Solaris Advanced Installation Guide

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043-1100
U.S.A.

Part No: 802-5740
August 1997

Contents

About This Book

This book describes how to install the Solaris™ operating environment on both networked and non-networked SPARC and x86 based systems, and it focuses on the Solaris advanced installation features that should be used for large, enterprise sites. The advanced installation features include setting up network installations and automating installations by using the custom JumpStart technology. If you are installing a single system from a local CD-ROM, you should use the *Installation Instructions for Solaris 2.6*.

Note - This book does not include instructions for setting up system hardware or other peripherals.

Note - The term “x86” refers to the Intel 8086 family of microprocessor chips, including the Pentium and Pentium Pro processors and compatible microprocessor chips made by AMD and Cyrix. In this document the term “x86” refers to the overall platform architecture, whereas “*Intel Platform Edition*” appears in the product name.

Who Should Use This Book

This book is for anyone who wants to reduce the time and effort involved in installing the Solaris operating environment on a lot of systems. To use this book, you should have 1-2 years of UNIX system administration experience and preferably a Computer Science B.S. degree or equivalent knowledge.

Related Information

Table P–1 lists related information that you may need when installing the Solaris software.

TABLE P–1 Related Information

Platform	Information	Description
All	*Solstice AdminSuite 2.3 Administration Guide*	Describes applications such as Solstice™ Host Manager, which you can use if you're setting up network installations.
	System Administration Guide	Describes how to back up system files.
x86	*Solaris 2.6 (Intel Platform Edition) Release Notes*	Describes any bugs, known problems, software being discontinued, and patches related to the Solaris 2.6 release.
	Information Library for Solaris 2.6 (Intel Platform Edition)	Contains device configuration and supported hardware information. It also contains Solaris installation instructions for desktop systems.
SPARC	*Solaris 2.6 (SPARC Platform Edition) Release Notes*	Describes any bugs, known problems, software being discontinued, and patches related to the Solaris 2.6 release.
	Information Library for Solaris 2.6 (SPARC Platform Edition)	Contains supported hardware information. It also contains Solaris installation instructions for desktop systems.
	Solaris 1.x to 2.x Transition Guide	Describes transition issues including backing up Solaris 1.x (SunOS 4.x) files before installing Solaris software, and restoring files after Solaris software is installed.

Revision History

TABLE P–2 Revision History

Revision	Date	Comments
Solaris 2.6	August 1997	**New: Solaris Web Start Installation Method** Solaris Web Start is a browser-based utility that guides users through the selection and installation of both Solaris software and co-packaged application software. Its graphical user interface also facilitates file system configuration. Use of the utility is optional: users can exit at any point and proceed with one of the more traditional Solaris installation methods.
		New: Upgrade With Disk Space Reallocation The upgrade option in the Solaris Interactive Installation program now provides the ability to reallocate disk space if the current file systems don't have enough space for the upgrade. By default, an auto-layout feature tries to determine how to reallocate the disk space so the upgrade can succeed. If auto-layout can't determine how to reallocate disk space, you must specify which file systems can be moved or changed and run auto-layout again. If you are using the custom JumpStart method, two new upgrade profile keywords, `backup_media` and `layout_constraint`, can be used to reallocate disk space.
		Change: Testing Upgrade Profiles In previous Solaris releases, you could only test profiles that used the initial option. The Solaris 2.6 release now enables the `pfinstall` command to test profiles that use the upgrade option, so you can see if a profile will do what is wanted before using it to upgrade a system. This is especially useful when you are creating upgrade profiles that reallocate disk space. To test an upgrade profile, you must run the `pfinstall –D` command on the system that you're going to upgrade (against the system's disk configuration and its currently installed software). You cannot test an upgrade profile using a disk configuration file. Also, the procedure to test an initial profile has changed in the Solaris 2.6 release.
		New: 8-bit English Locale When installing the base (English) Solaris 2.6 CD, you'll be prompted to select an English locale, because a new, 8–bit English local (`en_US`) has been added to the Solaris 2.6 release. If you don't want to be prompted for the locale, you must preconfigure the locale information.

