring IPv4 interfaces: hme0.
Hostname: starbug

INIT: SINGLE USER MODE

Type control-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Sun Microsystems Inc.   SunOS 5.9  Generic May 2002
# who -r
   .       run-level S  Sep 19 08:49     S      0  ?
(Perform some maintenance task)
# Press Control-D
```

## ▼ SPARC: How to Boot a System Interactively

Use this procedure to boot a system and you need to specify an alternate kernel or `/etc/system` file.

1. **Boot the system interactively.**

   ```
   ok boot -a
   ```

2. **Answer the system prompts as described in the following table.**

| System Prompt | Action |
|---|---|
| `Enter filename [kernel/[sparcv9]/unix]:` | Provide the name of kernel to use for booting. Or, press Return to use the default kernel. |
| `Enter default directory for modules [/platform/‘uname -i‘/kernel /platform/‘uname -m/kernel /kernel /usr/kernel]:` | Provide an alternate path for the `modules` directory. Or, press Return to use the default kernel modules directory. |
| `Name of system file [etc/system]:` | Provide the name of an alternate system file and press Return. Type `/dev/null` if your `/etc/system` file has been damaged. Or, press Return to use the default `etc/system` file. |
| `root filesystem type [ufs]:` | Press Return to use the default root (`/`) file system. Type UFS for local disk booting, or NFS for network booting. |
| `Enter physical name of root device` [*physical_device_name*]: | Provide an alternate device name and press Return. Or, press Return to use the default physical name of the root device. |

3. **If you are not prompted to answer the questions in the preceding table, verify that you typed the `boot -a` command correctly.**

## SPARC: Example—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 5.9 Version Generic 64-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
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: starbug
The system is coming up.  Please wait.
checking ufs filesystems
.
.
.
The system is ready.
starbug console login:
```

## ▼ SPARC: How to Boot a System From the Network

Any system can boot from the network if there is a boot server available. You might want to boot a standalone system from the network temporarily 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 163.

Two network configuration boot strategies are available on sun4u 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 strategy 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 finding your PROM version, see "SPARC: How to Find the PROM Revision for a System" on page 161.

---

If both methods are available, you can specify which service to use in the `boot` command temporarily. 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
```

This alias means that when you type `boot net`, the system boots by using the DHCP network book strategy.

**Caution –** 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*.

1. **If necessary, shut down the system.**
2. **Determine the method for booting from the network and select one of the following:**

   You must have already set up a RARP or DHCP boot server in your network to use either method to boot successfully.

   a. **Boot the system from the network by using the DHCP method.**

      ```
      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 method.**

      ```
      ok boot net[:rarp]
      ```

      Since 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

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 for your system again.**
4. **Type the appropriate `boot` command to start the boot process.**

   For more information, see `boot`(1M).

5. **Verify that the system is booted to the specified run level.**

```
# who -r
 .       run-level 3  May  2 07:39     3      0  S
```

## SPARC: Example—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.

Substitute the device name of the file system to be repaired for the *devicename* variable in the following procedures. If you need help identifying a system's device names, refer to "Accessing Devices (Overview)" in *System Administration Guide: Devices and File Systems*.

1. **Stop the system by using the system's stop key sequence.**

   Use the stop 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 "SPARC: How to Stop the System for Recovery Purposes" on page 170.

2. **Follow the instructions in the table, depending on whether you are booting from the Solaris installation CD or DVD or from the network.**

| Boot Type | Action |
|---|---|
| Solaris installation CD or DVD | 1. Insert the Solaris installation media into the drive.<br>2. Boot from the installation media in single-user mode:<br>`ok boot cdrom -s` |
| The network if an installation server or remote CD or DVD drive are available | Use the following command:<br>`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:
```

## SPARC: Example—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
```

## SPARC: Example—Booting a System if You Forgot Root Password

The following example shows how to recover when you forget the root password by booting from the network. 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: How to Boot the System With the Kernel Debugger (kadb)

1. **If you need to stop the system, 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
   ```

3. **When you see the `syncing file systems...` message, press the stop key sequence for your system again.**

4. **Boot the system with the kernel debugger.**

   ```
   ok boot kadb
   ```

5. **Check `kadb` boot messages to verify that the system has booted with the kernel debugger.**

   ```
   Rebooting with command: kadb
   Boot device: /iommu/sbus/espdma@4,800000/esp@4,8800000/sd@3,0
   .
   .
   .
   ```

### SPARC: Example—Booting the System With the Kernel Debugger (kadb)

```
Press Stop-A
ok sync
syncing file systems...
Press Stop-A
ok boot kadb
```

## 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 "Managing System Crash Information (Tasks)" in *System Administration Guide: Advanced Administration*.

### ▼ SPARC: How to Force a Crash Dump and Reboot the System

Use this procedure to force a crash dump and reboot the system when the `savecore` feature is enabled.

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:`

## SPARC: Example—Forcing a Crash Dump and Rebooting the System

*Press Stop-A*

```
ok sync
```

CHAPTER 11

# x86: Booting a System (Tasks)

This chapter describes the procedures for booting an x86 based system.

For information on the procedures associated with booting an x86 system, see “x86: Booting a System (Task Map)” on page 177.

For overview information about the boot process, see Chapter 7.

For step-by-step instructions on booting a SPARC based system, see Chapter 10.

## x86: Booting a System (Task Map)

| Task | Description | For Instructions |
|---|---|---|
| Boot the Solaris Device Configuration Assistant | 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 Boot the Solaris Device Configuration Assistant” on page 179 |
| Boot the system | Select one of the following boot methods:<br>Boot to run level 3 – Used after shutting down the system or performing some system hardware maintenance task. | “x86: How to Boot a System to Run Level 3 (Multiuser Level)” on page 179 |

| Task | Description | For Instructions |
|---|---|---|
| | Boot to run level S - Used after performing some system maintenance task such as backing up a file system. | "x86: How to Boot a System to Run Level S (Single-User Level)" on page 180 |
| | Boot interactively – Used after making temporary changes to a system file or the kernel for testing purposes. | "x86: How to Boot a System Interactively" on page 181 |
| | Boot from the network - Used to boot a system from the network. This method is used for booting a diskless client. | "x86: How to Boot a System From the Network" on page 183 |
| | 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: | |
| | First, stop the system to attempt recovery. | "x86: How to Stop a System for Recovery Purposes" on page 184 |
| | Boot to repair an important system file that is preventing the system from booting successfully. | "x86: How to Boot a System for Recovery Purposes" on page 184 |
| | Boot `kadb` – Used to troubleshoot system problems. | "x86: How to Boot a System With the Kernel Debugger (`kadb`)" on page 189 |
| | Force a crash dump and reboot the system - Used to force a crash dump for troubleshooting purposes. | "x86: Forcing a Crash Dump and Rebooting the System" on page 190 |

# x86: Booting the Solaris Device Configuration Assistant

The Device Configuration Assistant (Solaris x86 Platform Edition) is a program that enables you to perform various hardware configuration and booting tasks. You can access the Solaris Device Configuration Assistant from either of the following:

- Solaris boot diskette

- Solaris Installation CD or DVD

In the procedures in this chapter, you might be requested to insert the Solaris Device Configuration Assistant boot diskette to boot the Configuration Assistant. If your system's BIOS supports booting from the CD or DVD, you can, instead, insert the Solaris installation CD or DVD to boot the Configuration Assistant.

### ▼ x86: How to Boot the Solaris Device Configuration Assistant

1. **Insert the Solaris Device Configuration Boot Diskette or the Solaris Installation CD or DVD in the appropriate drive.**
2. **If the system displays the `Type any key to continue` 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 first menu of the Configuration Assistant is displayed after a few minutes.

## 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 the Ctrl-Alt-Del keys 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.

1. **If the system displays the `Type any key to continue` 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:`

## x86: Example—Booting a System to Run Level 3 (Multiuser Level)

```
Type any key to continue
           .
           .
           .
                     <<< 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
             .
             .
             .
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.

1. **If the system displays the `Type any key to continue` 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 by using the `who -r` command.**

```
# 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. **Press Control-D to bring the system back to run level 3.**

## x86: Example—Booting a System to Run Level S (Single-User Level)

```
Type any key to continue
            .
            .
            .

                     <<< 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
               .
               .
               .
INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance):  xxx
Entering System Maintenance Mode
               .
               .
               .
# who -r
   .       run-level S  Jul 19 14:37     S      0  3
(Perform some maintenance task)
# Press Control-D
```

## ▼ x86: How to Boot a System Interactively

Use this procedure to boot a system and you need to specify an alternate kernel or `/etc/system` file.

1. **If the system displays the `Type any key to continue` 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 and 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 system prompts as described in the following table.**

| System Prompt | Action |
|---|---|
| `Enter default directory for modules: [/platform/i86pc/kernel /kernel /usr/kernel]:` | Provide an alternate path for the `modules` directory and press Enter. Or, press Enter to use the default `modules` directory path. |
| `Name of system file [etc/system]:` | Provide the name of an alternate system file and press Enter. Or, press Enter to use the default `/etc/system` file. Type `/dev/null` if your `/etc/system` file has been damaged. |
| `root filesystem type [ufs]:` | Press Enter to use the default root (`/`) file system. Type: UFS for local disk booting, or NFS for network booting. |
| `Enter physical name of root device [`*physical_device_name*`]:` | Provide an alternate device name and press Enter. Or, press Enter to use the default physical name of the root device `bootpath`. |

## x86: Example—Booting a System Interactively

In the following example, the default choices (shown in square brackets `[]`) are accepted.

```
Type any key to continue
            .
            .
            .

                    <<< 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 -a
Enter default directory for modules [/platform/i86pc/kernel /kernel
/usr/kernel]: Press Enter
Name of system file [etc/system]:Press Enter
SunOS Release 5.9 Version Generic 32-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
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
configuring IPv4 interfaces: dnet0.
Hostname: venus
(fsck messages)
The system is coming up.  Please wait
(More messages)
venus console login:
```

## ▼ x86: How to Boot a System From the Network

Any system can boot from the network if there is a boot server available. You might want to boot a standalone system from the network temporarily if the system cannot boot from the local disk.