TABLE P–2 Revision History *(continued)*

Revision	Date	Comments
		New: Preconfiguring System Configuration Information with `sysidcfg(4)` In previous releases, the first part of the Solaris installation program tried to obtain system configuration information about a system (such as the system's peripheral devices, host name, IP address, name service) from the name service databases. The installation program prompted you if it couldn't find the information. If you didn't want to be prompted, you had to preconfigure the system configuration information in the name service. Using the Solaris 2.6 `sysidcfg(4)` file, you can now preconfigure system configuration information through a set of keywords. You can choose to provide one or more of the keywords to preconfigure varying levels of system configuration information.
		New: Changing a System's Boot Device During the Installation The Solaris 2.6 release enables you to change a system's boot device during the installation. A system's boot device is the disk slice where the root file system is installed and, consequently, where the installed system will boot from. Also, the installation program can now update the system's EEPROM if you change the new boot device, so the system can automatically boot from it (SPARC systems only). In previous releases, changing the system's boot device during an installation meant that you had to manually change the system's EEPROM so it could automatically boot from the new boot device. This new feature is provided in the Solaris Interactive Installation program and by the new custom JumpStart `boot_device` profile keyword.
		Change: Solaris CD-ROM Layout The Solaris CD layout has been changed in the Solaris 2.6 release. Slice 0 has been reorganized to make it more intuitive and extensible. It contains only control files and the `Solaris_2.6` directory at the top level. The control file are the same control files found on previous Solaris CDs (`.cdtoc`, `.slicemapfile`, and `.install_config`). The `Solaris_2.6` directory contains all the tools, software, and configuration necessary to install, at a minimum, the Solaris 2.6 software product.

TABLE P–2 Revision History *(continued)*

Revision	Date	Comments
		New: Disk Space Planning Section This document now provides an appendix to help you plan your disk space. The appendix includes the list of packages contained in each software group.
		Change (x86): Booting Software The Solaris boot diskette has been replaced by the Solaris Device Configuration Assistant diskette. The Configuration Assistant program is part of the new booting system for the Solaris (Intel Platform Edition) software, and it determines which hardware devices are in the system, accounts for the resources each device uses, and enables you to choose which device to boot from. The Configuration Assistant must be re-run any time the hardware configuration is changed.
		Change (x86): `kdmconfig` **Program Automatically Configures Peripherals** The `kdmconfig` program has been updated to automatically configure the mouse, graphics adapter, and monitor on an x86 system. If an `OWconfig` file already exists on the system, `kdmconfig` will extract any usable information from it. In addition, `kdmconfig` will also retrieve information left in the `devinfo` tree by the Configuration Assistant program, and use that information to automatically identify devices. Any attribute values, such as manufacturer or model information, returned by probes for a given device (unless it is "Unknown") take precedence over the value received for the same attribute from the `OWconfig` file.
		New: `root_device` **Profile Keyword** When creating upgrade profiles, you must now specify `root_device` if more than one root file system can be upgraded on a system. The `root_device` profile designates the root file system (and the file systems mounted by its `/etc/vfstab` file) to be upgraded. For initial profiles, `root_device` designates the system's root disk and sets the `rootdevice` variable. How the system's root disk is determined during a custom JumpStart installation has also changed in the Solaris 2.6 release.
		Change (SPARC): Hardware Support Dropped The SPARCserver 6xx systems are no longer supported.

TABLE P–2 Revision History *(continued)*

Revision	Date	Comments
Solaris 2.5	November 1995	**Change: Service Setup For Clients During Installation** The Solaris installation program (interactive and custom JumpStart) no longer sets up services for clients. You can specify the number of clients and allocate space for them during installation, but you must use Solstice Host Manager to complete client set up after Solaris software is installed.
		New: bootparams Keyword/Value A new bootparams keyword/value forces `sysidtool` to attempt to configure a specified names service (overriding the default NIS+), thus enabling clients to be set up for off-subnet servers. See `bootparams(4)`. The bootparams keyword/value has the following syntax: `ns=[`*server*`]:`*nameservice*`[(`*netmask*`)]` This addition affects the `/etc/bootparams` file, Solstice Host Manger, and `add_install_client` script (where `-n` <*ns_string*> is the string to put in the bootparams table.).
		Change: Location of Diskless Client Booting Information Information on how to boot diskless clients has been moved to the *System Administration Guide*
		Change: Solstice Host Manager Replaces `add_install_client` The Solstice Host Manager now supports remote installations, and it is the recommended tool for setting up network install services (instead of `add_install_client` command). Solstice Host Manager can also now be used to set up custom JumpStart installations.

TABLE P–2 Revision History *(continued)*

Revision	Date	Comments
		Change (SPARC): Hardware Support Dropped The sun4 and sun4e hardware is no longer supported.
		Change: Underlying Software The `/usr/kvm` directory is replaced by the `/usr/platform` directory. Servers no longer have to export `/usr/kvm` for each supported platform, and clients do not have to mount the exported `/usr/kvm` directory appropriate for their platform. With `/usr/platform`, the same exported `/usr` file system can support all platforms. The terms *kernel architecture* and *architecture*, have been replaced by the terms *platform name* (for example SUNW,S240), and *platform group* (for example, sun4m).

Ordering Sun Documents

The SunDocs^{SM} program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals using this program.

For a list of documents and how to order them, see the catalog section of SunExpress^{TM} Internet site at `http://www.sun.com/sunexpress`.

CHAPTER 1

Overview of Installing Solaris

Successfully installing Solaris software requires a clear sense of what you're doing and why, especially if you have to install a large number of systems. This chapter provides all the information you need to determine the best way to install the Solaris software at your site, including installing systems over the network and automating the installation process.

- "Maybe You Don't Need to Install" on page 1
- "Installing Standalone Systems and Servers" on page 2
- "Methods for Installing Solaris Software" on page 2

Note - If you want to install a desktop system from a local CD-ROM, you should use the *Installation Instructions for Solaris 2.6*.

Maybe You Don't Need to Install

If you are managing a large site of desktop systems running Solaris software, you may not need this book. Instead of installing the Solaris software on every system, you may want to turn your systems into Solstice™ AutoClient™ systems or diskless clients. AutoClient systems and diskless clients do not have the Solaris software installed on their local disks; instead, the Solaris software is provided by an OS server.

AutoClient systems and diskless clients reduce the burden and cost of system administration by eliminating the work you do to configure and maintain your desktop systems, which includes installing the Solaris software. These systems help you centralize your administration. However, even if you are going to set up

AutoClient systems or diskless clients, you still need to use this book to install an OS server.

For more information about using these types of systems, refer to the *Solstice AdminSuite 2.3 Administration Guide*.

Installing Standalone Systems and Servers

There are two types of systems on which you must install the Solaris software:

- **Server -** A system that provides services and/or file systems, such as home directories or mail files, for other systems on the network. An *OS server* is a server that provides the Solaris software for other systems on the network. For diskless clients, OS servers provide `/usr`, root (`/`), and `swap` file systems. For AutoClient systems, an OS server provides all system software except the individual root (`/`) and `/usr` file systems required for local swapping and caching.
- **Standalone system -** A system that has all of its Solaris software on its local disk and does not require services from an OS server. Both networked and non-networked systems can be standalone systems in the Solaris operating environment.

Methods for Installing Solaris Software

There are four methods for installing Solaris software:

- **Interactive (Solaris Interactive Installation program) –** The Solaris Interactive Installation program guides you step-by-step in installing the Solaris software. The Solaris Interactive Installation program does not enable you to install all the software (Solaris software and co-packaged software) in your product box at once; it only installs the Solaris software. After you install the Solaris software, you have to install the other co-packaged software by using the co-packaged installation programs.
- **Interactive (Solaris Web Start) –** Solaris Web Start provides a web browser user interface that enables you to install all the software (Solaris software and co-packaged software) in your product box at once. You can install all the software with a default option, or you can use a customize option to install only the software you want.
- **Custom JumpStart (formerly called auto-install) –** This method enables you to automatically and identically install groups of systems. It requires up-front work

before the systems can be installed, but it's the most cost-effective way to automatically install Solaris software for large enterprise sites. See Chapter 8 for more information.

- **JumpStart –** This method enables you to automatically install the Solaris software on a new SPARC-based system just by inserting the Solaris CD into the system and powering on the system. The software that gets installed is specified by a default profile that is picked based on the system's model and the size of its disks; you don't have a choice of the software that gets installed.

  All new SPARC-based systems have the JumpStart software (a preinstalled boot image) pre-installed on its boot disk, which is required to use this method on a system. You can install the JumpStart software on existing systems with the `re-preinstall` command.