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 Configuration Assistant boot diskette or the installation CD or DVD.

The menu, Set Network Configuration Strategy, on the Configuration Assistant's Boot Tasks Menu, enables you to select the appropriate boot strategy.

1. **Determine whether you want to boot from the network by using the RARP/bootparams method or the DHCP method.**

   There are two network configuration strategies to choose from, RARP (Reverse Address Resolution Protocol) or DHCP (Dynamic Host Configuration Protocol). The default network boot strategy is set to RARP. You can use either strategy depending on whether a RARP boot server or a DHCP boot server is available on your network.

   The PXE network boot is available only with DHCP.

2. **Insert the Configuration Assistant boot diskette or the installation CD or DVD that you wish to boot from.**

   Or, use the system or network adapter BIOS configuration program to enable the PXE network boot.

3. **If the system displays the `Type any key to continue` 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 Solaris Device Configuration Assistant screen is displayed.

4. **Press the F2 key (F2_Continue) to scan for devices.**

   Device identification is performed and the Identified Devices screen is displayed.

5. **Press the F2 key (F2_Continue) to load drivers.**

   Bootable drivers are loaded.

   The Boot Solaris menu is displayed.

6. **Press the F4 key (F4_Boot Tasks).**

7. **Select `Set Network Configuration Strategy` and press the F2 key (F2_Continue).**

8. **Select either `RARP` or `DHCP` and press the F2 key (F2_Continue).**

   A screen that confirms your new network boot strategy appears.

   Your network boot strategy selection is saved as the default network boot method for the next time this diskette is used for booting.

9. **Press F3_Back to return to the Boot Solaris menu.**

10. **Select `NET` as the boot device. Then, press F2_Continue to boot the network device.**

    The Solaris boot option screen is displayed.

## ▼ x86: How to Stop a System for Recovery Purposes

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 `Type any key to continue` 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 Installation 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 *devicename* variable in the following procedure. If you need help with identifying a system's device names, refer to "Accessing Devices (Overview)" in *System Administration Guide: Devices and File Systems*.

1. **Stop the system first by using the system stop key sequence.**

   Use the stop 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 184.

2. **Boot from the Solaris installation CD or DVD (or from the network) to single-user mode.**

   a. **Insert the Configuration Assistant boot diskette or the installation CD or DVD that you wish to boot from.**

   b. **If the system displays the `Type any key to continue` 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 Solaris Device Configuration Assistant screen is displayed.

   c. **Press the F2 key (F2_Continue).**

      Device identification is performed and the Identified Devices screen is displayed.

   d. **Press the F2 key (F2_Continue).**

      Bootable drivers are loaded.

      The Boot Solaris menu is displayed.

   e. **Select the CD-ROM drive or network device. Then press the F2 key (F2_Continue).**

      The Current Boot Parameters menu is displayed.

   f. **Type `b -s` at the prompt. Press Enter.**

      After a few minutes, the single-user mode # prompt is displayed.

3. **Mount the root (/) file system that contains the invalid `passwd` file.**

   ```
   # mount /dev/dsk/devicename /a
   ```

4. **Change to the newly mounted `etc` directory.**

   ```
   # cd /a/etc
   ```

5. **Make the necessary change to the file by using an editor.**

   ```
   # vi filename
   ```

6. **Change to the root (/) directory.**

```
# cd /
```

7. **Unmount the /a directory.**

```
# umount /a
```

8. **Reboot the system.**

```
# init 6
```

9. **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:`

## x86: Example—Booting a System for Recovery Purposes

The following example shows how to repair the `/etc/passwd` file after booting from a local CD-ROM.

```
Type any key to continue

SunOS Secondary Boot version 3.00


Solaris Intel Platform Edition Booting System

Running Configuration Assistant...
Autobooting from Boot path: /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.

Press ESCape to interrupt autoboot in 5 seconds.
      .
      .
      .
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].

[ ]  NET : DEC 21142/21143 Fast Ethernet
on Board PCI at Dev 3
[ ]  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
[X]  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
      .
      .
      .
               <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args: kernel/unix -r

Select the type of installation you want to perform:

1 Solaris Interactive
2 Custom JumpStart
3 Solaris Web Start

Enter the number of your choice followed by <ENTER> the key.

If you enter anything else, or if you wait for 30 seconds,
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: Example—Booting a System if You Forgot Root Password

The following example shows how to recover when you forget the root password by booting from the network. This example assumes that the boot server is already available. Be sure to apply a new root password after the system has rebooted.

```
Type any key to continue

SunOS Secondary Boot version 3.00
```

```
Solaris Intel Platform Edition Booting System

Running Configuration Assistant...
Autobooting from Boot path: /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.

Press ESCape to interrupt autoboot in 5 seconds.
      .
      .
      .
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]  NET : DEC 21142/21143 Fast Ethernet
on Board PCI at Dev 3
[ ]  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
      .
      .
      .
               <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args: kernel/unix -r

Select the type of installation you want to perform:

1 Solaris Interactive
2 Custom JumpStart
3 Solaris Web Start

Enter the number of your choice followed by <ENTER> the key.

If you enter anything else, or if you wait for 30 seconds,
an interactive installation will be started.
```

```
Select type of installation:  b -s
      .
      .
      .
# mount /dev/dsk/c0t0d0s0 /a
      .
      .
      .
# cd /a/etc
# vi shadow
(Remove root's encrypted password string)
# cd /
# umount /a
# init 6
```

## ▼ x86: How to Boot a System With the Kernel Debugger (kadb)

1. **If the system displays the `Type any key to continue` 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.

2. **Type `b kadb` to boot the kernel debugger. 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:`

4. **Verify that you can access the kernel debugger by pressing F1-A.**

   The `kadb[0]:` prompt is displayed when you enter the kernel debugger.

## x86: Example—Booting a System With the Kernel Debugger (kadb)

```
Type any key to continue
        .
        .
        .
            <<< 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 kadb
        .
        .
        .
naboo console login: (Enter login and password)
(Press F1-A to verify that you can access the kernel debugger)
```

# 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 "Managing System Crash Information (Tasks)" in *System Administration Guide: Advanced Administration*.

## ▼ x86: How to Force a Crash Dump and Reboot the System

The system must be booted with the kernel debugger option, `kadb`, to get to the `kadb[0]:` prompt and to enable you to force the crash dump.

---

**Note –** You must be in text mode to enter the kernel debugger (`kadb`). So, first exit any window system.

---

1. **Press F1–A.**

   ```
   kadb[0]:
   ```

   The `kadb[0]:` prompt is displayed.

2. **Type the following commands at the `kadb[0]:` prompt.**

   ```
   Press <F1-a>
   kadb[0]: vfs_syncall/W ffffffff
   kadb[0]: 0>eip
   kadb[0]: :c
   ```

```
kadb[0]: :c
kadb[0]: :c
```

After you type the first `:c` , the system panics, so you need to type `:c` again. The system panics again, so type `:c` a third time to force the crash dump and reboot the system.

After the crash dump is written to disk, the system continues to reboot.

3. **Verify that the system has rebooted by logging in at the console login prompt.**

CHAPTER 12

# 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 10 or Chapter 11.

## SPARC: The Boot PROM

Each SPARC based system has a PROM (programmable read-only memory) 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 checks 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 12–1** 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. Then, the PROM loads the primary boot program, `bootblk`, whose 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, which starts other processes by reading the `/etc/inittab` file. |
| `init` | 7. The `/sbin/init` process starts the run control (`rc`) scripts, which execute a series of other scripts. These scripts (`/sbin/rc*`) check and mount file systems, start various processes, and perform system maintenance tasks. |

# 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

At three points during the Solaris boot process, you can make the following choices about a booting system as follows:

- **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 non-Solaris operating system, you cannot reach the next two menus.
- **Interrupt the Autoboot Process** – If the autoboot process is interrupted, you can access the Solaris Device Configuration Assistant.

  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 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 customize the boot.

The following table summarizes the purpose of the primary x86 boot interfaces. See the sections that follow for a detailed description and example of each boot interface.

**TABLE 12–2** x86: Boot Subsystems

| Boot Subsystem | Purpose |
|---|---|
| Primary Boot Subsystem | This menu appears if the disk you are booting from contains multiple operating systems, including the Solaris operating system. |
| 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. |

**TABLE 12–2** x86: Boot Subsystems *(Continued)*

| Boot Subsystem | Purpose |
|---|---|
| Solaris Device Configuration Assistant/Boot Diskette | There are two ways to access the Solaris Device Configuration Assistant menus:<br>1. Use the Solaris Device Configuration Assistant boot diskette or the Solaris installation CD (on systems that can boot from the CD-ROM drive) to boot the system.<br>2. 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://soldc.sun.com/support/drivers/dca_diskettes`.

During the boot process, the boot subsystem menus allow you to customize boot choices. If the system receives no response during the time-out 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.

The following section provides examples of each boot subsystem screen.

# x86: Booting the Solaris Release

During the device identification phase, the Solaris 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 and editing devices and their resources

During the boot phase, the Solaris 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 Solaris Device Configuration Assistant attempts to identify devices on the system.

## x86: Configuration Assistant Screen

This screen appears each time you boot the Solaris Device Configuration Assistant. The Solaris Device Configuration Assistant runs every time the system is booted, although the autoboot process bypasses the menus.