Note - If you power on a new system and you've set it up to use the custom JumpStart installation method, the new system will perform a custom JumpStart installation (by using the custom profile) instead of a JumpStart intallation (by using the default profile). Basically, when installing new systems with the custom JumpStart installation method, the preinstalled JumpStart software on the new system enables you to power the system on to start the installation instead of having to specify a boot command.

Installing Over the Network

Because the Solaris software is distributed on a CD, a system has to have access to a CD-ROM drive to install it. However, if you have a large number of systems that don't have a local CD-ROM drive or if you don't want to insert the Solaris CD into every system's CD-ROM drive, you can set upthe systems to install from a remote Solaris CD image. The remote Solaris CD image must be provided by an install server, which has either the Solaris CD copied to its hard disk or the Solaris CD mounted from its CD-ROM drive.

You can use all of the installation methods when installing a system over the network. However, installing systems over the network with the custom JumpStart method is a good way to centralize and automate the installation process for a large enterprise site.

Requirements for a Totally Automated Installation

To set up your site to install Solaris software on systems over the network with no user intervention, you must:

- Preconfigure network information for the systems, such as the date, time, geographic region, site subnet mask, and language. This eliminates many prompts that are otherwise necessary to identify the systems during an installation. (See Chapter 6.)

- Set up the custom JumpStart files for the systems. (See Chapter 8.)
- Set up the systems to install over the network. (See Chapter 7.)

CHAPTER **2**

Performing an Interactive Installation

- "SPARC: How to Perform an Interactive Installation" on page 5
- "x86: How to Perform an Interactive Installation" on page 10

This chapter provides procedures to perform an interactive installation with either the Solaris Interactive Installation program or Solaris Web Start. These procedures should be done on the system that is being installed.

Note - If you want to install a desktop system from a local CD-ROM, you should use "Planning Your Installation" in the *Information Library for Solaris 2.6 (Intel Platform Edition)* or the *Information Library for Solaris 2.6 (SPARC Platform Edition)*. If you want to perform a custom JumpStart installation on a system, see Chapter 3.

SPARC: How to Perform an Interactive Installation

1. **Check Table 2–1 to make sure the system to be installed is prepared for an interactive installation.**

TABLE 2–1 **SPARC:** Setting Up a System for an Interactive Installation: Task Map

Task	Description	For Instructions, Go To
Back up existing Solaris 1.x (SunOS 4.x) files	If the system has a previous Solaris 1.x (SunOS 4.x) release installed, some Solaris 1.x files can be converted or merged into the Solaris files.	*Solaris 1.x to 2.x Transition Guide*
Check if the system is supported	Check the hardware documentation to see if the system is supported for Solaris 2.6.	"Supported Hardware" in the *Information Library for Solaris 2.6 (SPARC Platform Edition)*
Decide how to upgrade the system if it has a previous version of Solaris installed	If the system has a previous Solaris release installed, you need to determine how to upgrade the system. Make sure you know what to do before and after you upgrade a system.	Chapter 4
Check if the system has enough disk space for the Solaris software	*Optional.* There are many considerations when planning disk space, such as deciding which software group you want to install.	Appendix A
Preconfigure system configuration information	*Optional.* You can use the `sysidcfg` file or the name service to preconfigure installation information (for example, `locale`) for a system, so you won't be prompted to supply the information during the installation.	Chapter 6
Set up the system to install over the network	**For network installations only.** To install a system from a remote Solaris CD image, you need to set up the system to boot and install from an install server or boot server.	Chapter 7

2. **If the system is part of a network, make sure an Ethernet connector or similar network adapter is plugged into your system.**

3. **If you are installing a system connected through a tip line, make sure that your window display is at least 80 columns wide and 24 rows long.**

   Otherwise, the character installation interface will be displayed improperly. You can use the `stty` command to determine the current dimensions of your tip window.

4. **Choose a method to install the software in your product box.**

With This Method ...	You Can ...	You Can't ...
Solaris Web Start	■ Install all the software in your product box (the Solaris software and co-packaged software) at once from a single, browser-based tool ■ Install all the software with the default option, or pick only the software you want to install with the customize option ■ Install software on systems without graphic cards (headless systems). The headless system must have network access to another system with browser capabilities	■ Upgrade from a previous version of Solaris ■ Customize the software installations with choices at the lowest levels (for example, selecting/deselecting packages) ■ Use this method on systems with less than 48 Mbytes of system memory ■ Use this method on systems with less than a 1.05 Gbyte boot disk[1]
Solaris Interactive Installation program, followed by other product installation programs	■ Install the Solaris software first, and then install the co-packaged software separately ■ Upgrade from a previous version of the Solaris software ■ Customize the software installations at the lowest levels (for example, selecting/deselecting packages) ■ Install software on systems without graphic cards (headless systems) ■ Install on systems with the minimum hardware requirements specified for the Solaris software	■ Install all the software in your product box (Solaris software) and co-packaged software) at once from a single tool

[1] This restriction applies only to the size required to run Solaris Web Start; Solaris Web Start will determine whether your system has enough disk space to install the products you select.