```
Solaris Device Configuration Assistant

The Solaris(TM) (Intel Platform Edition) 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 Solaris 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 Solaris 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 perform optional device tasks, such as setting a keyboard configuration, viewing and editing devices, setting up a serial console, and 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.

## 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 time-out 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 12–3** 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 physical 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`, which contains disk information needed to find the active partition and the location of the Solaris boot program, `pboot`, loads and executes `pboot`. |
| | 4. The Solaris boot program, `pboot` loads `bootblk`, the primary boot program, whose purpose is to load the secondary boot program that 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-second interval to select an alternate partition from which to boot. This step only occurs if there is more than one bootable partition present on the system. |
| | 6. `bootblk` finds and executes the secondary boot program, `boot.bin` or `ufsboot`, in the root (`/`) file system. You have a 5–second interval 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, which 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. |

**TABLE 12–3** x86: Description of the Boot Process *(Continued)*

| Boot Phase | Description |
|---|---|
| | 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. The `/sbin/init` process starts the run control (`rc`) scripts, which execute a series of other scripts. These scripts (`/sbin/rc*`) check and mount file systems, start various processes, and perform system maintenance tasks. |

CHAPTER 13

# Managing Software (Overview)

The management of software involves adding and removing software from standalone systems, servers, and their clients. This chapter describes background and other information about the various tools available for installing and managing software.

This chapter does not describe installing the Solaris software on a new system, nor does it describe installing or upgrading a new version of the Solaris software. For information on installing or upgrading Solaris software, see *Solaris 9 9/04 Installation Guide*.

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



For step-by-step instructions on managing software, see Chapter 14.

## What's New in Software Management in the Solaris 9 Update Releases?

This section describes a new software management feature in this Solaris release.

## pkgadd and patchadd Support for Signed Packages and Patches

**Solaris 9 12/03** – This Solaris release enables you to securely download Solaris packages and patches that include a digital signature by using the updated `pkgadd` and `patchadd` commands.

In previous Solaris releases, you could download the Solaris patch management tools and use the `smpatch` command with PatchPro to manage signed patches. For step-by-step instructions on using the `smpatch` command to manage signed patches, see "Managing Signed Patches by Using Solaris Patch Management Tools (Tasks)" in *Signed Patches Administration Guide for PatchPro 2.2*.

For overview information about signed packages, see "Overview of Software Packages" on page 206.

For step-by-step instructions on using the `patchadd` command to add signed patches, see "Adding Signed Patches With `patchadd` Command (Task Map)" on page 269.

For step-by-step instructions on using the `pkgadd` command to add signed packages, see "Adding and Removing Signed Packages (Task Map)" on page 243.

## prodreg Command Enhancements

**Solaris 9 4/03** – You can now use several options to the `prodreg` command to access and manage the Solaris Product Registry from the command line.

For information on using the `prodreg` command to administer software packages, see "Managing Software With the Solaris Product Registry Command-Line Interface (Task Map)" on page 224.

# What's New in Software Management in the Solaris 9 Release?

This section describes new software management features in the Solaris 9 release.

## Signed Patches

All patches that are available for the Solaris 2.6, 7, 8, and 9 releases include a digital signature. A valid digital signature ensures that the patch has not been modified since the signature was applied.

Using signed patches is a secure method of downloading or applying patches because the patches include a digital signature that can be verified before the patch is applied to your system.

Signed patches are stored in Java™ archive format files (*abc*`.jar`) and are available from the SunSolve Online℠ Web site.

For information about adding signed patches with the `smpatch` command, see "Managing Signed Patches by Using Solaris Patch Management Tools (Tasks)" in *Signed Patches Administration Guide for PatchPro 2.2*.

## Solaris Product Registry 3.0

The Solaris Product Registry 3.0 is a GUI tool that enables you to install and uninstall software packages.

For information on using this product to manage software packages, see "Managing Software With the Solaris Product Registry GUI (Task Map)" on page 220.

## Patch Analyzer

When you use the Solaris™ Web Start program to upgrade to a Solaris 9 Update Release, the patch analyzer performs an analysis on your system to determine which (if any) patches will be removed or downgraded by upgrading to the Solaris Update Release. You do not need to use the Patch Analyzer when you upgrade to the Solaris 9 release.

For information on using this tool when you are upgrading to a Solaris 9 update release, see "Upgrading to a Solaris Update Release (Tasks)" in *Solaris 9 9/04 Installation Guide*.

## Solaris Management Console Patch Manager

The Solaris Management Console provides a new Patches Tool for managing patches. You can only use the Patches Tool to add patches to a system running the Solaris 9 release.

For information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 40.

# 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 9 9/04 Installation Guide* |
| Adding or removing Solaris software packages after installation | Chapter 14 |
| Adding or removing Solaris patches after installation | Chapter 16 |
| Troubleshooting software package problems | "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 operating environment 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 15.

# 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 adding patches because the digital signature can be verified before the patch is added to your system.

For more information about *adding* signed patches to your system, see "Adding Signed Patches With `patchadd` Command (Task Map)" on page 269.

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 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.

The following table describes the general terms associated with signed packages and patches.

| Term | Definition |
| --- | --- |
| Keystore | A repository of certificates and keys that is queried when needed.<br>■ Java keystore – A repository of certificates that is installed by default with the Solaris release.<br>The Java keystore is usually stored in the `/usr/j2se/jre/lib/security` directory.<br>■ Package keystore – A repository of certificates that you import when adding signed packages and patches to your system.<br>The package keystore is stored in the `/var/sadm/security` directory by default. |

| Term | Definition |
| --- | --- |
| 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.<br><br>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 adding a signed package or patch to your system involves three basic steps:

1. Adding the certificates to your system's package keystore with the `pkgadm` command
2. (Optional) Listing the certificates with the `pkgadm` command
3. Adding the package with the `pkgadd` command or adding the patch with the `patchadd` command

For step-by-step instructions on adding signed packages to your system, see "Adding and Removing Signed Packages (Task Map)" on page 243. For step-by-step instructions on adding signed patches to your system, see "Adding Signed Patches With `patchadd` Command (Task Map)" on page 269.

## Using Sun's Certificates to Verify Signed Packages and Patches

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:

- 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.

The following table describes the encryption terminology associated with signed packages and patches.

| Term | Definition |
| --- | --- |
| ASN.1 | Abstract Syntax Notation 1 (ASN.1) is a way to express a set of abstract objects. For example, ASN.1 defines a public key certificate, all of the objects that make up the certificate, and the order in which the objects are collected. However, ASN.1 does not specify how the objects are serialized for storage or transmission. |
| base64 | base64 is a method of encoding arbitrary binary data as ASCII text. |
| DER | Distinguished Encoding Rules (DER) is a binary representation of an ASN.1 object. DER defines how an ASN.1 object is serialized for storage or transmission in computing environments. |
| PEM | The Privacy Enhanced Message (PEM) is a way to encode a file (in DER or other binary format) using base64 encoding and some optional headers. Initially used for encoding MIME-type email messages. PEM is also used extensively for encoding certificates and private keys into a file that exists on a file system or in an email message. |
| PKCS7 | The Public Key Cryptography Standard #7 (PKCS7) describes a general syntax for data that may have cryptography applied to it, such as digital signatures and digital envelopes. |
| X.509 | The International Telecommunication Union-Telcom (ITU-T) recommendation X.509 specifies the widely adopted X.509 public key certificate syntax.<br><br>This recommendation defines a framework for the provision of authentication services. X.509 describes two levels of authentication:<br>■ Simple authentication – using a password as a verification of claimed identity.<br>■ Strong authentication – involving credentials formed using cryptographic techniques. While simple authentication offers some limited protection against unauthorized access, use only strong authentication as the basis for providing secure services. |

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 keystore.

All Sun certificates are issued by Baltimore Technologies, which recently bought GTE CyberTrust.

Access to a keystore is protected by a special password that you specify when you import the Sun certificates into your system's 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 table describes the output of the `pkgadm listcert` command.

| Field | Description |
|---|---|
| Keystore Alias | When you retrieve certificates for printing, signing, or removing, this name must be used to reference the certificate. |
| Common 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:<br>■ 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.<br>■ Signing Certificate - A certificate that can be used when signing a package or patch. A private key is associated with a signing certificate. |
| Issuer Common 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.

## SunSolve Online's Trusted Certificates

SunSolve Online uses the following certificates to verify the digital signatures on signed patches with the previous Solaris patch management tools (`smpatch` command), including PatchPro:

- Top-level certificate, called the Root Certificate Authority (CA)
- A subordinate CA, which is the Sun Microsystems Inc., CA Class B certificate.
- An additional certificate issued by Sun Enterprise™ Services, called the *patch management certificate*

A *certificate authority* certifies the relationship between public keys that are used to decrypt the digital signature with the patch and the owner of the public keys.

The Sun Root CA, Sun Class B CA, and the patch signing certificate are included with the Solaris patch management tools, including PatchPro. These three certificates provide a certificate chain of trust in the patch verification process whereby the Sun Root CA trusts the Class B CA, and the Class B CA trusts the patch management certificate. And, ultimately, the GTE CyberTrust CA trusts the Sun Root CA.

## 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 adding 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 into the Package Keystore (`pkgadm addcert`)" on page 243.

For complete instructions on adding signed packages with the `pkgadd` command, see "Adding and Removing Signed Packages (Task Map)" on page 243.