5. **If you are using the system's CD-ROM drive to install the Solaris software on the system, insert the Solaris CD-ROM into the system's CD-ROM drive.**

6. **Boot the system.**

If the System Is ...	And You Want To Use ...	Then ...
New, out-of-the-box	Solaris Web Start or the Solaris Interactive Installation program	Turn the system on.
Existing	Solaris Web Start	■ If you are installing from an install server on the network, get the system to the `ok` prompt and type: `ok` **`boot net - browser`** *(A space is required before and after the dash.)* ■ If you are installing from the system's local CD-ROM drive, get the system to the `ok` prompt and type: `ok` **`boot cdrom - browser`** *(A space is required before and after the dash.)*
	Solaris Interactive Installation program	■ If you are installing from an install server on the network, get the system to the `ok` prompt and type: `ok` **`boot net`** ■ If you are installing from the system's local CD-ROM drive, get the system to the `ok` prompt and type: `ok` **`boot cdrom`** **Note -** For systems with older EEPROMs, replace `cdrom` with `sd(0,6,2)` to boot from the system's CD-ROM.

If you have any problems from this point, go to Chapter 5.

Note - For more information about getting the system to the `ok` prompt, see the *System Administration Guide.*

7. **Wait for booting to complete.**

After you enter the `boot` command, the system will check various hardware and system components. This lasts for several minutes.

8. **If prompted, answer the system configuration questions.**

   After booting, the Solaris installation program may prompt you to provide configuration information about the system. If you have preconfigured all the system configuration information, you can skip this step and go to Step 9 on page 9.

9. **Follow the instructions on the screen to install the software on the system, using the installation method that you chose.**

   After the installation is finished, a log of how the Solaris software was installed on the system is saved to a file, as shown in Table 2–2.

TABLE 2–2 Installation Log Locations

If the System Was Installed Using The ...	**Then the Log Is Saved To ...**
Initial installation option	■ Before the system reboots: `/a/var/sadm/system/logs/install_log` ■ After system reboots: `/var/sadm/system/logs/install_log`
Upgrade option	■ Before the system reboots: `/a/var/sadm/system/logs/upgrade_log` ■ After system reboots: `/var/sadm/system/logs/upgrade_log`

10. **If you used the Solaris Interactive Installation program and allocated space for diskless clients or AutoClient systems, use Solstice Host Manager to complete the setup of those clients.**

11. **If you used the Solaris Interactive Installation program, go to the *Information Library* or the *Roadmap* (if provided) to find out how to install the co-packaged software in the product box after the Solaris software is installed.**

x86: How to Perform an Interactive Installation

1. **Check Table 2–3 to make sure the system to be installed is prepared for an interactive installation.**

TABLE 2–3 x86: Setting Up a System for an Interactive Installation: Task Map

Task	Description	For Instructions, Go To
Determine if you need to preserve an existing operating system and user data	If the system has an existing operating system that uses the entire disk, you will have to preserve the existing operating system so it can co-exist with the Solaris software.	Appendix F
Check if the system is supported	Check the hardware documentation to see if the system is supported for Solaris 2.6.	"Supported Hardware" in the *Information Library for Solaris 2.6 (Intel Platform Edition)*
Decide how to upgrade the system if it has a previous version of Solaris installed	If the system has a previous Solaris release installed, you need to determine how to upgrade the system. Make sure you know what to do before and after you upgrade a system.	Chapter 4
Check if the system has enough disk space for the Solaris software	*Optional.* There are many considerations when planning disk space, such as deciding which software group you want to install.	Appendix A

TABLE 2–3 x86: Setting Up a System for an Interactive Installation: Task Map *(continued)*

Task	Description	For Instructions, Go To
Preconfigure system configuration information	*Optional.* You can use the `sysidcfg` file or the name service to preconfigure installation information (for example, `locale`) for a system, so you won't be prompted to supply the information during the installation.	Chapter 6
Set up the system to install over the network	**For network installations only.** To install a system from a remote Solaris CD image, you need to set up the system to boot and install from an install server or boot server.	Chapter 7

2. **If the system is part of a network, make sure an Ethernet connector or similar network adapter is plugged into your system.**

3. **If you are installing a system connected through a tip line, make sure that your window display is at least 80 columns wide and 24 rows long.**

   Otherwise, the character installation interface will be displayed improperly. You can use the `stty` command to determine the current dimensions of your tip window.