# Tools for Managing Software Packages

The tools for adding and removing software packages from a system after the Solaris release is installed on a system are the following:

**TABLE 13–1** Software Package Tools

| Add, Remove, and Display Software Package Information With This Tool | Additional Features |
|---|---|
| The Solaris Web Start program | Launch an installer to add products included in the Solaris 9 media pack. You cannot add individual software packages. |
| Solaris Product Registry (GUI) | Launch an installer to add, remove, or display software product information. Use Product Registry to remove or display information about software products that were originally installed by using the Solaris Web Start program or the Solaris `pkgadd` command. |
| Solaris Product Registry `prodreg` Viewer (command line interface) | Use the `prodreg` command to remove or display information about software products that were originally installed by using the Solaris Web Start program or the Solaris `pkgadd` command. |

**TABLE 13–1** Software Package Tools *(Continued)*

| Add, Remove, and Display Software Package Information With This Tool | Additional Features |
|---|---|
| Package commands (`pkgadd`, `pkgrm`, `pkginfo`) | Incorporate these commands into scripts, set up optional files to avoid user interaction or perform special checks, and copy software packages to spool directories. |

# Adding or Removing a Software Package (`pkgadd`)

All the software management tools that are listed in Table 13–1 are used to add, remove, or query information about installed software. Admintool, the Solaris Product Registry `prodreg` viewer, and the Web Start program all 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 product 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 Web Start program, Solaris Product Registry `prodreg` viewer (either GUI or CLI), Admintool, 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 these tools 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, but that 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, which will update the software product database correctly so that the file is no longer a part of the package. For more information, see `removef`(1M).

If you intend to keep multiple versions of a package (for example, multiple versions of a document processing application), install new versions into a different directory than the already installed package with the `pkgadd` command. 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 "Avoiding User Interaction When Adding Packages (`pkgadd`)" on page 215 and `admin`(4).

---

**Note –** If you use the upgrade option when installing the Solaris software, the Solaris installation software consults 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.

**Caution –** 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 will prevent 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 with 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), which specifies where the package will be installed. For more information, see `admin`(4).

## 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 or not optional pieces of the package should be installed.

If prior to installation, you know that the package you want to install is an interactive package, and you want to store your answers to prevent user interaction during future installations of this package, you can 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 14

# Managing Software (Tasks)

This chapter describes how to add, verify, and remove software packages.

For information on the procedures associated with performing software management tasks, see:

- "How to Install Software With the Solaris Web Start Program" on page 219
- "Managing Software With the Solaris Product Registry GUI (Task Map)" on page 220
- "Managing Software With the Solaris Product Registry Command-Line Interface (Task Map)" on page 224
- "Adding and Removing Signed Packages (Task Map)" on page 243
- "Managing Software Packages With Package Commands (Task Map)" on page 248
- "Adding and Removing Software Packages With Admintool (Task Map)" on page 257

## Commands for Managing Software Packages

The following table lists the commands to use for adding, removing, and checking the installation of software packages after the Solaris release is installed.

**TABLE 14–1** Tools or Commands for Managing Software Packages

| Tool or Command | Man Page | Description |
|---|---|---|
| `admintool` | `admintool`(1M) | Installs or removes a software package with a graphical tool. |

**TABLE 14–1** Tools or Commands for Managing Software Packages *(Continued)*

| Tool or Command | Man Page | Description |
|---|---|---|
| `installer` | `installer`(1M) | Installs or removes a software package with an installer. |
| `pkgadd` | `pkgadd`(1M) | Installs a signed or unsigned software package. |
| `pkgadm` | `pkgadm`(1M) | Maintains the keys and certificates used to manage signed packages and signed patches. |
| `pkgchk` | `pkgchk`(1M) | Checks the installation of a software package. |
| `pkginfo` | `pkginfo`(1) | Lists software package information. |
| `pkgparam` | `pkgparam`(1) | Displays software package parameter values. |
| `pkgrm` | `pkgrm`(1M) | Removes a software package. |
| `prodreg` | `prodreg`(1M) | Browse, unregister, and uninstall software in the Solaris Product Registry. |
| `pkgtrans` | `pkgtrans`(1) | 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. |

# Adding Software With the Solaris Web Start Program

This section describes how to use the Solaris Web Start program to add software to a system on which you have installed the Solaris operating system. The Solaris Web Start program installs only the components of the software groups that you skipped when you initially installed the Solaris operating system. You cannot upgrade to another software group after installing or upgrading. For a description of the four software groups, see *Solaris 9 9/04 Installation Guide*.

## ▼ How to Install Software With the Solaris Web Start Program

**Note –** This procedure assumes that the system is running volume management (`vold`). If your system is not running volume management, see "Accessing Removable Media (Tasks)" in *System Administration Guide: Devices and File Systems* for information on accessing removable media without volume management.

1. **Become superuser or assume an equivalent role.**
2. **Decide to install 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 9 Languages CD, the Solaris Web Start program starts automatically. Proceed to Step 6.
   - 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 Web Start installer.**

   Solaris Web Start installers are located in various directories on the CDs and on the DVD.
   - Solaris 9 Software 1 of 2 CD.
   - Solaris 9 Software 2 of 2 CD.
   - Solaris 9 Documentation CD.
   - Solaris 9 Languages CD. The Solaris Web Start program automatically starts when the CD is inserted.

   For specific information about CD and DVD structures, see "Organization of Solaris 9 Media (Reference)" in *Solaris 9 9/04 Installation Guide*.
4. **Follow the instructions to install the software.**
   - From a file manager, double-click Installer or installer.
   - From the command line, type the following:

     ```
     % ./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. |

Follow the instructions to install the software.

5. **Double-click Installer or installer.**
   An Installer window is displayed, followed by the Solaris Web Start dialog box.
6. **Follow the directions on the screen to install the software.**
7. **When you have finished adding software, click Exit.**
   The Solaris Web Start program exits.

# Managing Software With the Solaris Product Registry GUI (Task Map)

The following task map describes the software management tasks that you can do with the Solaris Product Registry.

| Task | Description | For Instructions |
|---|---|---|
| View installed or uninstalled software with the Product Registry | You can view installed or uninstalled software with the Product Registry. | "How to View Installed or Uninstalled Software Information With the Product Registry GUI" on page 222 |
| Install software with the Product Registry | You can use Product Registry to find software and launch the Solaris Web Start program, which leads you through the installation of that software. | "How to Install Software With the Product Registry GUI" on page 222 |
| Uninstall software with the Product Registry | You can uninstall software with the Product Registry. | "How to Uninstall Software With the Product Registry GUI" on page 223 |

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 Web Start program 3.0 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 224.

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 areas 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

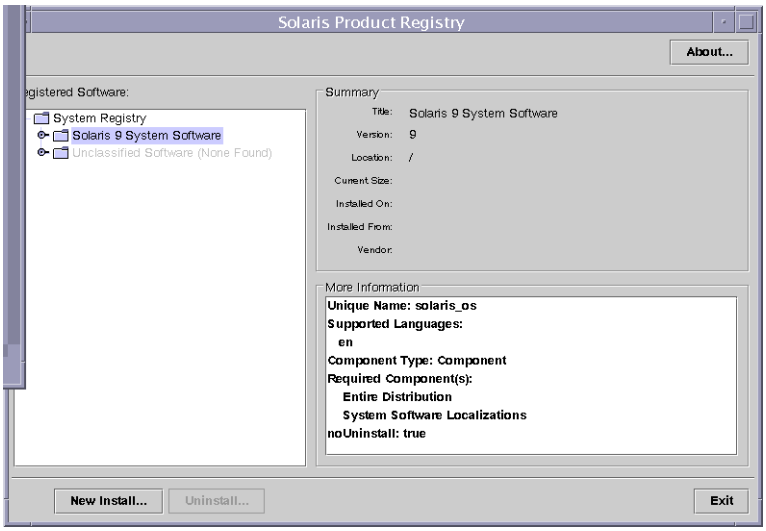



**FIGURE 14–1** Solaris Product Registry Window

## ▼ How to View Installed or Uninstalled Software Information With the Product Registry GUI

1. **Become superuser or assume an equivalent role.**
2. **Start the 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.**

   Notice that the turner control changes from pointing to the right to pointing down. 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:

   - The configuration software group that you chose when installing the Solaris release. Software groups that can be displayed include Core, End User System Support, Developer System Support, Entire Distribution, or Entire Distribution Plus OEM Support.
   - Additional system software, which is Solaris products that are not part of the software group you chose.
   - Unclassified software, which is any package that you installed by using the `pkgadd` command that is not a Solaris product or part of the software group.
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 Web Start program, the 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 Product Registry GUI

You can use Product Registry to find software and launch the Solaris Web Start program, which leads you through the installation of that software.

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

2. **Start the Product Registry tool.**

   ```
   # prodreg
   ```

   The Solaris Product Registry 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 Product Registry displays the Select Installer dialog box, which initially points to the `/cdrom` directory or the directory you are in.

6. **Select directories to find the Solaris Web Start program installer.**

   Solaris Web Start installers are located in various directories on the CDs and on the DVD. For specific information about CD and DVD structures, see "Organization of Solaris 9 Media (Reference)" in *Solaris 9 9/04 Installation Guide*.