4. **Choose a method to install the software in your product box.**

With This Method ...	You Can ...	You Can't ...
Solaris Web Start	■ Install all the software in your product box (the Solaris software and co-packaged software) at once from a single, browser-based tool ■ Install all the software with the default option, or pick only the software you want to install with the customize option	■ Upgrade from a previous version of the Solaris software ■ Customize the software installations at the lowest levels (for example, selecting/deselecting packages) ■ Use this method on systems with less than 48 Mbytes of system memory ■ Use this method on systems with less than a 1.05 Gbyte boot disk[1]
Solaris Interactive Installation program, followed by other product installation programs	■ Install the Solaris software first, and then install the co-packaged software separately ■ Upgrade from a previous version of the Solaris software ■ Customize the software installations at the lowest levels (for example, selecting/deselecting packages) ■ Install on systems with the minimum hardware requirements specified for the Solaris software	■ Install all the software in your product box (Solaris software and co-packaged software) at once from a single tool

[1] This restriction applies only to the size required to run Solaris Web Start; Solaris Web Start will determine whether your system has enough disk space to install the products you select.

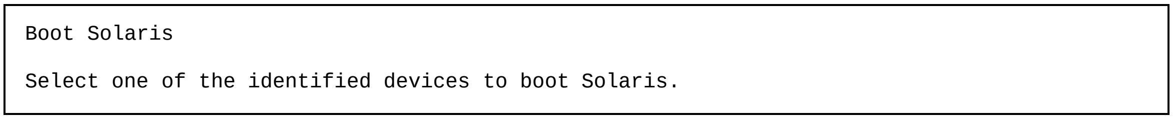
5. **Insert the Configuration Assistant diskette into the system's boot diskette drive (usually the a: drive).**

6. **If you are using the system's CD-ROM drive to install the Solaris software on the system, insert the Solaris CD-ROM into the system's CD-ROM drive.**

7. **If the system is off, turn it on. If the system is on, reboot it.**

   The Device Configuration Assistant program runs to identify the system's devices.

8. **Boot from the system's CD-ROM drive (CD) or from an install server on the network (NET).**



```
Boot Solaris

Select one of the identified devices to boot Solaris.
```

(Continuation)

```
> To make a selection, use the arrow keys, then press Enter to mark it [X].


  Boot Solaris
  -------------------------------------------------------------------
  [ ]  NET : Xircom Pocket Ethernet parallel port card
             Port: 3BC-3BF; IRQ: 7
  [ ]  DISK: IDE(ATA) QUANTUM FIREBALL1080A
             Target: 0; Port: 1F0-1F7, 3F6-3F7; IRQ: 14
  [ ]  CD  : IDE(ATA) IBM-H2344-A4
             Target: 0; Port: 1F0-1F7, 3F6-3F7; IRQ: 14
```