   - Solaris 9 Software 1 of 2 and 2 of 2 CD.
   - Solaris 9 Software 2 of 2 CD.
   - Solaris 9 Documentation CD.
   - Solaris 9 Languages CD. The Solaris Web Start program 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 Product Registry GUI

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

2. **Start the Product Registry tool.**

   ```
   # prodreg
   ```

The Solaris Product Registry 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 can do with the Solaris Product Registry command-line interface.

| Task | Description | For Instructions |
|---|---|---|
| View installed or uninstalled software with `prodreg` | You can view software information with the `browse` subcommand. | "How to View Installed or Uninstalled Software Information (`prodreg`)" on page 225 |
| View software attributes with `prodreg` | You can view specific software attributes with the `info` subcommand. | "How to View Software Attributes (`prodreg`)" on page 228 |
| Check dependencies between software components with `prodreg` | You can view the components that depend on a specific software component with the `info` subcommand. | "How to Check Dependencies Between Software Components (`prodreg`)" on page 230 |
| Identify damaged software products with `prodreg` | 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 231 |

| Task | Description | For Instructions |
|---|---|---|
| Uninstall software with `prodreg` | You can remove software from your system with the `uninstall` subcommand. | "How to Uninstall Software (`prodreg`)" on page 234 |
| Uninstall damaged software with `prodreg` | 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 238 |
| Reinstall damaged software components with `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. | "How to Reinstall Damaged Software Components (`prodreg`)" on page 241 |

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 man page `prodreg`(1M).

## ▼ 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.

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 9 4/03
                                                            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 operating system in this sample output is the Solaris 9 4/03 release.

3. **Browse the information for one of the software components that are listed in the Solaris Product Registry.**

   ```
   % prodreg browse  -m "name"
   ```

   | | |
   |---|---|
   | `-m` *"name"* | Displays information on the software component with the name *name* |

   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
   ```

   | | |
   |---|---|
   | `-u` *name-UUID* | Displays information on the *name* software component with the unique identifier *name-UUID* |
   | `-i` *instance* | Displays information on *name* software component with the instance number *instance* |

4. **Repeat Step 3 for each software component you want to browse.**

## Example—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 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg browse -m "Solaris 9 4/03 System Software"
```

## Example—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 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg browse -n 2
```

## Example—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.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             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 to the `prodreg` command. The `prodreg info` command displays a variety of information about registered software, including the following items:

- Name of software component
- Software component description
- Required components of the software
- Other components that require the software
- Base directory of the software
- Path to the software component

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 9 4/03
                                                           System
                                                           Software
    3         +    8f64eabf-1dd2-11b2-a3f1-0800209a5b6b 1 Unclassified
                                                           Software
```

3. **View the attributes for one of the software components that are listed in the Solaris Product Registry.**

   You can view the attributes for one of the software components that are listed in the output of the `prodreg info` command in the following ways.

   ```
   % prodreg info  -m "name"
   ```

   | | |
   |---|---|
   | `-m` *"name"* | Displays the attributes of the software component with the name *name* |

4. **Repeat Step 3 for each software component you want to view.**

## Example—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  9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg info -m "Solaris 9 4/03 System Software"
```

## Example—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 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg info -n 2
```

## Example—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.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg info -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
```

# ▼ How to Check Dependencies Between Software Components (prodreg)

You can use the `prodreg info` command to view the components that depend on a specific software component. You might want to check dependencies between software products before you uninstall specific components.

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 9 4/03
                                                         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 in the Solaris Product Registry. See "How to View Installed or Uninstalled Software Information (`prodreg`)" on page 225 for more information on browsing the Solaris Product Registry with 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 the components that depend on *name* software by displaying the values of the Dependent Components attribute. |

This command outputs a list of the software components that depend on *name* software.

## Example—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 to the `prodreg` command to help you determine if a software product is damaged.

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 9 8/03
                                                        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` column. 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 `isDamaged=TRUE` output 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—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 9 8/03
                                                          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` column of the Solaris Product Registry. The `ExampleSoft` software might be damaged. 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 of the previous command indicates that the `ExampleSoft` software is damaged. Use the `-a PKGS` option to 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
```

Use the `pkginfo` command to verify that the `EXSOzzt` and `EXSOblob` packages are installed on the system.

```
% 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.

## ▼ How to Uninstall Software (`prodreg`)

You can use the `uninstall` subcommand to the `prodreg` command to remove software from your system. When you uninstall software with 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 Dependencies Between Software Components (`prodreg`)" on page 230 for instructions on how to check software dependencies.

After you uninstall a software component, you can remove the software and all the child components of that software from the Solaris Product Registry by using the `prodreg unregister -r` command.

1. **Become superuser or assume an equivalent role.**
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 9 8/03
                                                          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 previous `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
```

In the previous sample output, no other software depends on the child components of *name* software.

6. **Unregister the software and 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—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 9 8/03
                                                          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.

1. **Become superuser or assume an equivalent role.**
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 9 8/03
                                                        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.**

You can remove the software 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—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 9 8/03
                                                          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` to perform a forced unregister of the damaged component, and then reinstall the component.

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

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 9 8/03
                                                        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 perform a forced unregister of *name* software, then reinstall *name* software.

4. **Unregister only the software component you want to reinstall.**

   # **prodreg unregister -f -u** *UUID*

5. **Reinstall the software component.**

   # **/usr/bin/java -cp /usr/installers/***installer*

   | | |
   |---|---|
   | *installer* | Specifies the name of the installer program for *name* software. |

## Example—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 9 8/03
                                                        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
```

# Adding and Removing Signed Packages (Task Map)

The following task map describes the tasks for adding and removing signed packages.

| Task | Description | For Instructions |
|---|---|---|
| Import a certificate | Import a trusted certificate with the `pkgadm addcert` command. | "How to Import a Trusted Certificate into the Package Keystore (`pkgadm addcert`)" on page 243 |
| (Optional) Display the details of one or more certificates | Display the details of a certificate with the `pkgadm listcert` command. | "How to Display Certificate Information (`pkgadm listcert`)" on page 245 |
| (Optional) Remove a certificate | Remove a certificate with the `pkgadm removecert` command. | "How to Remove a Certificate (`pkgadm removecert`)" on page 246 |
| (Optional) Set up a proxy server | Specify a proxy server if your system is behind a firewall with a proxy. | "How to Set Up a Proxy Server (`pkgadd`)" on page 246 |
| Add a signed package | After the root certificate is imported, you can add a signed package with the `pkgadd` command. | "How to Add a Signed Package (`pkgadd`)" on page 247 |
| (Optional) Remove a signed package | Removing a signed package is identical to removing an unsigned package. | "How to Remove Software Packages (`pkgrm`)" on page 256 |

## ▼ How to Import a Trusted Certificate into the Package Keystore (`pkgadm addcert`)

1. **Become superuser or assume an equivalent role.**
2. **Verify that the Root 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` command, see keytool-Key and Certificate Management Tool. |
| `-storepass` *storepass* | Specifies the password that protects the integrity of the Java keystore. |
| `-list` | By default, prints the MD5 fingerprint of a certificate. |
| `-keystore` *certfile* | Specifies the name and location of the persistent Java 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 /usr/j2se/jre/lib/security/cacerts -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 command output includes the details of the certificate, which the user is asked to verify. |
| `-f` *format* | Specifies the format of the certificates 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. |

For more information, see the `pkgadm` man page.

5. **Remove the temporary file.**

## Example—Importing a Trusted Certificate

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 with 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
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 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...
Trusting certificate <GTE CyberTrust Root>
Type a Keystore protection Password.
Press ENTER for no protection password (not recommended): xxx
For Verification: Type a Keystore protection Password.
Press ENTER for no protection password (not recommended): xxx
Certificate(s) from </tmp/root.crt> are now trusted
# rm /tmp/root.crt
```

## ▼ How to Display Certificate Information (`pkgadm listcert`)

1. **Become superuser or assume an equivalent role.**
2. **Display the contents of the package keystore.**

   ```
   # pkgadm listcert
   ```

## Example—Displaying Certificate Information (`pkgadm listcert`)

The following example shows how to display the details of a locally stored certificate.

```
# pkgadm listcert -P pass: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 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...
```

## ▼ How to Remove a Certificate (`pkgadm removecert`)

1. **Become superuser or assume an equivalent role.**
2. **Remove the trusted certificate from the package keystore.**

   # **pkgadm removecert -n** "*certfile*"

   The `-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 with the `pkgadm listcert` command.

---

### Example—Removing a Certificate (`pkgadm removecert`)

The following example shows how to remove a certificate.

```
# 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 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...
# 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 with the `pkgadd` command.

1. **Become superuser or assume an equivalent role.**
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 a admin 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 admin file with 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 into the Package Keystore (`pkgadm addcert`)" on page 243.

For information about setting up a proxy server, see "How to Set Up a Proxy Server (`pkgadd`)" on page 246.

1. **Become superuser or assume an equivalent role.**
2. **Add a signed package.**

```
# pkgadd -d /pathname/package-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.

## Examples—Adding a Signed Package (pkgadd)

The following example shows how to add a signed package that has already been downloaded.

```
# # 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 With Package Commands (Task Map)

The following task map describes the software management tasks that you can do with the package commands for both signed and unsigned packages.

| Task | Description | For Instructions |
|---|---|---|
| Add a software packages to the local system | You can add software packages to the local system with the pkgadd command. | "How to Add Software Packages (pkgadd)" on page 249 |
| 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 252 |

| Task | Description | For Instructions |
|---|---|---|
| List information about all installed software packages | You can list information about installed packages with the `pkginfo` command. | "How to List Information About All Installed Packages (`pkginfo`)" on page 253 |
| Check the integrity of installed software packages | You can verify the integrity of install software packages with the `pkgchk` command. | "How to Check the Integrity of Installed Software Packages (`pkgchk`)" on page 254 |
| Remove software packages | You can remove unneeded software packages with the `pkgrm` command. | "How to Remove Software Packages (`pkgrm`)" on page 256 |

## ▼ How to Add Software Packages (`pkgadd`)

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

2. **Remove any already installed packages with the same names as the ones you are adding.**

   This step ensures that the system keeps a proper record of software that has been added and removed. There might be times when you 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 214, and for task information, see "How to Remove Software Packages (`pkgrm`)" on page 256.

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 consult during the installation. For details about using an administration file, see "Using an Administration File" on page 215 in the previous chapter. |
   | `-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. |

   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—Adding Software Packages From a Mounted CD

The following example shows how install the `SUNWpl5u` package from a mounted Solaris 9 CD. The example also shows how to verify that the package files were installed properly.

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product SUNWpl5u
    .
    .
    .
Installation of <SUNWpl5u> was successful.
# pkgchk -v SUNWpl5u
/usr
/usr/bin
/usr/bin/perl
/usr/perl5
/usr/perl5/5.00503
.
.
.
```

## Example—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`, and 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.
```

The following 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 215. 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—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 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`)

1. **Become superuser or assume an equivalent role.**
2. **Remove any already spooled packages with the same names as the packages you are adding.**

   For information on removing spooled packages, see "Example—Removing a Spooled Software Package" on page 257.
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* is copied correctly, the `pkginfo` command returns a line of information about the *pkgid*. Otherwise, the `pkginfo` command returns the system prompt.

### *Example—Setting Up a Spool Directory From a Mounted CD*

The following example shows how to transfer the `SUNWman` package from a mounted SPARC Solaris 9 CD to the default spool directory (`/var/spool/pkg`).

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product -s /var/spool/pkg SUNWman
Transferring <SUNWman> package instance
```

### *Example—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 to do 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`, and 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—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`)

List information about installed packages with the `pkginfo` command.

```
$ pkginfo
```

## Example—Listing All Packages Installed

The following example shows the `pkginfo` command to list all packages installed on a local system, whether that system is a standalone or 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—Displaying Detailed Information About Software 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:  leo20031111173915
  INSTDATE:  Apr 02 2004 11:29
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:      114 installed pathnames
                  36 shared pathnames
                  40 directories
                  57 executables
                21725 blocks used (approx)
```

# ▼ How to Check the Integrity of Installed Software Packages (`pkgchk`)

1. **Become superuser or assume an equivalent role.**
2. **Check the status of an installed package.**

   ```
   # pkgchk -a| -c -v pkgid ...
   # pkgchk -d spooldir pkgid ...
   ```

| | |
|---|---|
| `-a` | Specifies to audit only the file attributes, that is, the permissions, rather than the file attributes and 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—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—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—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>
```

**Note –** The checks made on a spooled package are limited because not all information can be audited until a package is installed.

## Removing Software Packages

Use the associated tool that you used to add or install a software package to remove or uninstall a software package. For example, if you used the Web Start installer to install software, use the Web Start uninstaller to uninstall software.

**Caution –** Do not use the `rm` command to remove software packages.

### ▼ How to Remove Software Packages (`pkgrm`)

1. **Become superuser or assume an equivalent role.**
2. **Remove an installed package.**

   # **pkgrm** *pkgid* ...

   *pkgid* identifies the name of one or more packages, separated by spaces, to be removed. If omitted, `pkgrm` removes all available packages.

### Example—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—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>
```

# Adding and Removing Software Packages With Admintool (Task Map)

The following task map describes the software management tasks that you can do with Admintool.

| Task | Description | For Instructions |
|---|---|---|
| Add software packages with Admintool | You can view or add software packages. | "How to Add Software Packages With Admintool" on page 257 |
| Remove software packages with Admintool | You can view or remove software packages. | "How to Remove Software Packages With Admintool" on page 259 |

The Solaris operating system includes Admintool, which is a graphical user interface for performing several administration tasks, including adding and removing software packages. Specifically, you can use Admintool to do the following:

- Add software packages to a local system
- Remove software packages from a local system
- View software already installed on the local system
- Customize software packages to be installed
- Specify an alternate installation directory for a software package

## ▼ How to Add Software Packages With Admintool

1. **Become superuser.**

   Unless you are a member of the `sysadmin` group (group 14), you must become superuser or assume an equivalent role to add or remove software packages with Admintool.

2. **Load a Solaris 9 Software CD or DVD into the drive.**
   Volume Manager automatically mounts the CD.

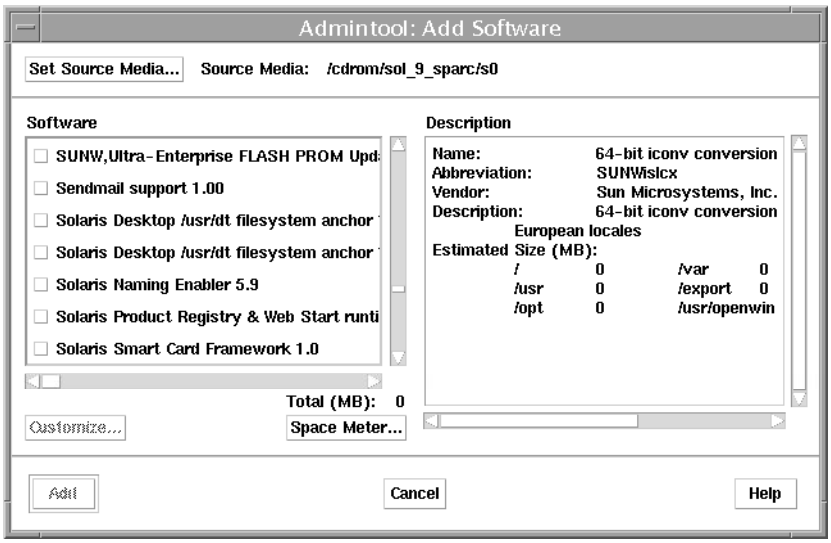
3. **Start Admintool.**

   ```
   # admintool &
   ```

   The Users window is displayed.

4. **Choose Software from the Browse menu.**
   The Software window is displayed.

5. **Choose Add from the Edit menu.**
   The Set Source Media window might appear. If so, specify the path to the installation media and click OK. The default path is a mounted Solaris CD.
   The Add Software window is displayed.




6. **Select the software you want to install on the local system.**
   In the Software portion of the window, click the check boxes that correspond to the software you want to install.

7. **Click Add.**
   A Command Tool window appears for each package being installed, displaying the installation output.
   The Software window is refreshed to display the packages just added.

## ▼ How to Remove Software Packages With Admintool

1. **Become superuser.**

   Unless you are a member of the `sysadmin` group (group 14), you must become superuser or assume an equivalent role to add or remove software packages with Admintool.

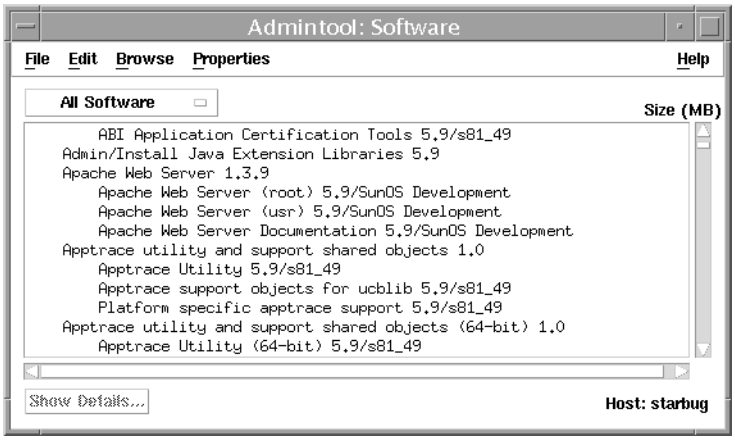
2. **Start Admintool.**

   ```
   # admintool &
   ```

   The Users window is displayed.

3. **Choose Software from the Browse menu.**

   The Software window is displayed.




4. **Select the software you want to delete from the local system.**

5. **Choose Delete from the Edit menu.**

   A warning pop-up window is displayed to confirm whether you really want to delete the software.

6. **Click Delete to confirm that you want to delete the software.**

   For each package that is being deleted, a Command Tool window is displayed that asks for confirmation, again, before deleting the software. Type `y`, `n`, or `q`. If you choose to delete the software, the output from the removal process is displayed.

CHAPTER 15

# Managing Solaris Patches (Overview)

Patch management involves listing or adding Solaris patches from a system running the Solaris release. Patch management might also involve removing unwanted or faulty patches. Removing patches is also called *backing out* patches.

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



For step-by-step instructions on adding a patch to your system, see "Managing Patches in the Solaris Environment (Road Map)" on page 267.

For information on adding patches to diskless client systems, see "Patching Diskless Client OS Services" on page 120.

---

**Note –** Overview information about using the `smpatch` command with PatchPro has been removed from this guide. For information about using the `smpatch` command with PatchPro, see *Signed Patches Administration Guide for PatchPro 2.2*.

---

## What Is a Patch?

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. The existing software is derived from a specified *package* format, which conforms to the Application Binary Interface. For details about packages, see Chapter 13.

You can manage patches on your system with the `patchadd` command. For step-by-step instructions on adding an unsigned patch to your system, see "Managing Unsigned Solaris Patches (Task Map)" on page 273.

## What Is a Signed Patch?

A *signed* patch is a patch with a digital signature. A patch with a valid digital signature ensures that the patch has not been modified after the signature was applied to the patch. Using signed patches is a more secure method of downloading or adding patches because the patches include a digital signature that can be verified before the patch is added to your system.

Patches that are available for the Solaris 2.6, 7, 8, and 9 releases include a digital signature. Patches without a digital signature, or *unsigned patches*, are also available, but eventually, all patches will be *signed patches*. A valid digital signature ensures that the patch has not been modified since the signature was applied.

Signed patches are stored in Java archive format (JAR) files and are available from the SunSolve Online$^{SM}$ web site.

In previous Solaris releases, you could use the `smpatch` command with PatchPro to add signed patches to your system. For step-by-step instructions on using the `smpatch` command, see "Managing Signed Patches by Using Solaris Patch Management Tools (Tasks)" in *Signed Patches Administration Guide for PatchPro 2.2*.

In this Solaris release, you can use the `patchadd` command to add signed patches to your system. For step-by-step instructions on using the `patchadd` command, see "Adding Signed Patches With `patchadd` Command (Task Map)" on page 269.

For additional overview information about signed patches, see "Signed Packages and Patches" on page 207.

# Accessing Solaris Patches

All Sun customers can access patches through the SunSolve Online$^{SM}$ web site. The following table describes the various ways to access Solaris patches.

**TABLE 15–1** Ways to Access Solaris Patches

| Customer Type | Description |
|---|---|
| SunSpectrum contract customer | You have access to the SunSolve database of patches and patch information. They are available from the SunSolve Online web site or by using anonymous `ftp`.<br><br>These patches are updated nightly. |
| Not a SunSpectrum contract customer | You have access to a general set of security patches and other recommended patches. These patches are available through SunSolve Online. |

You can access Solaris patches from a web site or by using anonymous `ftp`.

To access patches from a web site, you need a system that is:

- Connected to the Internet
- Capable of running a web browser such as the Netscape™ software.

To access patches by anonymous `ftp`, you need a system that is:

- Connected to the Internet
- Capable of running the `ftp` program

Access patches from the SunSolve Online℠ web site by using the following URL:

```
http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access
```

You can install either a patch cluster of recommended patches or individual patches that are freely available. Patch reports are also available.

# Solaris Patch Numbering

Patches are identified by unique alphanumeric strings, with the patch base code first, a hyphen, and a number that represents the patch revision number. For example, patch 108528-10 is a *patch ID* for the SunOS 5.8 kernel update patch.

# Tools for Managing Solaris Patches

The following table summarizes Solaris patch management features.

| Feature | `patchadd`/`patchrm` Commands | Solaris 2.6, 7, and 8 Patch Management Tools | Solaris 9 Patch Management Tools | PatchPro Interactive or PatchPro Expert |
|---|---|---|---|---|
| How do I get this tool? | Bundled in Solaris release (`SUNWswmt`) | Must download tool from http://www.sun.com/PatchPro | Must download tool from http://www.sun.com/PatchPro | Run tool from http://www.sun.com/PatchPro |
| Solaris release availability | Solaris 2.6, 7, 8, and 9 releases | Solaris 2.6, 7, and 8 | Solaris 9 | Solaris 2.6, 7, 8, and 9 |
| Adds signed patches? | 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 |
| Adds unsigned patches? | Yes | No | Yes, but the patches must be unzipped first | Yes, but the patches must be unzipped first |
| GUI available? | No | No | Yes | No |
| Analyzes system for required patches and downloads signed or unsigned patches | No | Yes, both signed and unsigned patches | Yes, both signed and unsigned patches | Yes, unsigned patches only |
| Local and remote system patch support | Local | Local | Local and Remote | No |
| RBAC support? | Yes | No | Yes | No |

Detailed information about how to install and back out a patch is provided in the `patchadd`(1M) and `patchrm`(1M) man pages. Each patch also contains a `README` file that contains information about the patch.

# Selecting the Best Method for Adding Signed Patches

After you have installed a patch management tool, you can use several different methods of downloading or adding a signed patch or patches to your system. Use the following table to determine which method is best for your needs.

| Command or Tool | Description | For More Information |
|---|---|---|
| `patchadd` | Starting in the Solaris 9 12/03 release – Use this command to add signed patches to your system after your have set up your package keystore. | `patchadd`(1M) |
| `smpatch update` | **Solaris 2.6, Solaris 7, Solaris 8, and at least Solaris 9 4/03** – Use this command to identify the recommended patches and automatically download and apply the patches to your system. Notice that this command will not apply a patch that has the interactive property set. | `smpatch`(1M) |
| `smpatch analyze` | Use this command to identify required patches and display a list of required patch IDs for your system. Then, you could use the `smpatch download` and `smpatch add` commands to download and add the patches to your system. | `smpatch(1M)` |
| `smpatch download` and `smpatch add` | Use these commands to download and apply one or more patches to your system. These commands also download and apply any prerequisite patches. | "Downloading and Applying Signed Patches to a Solaris System (Task Map)" in *Signed Patches Administration Guide for PatchPro 2.2* |
| `ftp` and `smpatch add` | Use the `ftp` command to transfer a patch or patches to your system. Then, use the `smpatch add` command to add the patch or patches to your system. | "Downloading and Applying Signed Patches to a Solaris System (Task Map)" in *Signed Patches Administration Guide for PatchPro 2.2* |
| Solaris Management Console Patches Tool | **For Solaris 9 systems only –** Use this tool when you want the convenience of a GUI tool to manage signed patches. | Solaris Management Console online help |

CHAPTER 16

# Managing Solaris Patches (Tasks)

This chapter provides step-by-step instructions for managing patches in the Solaris environment.

This is a list of the task maps in this chapter.



For overview information about managing patches in the Solaris environment, see Chapter 15.

**Note –** Step-by-step instructions for using the `smpatch` command with PatchPro has been removed from this guide. For information about using the `smpatch` command with PatchPro, see *Signed Patches Administration Guide for PatchPro 2.2*.

## Managing Patches in the Solaris Environment (Road Map)

Use this map to identify all the tasks for managing patches in the Solaris environment. Each task points to a series of additional tasks such as managing signed or unsigned patches.

| Task | Description | For Instructions |
|---|---|---|
| Determine if adding signed or unsigned patches | Determine whether adding signed or unsigned patches is best for your environment. | "Selecting Signed or Unsigned Patches for Your Environment" on page 268 |
| Add a signed or unsigned patch to your system | You can add signed patches with either of the following commands: | |
| | Use the `patchadd` command starting in the Solaris 9 12/03 release. | "Adding Signed Patches With `patchadd` Command (Task Map)" on page 269 |
| | Use the `smpatch` command in the Solaris 2.6, 7, 8, or 9 releases. | "Downloading and Applying Signed Patches to a Solaris System (Task Map)" in *Signed Patches Administration Guide for PatchPro 2.2* |
| | Add an unsigned patch to your system. | "Managing Unsigned Solaris Patches (Task Map)" on page 273 |

# Selecting Signed or Unsigned Patches for Your Environment

The key factor in determining when to add signed or unsigned patches is whether or not the secure download of patches is important in your environment. If the secure download of patches is important in your environment, then add signed patches to your system.

# Adding Signed Patches With `patchadd` Command (Task Map)

| Task | Description | For Instructions |
|---|---|---|
| 1. Set up the package keystore | Import Sun's Root CA certificate into your package keystore. | "How to Import a Trusted Certificate into Your Package Keystore (`pkgadm addcert`)" on page 269 |
| (Optional) Set up a proxy server | Specify a proxy server if your system is behind a firewall with a proxy. | "How to Set Up a Proxy Server (`patchadd`)" on page 271 |
| 2. Download and add the signed patch | Select one of the following to download and add the signed patch to your system with the `patchadd` command. | |
| | You can manually download and add a signed Solaris patch. | "How to Manually Download and Add a Signed Solaris Patch (`patchadd`)" on page 271 |
| | You can automatically download and add a signed Solaris patch. | "How to Automatically Download and Add a Signed Solaris Patch (`patchadd`)" on page 272 |

## How to Import a Trusted Certificate into Your Package Keystore (`pkgadm addcert`)

To add signed patches to your system with the `patchadd` command, you will need to add Sun's Root CA certificate, at the very least, to verify the signature on your signed patch. You can import this certificate from the Java keystore into the package keystore.

1. **Become superuser or assume an equivalent role.**
2. **Export the Root CA certificate from the Java keystore into 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 to hold the exported certificate. |

3. **Import the Root CA certificate into the package keystore from the temporary file.**

   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 1996 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 details of the certificate, which the user is asked to verify. |
| `-f` *format* | Specifies the format of the certificates 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.**

   For example:

```
# pkgadm listcert -P pass: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 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
```

5. **Remove the temporary file.**

   For example:

   ```
   # rm /tmp/root.crt
   ```

## ▼ How to Set Up a Proxy Server (`patchadd`)

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 with the `patchadd` command.

1. **Become superuser or assume an equivalent role.**
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 `patchadd` command line.**

      For example:

      ```
      # patchadd -x mycache.domain:8080 -M http://www.sun.com/solaris/patches/latest 101223-02
      102323-02
      ```

## ▼ How to Manually Download and Add a Signed Solaris Patch (`patchadd`)

You can use this procedure when you want to manually download the signed Solaris patch, and then add the signed Solaris patch in a separate step.

This procedure assumes that you have set up the package keystore.

1. **(Optional) Log in to the system where the patch will be applied.**

   Or, you can download the patch and use the `ftp` command to copy the patch to the target system.

2. **Open a web browser and go to the SunSolve Online Web site:**

```
http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access
```

3. **Determine if you are going to download a specific patch or patch cluster. Then select one of the following:**

   a. **Type the patch number (*patch-ID*) in the "Find Patch" search field. Then, click on Find Patch.**

      Entering *patch-ID* downloads the latest patch revision.

      If this patch is freely available, the patch `README` is displayed. If this patch is not freely available, an `ACCESS DENIED` message is displayed.

      There are different patch numbers for SPARC and x86 systems, which are listed in the displayed patch `README`. Make sure you install the patch that matches your system architecture.

   b. **Click on a recommended patch cluster based on the Solaris release running on the system to be patched.**

4. **Click the Download Signed Patch (*n* bytes) HTTPS or FTP button.**

   After the signed patch or patches are downloaded successfully, close the web browser.

5. **Change to the directory that contains the downloaded patch package, if necessary.**

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

7. **Add the signed patch.**

   For example:

```
# patchadd /tmp/114861-01.jar
```

## ▼ How to Automatically Download and Add a Signed Solaris Patch (`patchadd`)

You can use this procedure when you want to automatically download and add a signed Solaris patch in one step.

This procedure assumes that you have set up the package keystore.

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

2. **Identify the HTTP URL for the patch you want to download.**

   a. **Open a web browser and go to the SunSolve Online web site:**

      `http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access`

   b. **Enter the patch number you wish to download.**

For example:

```
114861-01 Find Patch
```

c. **Place your mouse over the HTTPS link at the top of the patch page in hover mode.**

The URL for the patch is displayed in the browser status line at the bottom of the screen.

3. **Download and add the signed patch or patches from the SunSolve Online web site.**

For example:

```
# patchadd "http://sunsolve.central.sun.com/cgi/patchDownload.pl?target=
114684&method=hs"
.
.
.
Downloading patch from ...
+ dwnld_file http://sunsolve.central.sun.com/cgi/patchDownload.pl?target=
114684&method=hs /tmp/patchadd-dwnld /var/sadm/security  console patchadd
...........20%...........40%...........60%...........80%...........100%
## Downloading...
## Download Complete
.
.
.
Enter keystore password: xxx
.
.
.
```

# Managing Unsigned Solaris Patches (Task Map)

| Task | Description | For Instructions |
|---|---|---|
| 1. (Optional) Display information about unsigned patches | Display information about unsigned patches already installed on your system. | "How to Display Information About Solaris Patches" on page 274 |
| 2. Download an unsigned patch | Download an unsigned patch to your system. | "How to Download an Unsigned Solaris Patch" on page 276 |

| Task | Description | For Instructions |
|---|---|---|
| 3. Add an unsigned patch | Add an unsigned patch to your system. | "How to Add a Unsigned Solaris Patch (`patchadd`)" on page 276 |
| 4. (Optional) Remove an unsigned patch | If necessary, remove an unsigned patch from your system. | "How to Remove an Unsigned Solaris Patch" on page 278 |

# Displaying Information About Unsigned Solaris Patches

Before installing patches, you might want to know more about patches that have previously been installed. The following table describes commands that provide useful information about patches that are already installed on a system.

**TABLE 16–1** Commands for Solaris Patch Management

| Command | Description |
|---|---|
| `patchadd -p`, `showrev -p` | Shows all patches that have been applied to a system. |
| `pkgparam` *pkgid* `PATCHLIST` | Shows all patches that have been applied to the package identified by *pkgid*, the name of the package. For example, `SUNWadmap`. |
| `patchadd -S` *Solaris-OS* `-p` | Shows all the `/usr` patches installed on an OS server. |

# How to Display Information About Solaris Patches

Use the `patchadd -p` command to display information about patches installed on your system.

```
$ patchadd -p
```

Use the following command to verify whether a specific patch is installed on your system. For example:

```
$ patchadd -p | grep 111879
```

# Adding an Unsigned Solaris Patch

You can use the following tools to add unsigned patches to servers or standalone systems:

- `patchadd`

- `smpatch`
- Solaris Management Console's Patch Manager

If you need to add a patch to a diskless client system, see "Patching Diskless Client OS Services" on page 120.

When you add a patch, the patch tools call the `pkgadd` command to install the patch packages from the patch directory to a local system's disk. 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 information about patches obsoleted by the patch being installed, other patches required by this patch, and patches incompatible with this patch

During patch installation, the `patchadd` command keeps a log of the patch installation in the `/var/sadm/patch/`*patch-ID*`/log` file for current Solaris versions.

The `patchadd` command will not install a patch under the following conditions:

- The package is not fully installed on the host.
- The patch packages architecture differs from the system's architecture.
- The patch packages version does not match the installed package's version.
- A patch with the same base code and a higher version number is already installed.
- The patch is incompatible with another, already installed patch. Each installed patch keeps this information in its `pkginfo` file.
- The patch being installed requires another patch that is not installed.

When you add or remove patches with the `smpatch` command, the command prompts you for authentication information if you do not specify the information in the command line.

You can specify authentication information to the `smpatch` command using the following syntax:

```
# smpatch add -p mypassword -u root -- -i patch-ID-revision
```

The `smpatch` subcommands, such as `add` or `remove`, are separated from the authentication options and arguments by --.

Or, you can let the `smpatch` command prompt you for the authentication information.

```
# /usr/sadm/bin/smpatch add -i patch-ID-revision
Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.patchmgr.cli.PatchMgrCli from holoship
Login to holoship as user root was successful.
Download of com.sun.admin.patchmgr.cli.PatchMgrCli from holoship was
successful.
```

## ▼ How to Download an Unsigned Solaris Patch

1. **(Optional) Log in to the system where the patch will be applied.**

   Or, you can download the patch and use the `ftp` command to copy the patch to the target system.

2. **Open a web browser and go to the SunSolve Online web site:**

   `http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access`

3. **Determine if you are going to download a specific patch or patch cluster. Then, select one of the following:**

   a. **Type the patch number (*patch-ID*) in the "Find Patch" search field. Then, click on Find Patch.**

      Entering *patch-ID* downloads the latest patch revision.

      If this patch is freely available, the patch `README` is displayed. If this patch is not freely available, an `ACCESS DENIED` message is displayed.

      There are different patch numbers for SPARC and x86 systems, which are listed in the displayed patch `README`. Make sure you install the patch that matches your system architecture.

   b. **Click on a recommended patch cluster based on the Solaris release running on the system to be patched.**

4. **Click the Download Patch (*n* bytes) HTTP or FTP button.**

   After the patch or patches are downloaded successfully, close the web browser.

5. **Change to the directory that contains the downloaded patch package, if necessary.**

6. **Unzip the patch package.**

   `% unzip` *patch-ID-revision*

## ▼ How to Add a Unsigned Solaris Patch (`patchadd`)

This procedure assumes that the patch has already been downloaded and unzipped.

1. **Become superuser.**

2. **Add the patch or patches.**

   `# patchadd` *patch-ID-revision*

3. **Verify that the patch was added successfully.**

   `# patchadd -p | grep` *patch-ID-revision*

## Example—Adding an Unsigned Solaris Patch

In the following example, the Solaris 8 patch, 111879–01, is added to the system with the `patchadd` command. The patch had already been downloaded to the system previously and unzipped.

```
# patchadd /export/Sol8patch/111879-01

Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

Patch number 111879-01 has been successfully installed.
See /var/sadm/patch/111879-01/log for details

Patch packages installed:
  SUNWwsr
# patchadd -p | grep 111879-01
Patch: 111879-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWwsr
```

## ▼ How to Add a Unsigned Solaris Patch (`smpatch`)

Use this procedure on a system that runs the Solaris 9 release.

This procedure assumes that the patch is has already been downloaded to the `/var/sadm/spool` directory and is unzipped.

1. **Become superuser.**
2. **Add the patch or patches.**

   # **/usr/sadm/bin/smpatch add** *patch-ID-revision*

   For example:

   ```
   # /usr/sadm/bin/smpatch add -i 115028-01
   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.patchmgr.cli.PatchMgrCli from holoship
   Login to holoship as user root was successful.
   Download of com.sun.admin.patchmgr.cli.PatchMgrCli from holoship was
   successful.

   Patch 115028-01, or a patch required by patch 115028-01, requires a
   system reboot after installation. Perform a reconfiguration reboot
   immediately after the installation.

           On machine holoship ...
                   Installing patch 115028-01
   ```

3. **Check `smpatch` messages for instructions to reboot the system.**

Reboot the system if you are instructed to reboot.

```
# init 6
```

4. **Verify that the patch was added successfully.**

```
# patchadd -p | grep patch-ID-revision
```

## Removing an Unsigned Solaris Patch

When you back out a patch, the patch tools restore all files modified by that patch, unless any of the following are true:

- The patch was installed with the `patchadd -d` option, which instructs `patchadd` to not save copies of files being updated or replaced.
- The patch has been obsoleted by a later patch.
- The patch is required by another patch.

The patch tools call the `pkgadd` command to restore packages that were saved from the initial patch installation.

During the patch removal process, the `patchrm` command keeps a log of the back out process in `/tmp/backoutlog.`*process_id*. This log file is removed if the patch backs out successfully.

## ▼ How to Remove an Unsigned Solaris Patch

You can use the `smpatch` command, the `patchrm` command, or Solaris Management Console's Patch Manager if you need to remove an unsigned Solaris patch.

1. **Become superuser.**
2. **Remove the patch.**

```
# patchrm patch-ID-revision
```

Or,

```
# /usr/sadm/bin/smpatch remove patch-ID-revision
```

3. **Verify that the patch was removed.**

```
# patchadd -p | grep patch-ID-revision
```

## Examples—Removing an Unsigned Solaris Patch

The following example shows how to remove the Solaris 8 patch, 111879–01 with the `patchrm` command.

```
# patchrm 111879-01

Checking installed patches...

Backing out patch 111879-01...

Patch 111874-02 has been backed out.

# showrev -p | grep 111879-01
#
```

The following example shows how to remove a Solaris 9 patch with the `smpatch remove` command.

```
# /usr/sadm/bin/smpatch remove -i 115028-01
Authenticating as user: root
.
.
.
```

# Index

## C



## D

## E



## F



## G

## H



## I



## J



## K



## L



## M

## R



## S

## T



## U

## V



## W



## X