9. **Select the Solaris Interative Installation program or Solaris Web Start.**

```
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 the <ENTER> key.

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

   If you have any problems from this point, go to Chapter 5.

10. **If prompted, answer the system configuration questions.**

    After booting, the Solaris installation program may prompt you to provide configuration information about the system. If you have preconfigured all the system configuration information, you can skip this step and go to Step 11 on page 13.

11. **Follow the instructions on the screen to install the software on the system, using the installation method that you chose.**

    After the installation is finished, a log of how the Solaris software was installed on the system is saved to a file, as shown in Table 2–4.

TABLE 2–4 Installation Log Locations

If the System Was Installed Using The ...	**Then the Log Is Saved To ...**
Initial installation option	■ Before the system reboots: `/a/var/sadm/system/logs/install_log` ■ After system reboots: `/var/sadm/system/logs/install_log`
Upgrade option	■ Before the system reboots: `/a/var/sadm/system/logs/upgrade_log` ■ After system reboots: `/var/sadm/system/logs/upgrade_log`

12. **If you used the Solaris Interactive Installation program and allocated space for diskless clients or AutoClient systems, use Solstice Host Manager to complete the setup of those clients.**

13. **If you used the Solaris Interactive Installation program, go to the *Information Library* or the *Roadmap* (if provided) to find out how to install the co-packaged software in the product box after the Solaris software is installed.**

CHAPTER **3**

Performing a Custom JumpStart Installation

- "SPARC: How to Perform a Custom JumpStart Installation" on page 15
- "x86: How to Perform a Custom JumpStart Installation" on page 19

This chapter provides procedures to perform a custom JumpStart installation. These procedures should be done on the system that is being installed.

Note - If you want to install a desktop system from a local CD-ROM, you should use "Planning Your Installation" in the *Information Library for Solaris 2.6 (Intel Platform Edition)* or the *Information Library for Solaris 2.6 (SPARC Platform Edition)*. If you want to perform an interactive installation on a system, see Chapter 2.

SPARC: How to Perform a Custom JumpStart Installation

1. **Check Table 3–1 to make sure the system to be installed is set up for a custom JumpStart installation.**

TABLE 3–1 **SPARC:** Setting Up a System for a Custom JumpStart Installation: Task Map

Task	Description	For Instructions, Go To
Back up existing Solaris 1.x (SunOS 4.x) files	If the system has a previous Solaris 1.x (SunOS 4.x) release installed, some Solaris 1.x files can be converted or merged into the Solaris files. You can use begin and finish scripts to do this.	*Solaris 1.x to 2.x Transition Guide*
Check if the system is supported	Check the hardware documentation to see if the system is supported for Solaris 2.6.	"Supported Hardware" in the *Information Library for Solaris 2.6 (SPARC Platform Edition)*
Decide how to upgrade the system if it has a previous version of Solaris installed	If the system has a previous Solaris release installed, you need to determine how to upgrade the system. Make sure you know what to do before and after you upgrade a system; this will help you set up your profiles, begin scripts, and finish scripts.	Chapter 4
Check if the system has enough disk space for the Solaris software	*Optional.* There are many considerations when planning disk space, such as deciding which software group you want to install.	Appendix A
Preconfigure system configuration information	*Optional.* You can use the `sysidcfg` file or the name service to preconfigure installation information (for example, `locale`) for a system, so you won't be prompted to supply the information during the installation.	Chapter 6
Prepare system for custom Jumpstart installation	You need to do some initial setup work before you can install a system using custom JumpStart.	Chapter 8

TABLE 3–1 SPARC: Setting Up a System for a Custom JumpStart Installation: Task Map *(continued)*

Task	Description	For Instructions, Go To
Set up the system to install over the network	**For network installations only.** To install a system from a remote Solaris CD image, you need to set up the system to boot and install from an install server or boot server.	Chapter 7

2. **If the system is part of a network, make sure an Ethernet connector or similar network adapter is plugged into your system.**

3. **If you are installing a system connected through a tip line, make sure that your window display is at least 80 columns wide and 24 rows long.**

   Otherwise, the character installation interface will be displayed improperly. You can use the `stty` command to determine the current dimensions of your tip window.

4. **If you are using the system's CD-ROM drive to install the Solaris software on the system, insert the Solaris CD-ROM into the system's CD-ROM drive.**

5. **If you are using a profile diskette to perform a custom JumpStart installation, insert the profile diskette into the system's diskette drive.**

6. **Boot the system.**

If the System Is...	Then ...
New, out-of-the-box	Turn the system on.
Existing	■ If you are installing from an install server on the network, get the system to the `ok` prompt and type: ok **`boot net - install`** *(A space is required before and after the dash.)* ■ If you are installing from the system’s local CD-ROM drive, get the system to the `ok` prompt and type: ok **`boot cdrom - install`** *(A space is required before and after the dash.)* **Note -** For systems with older EEPROMs, replace `cdrom` with `sd(0,6,2)` to boot from the system’s CD-ROM.

If you have any problems from this point, go to Chapter 5.

Note - For more information about getting the system to the `ok` prompt, see the *System Administration Guide.*

7. **Wait for booting to complete.**

   After you enter the `boot` command, the system will check various hardware and system components. This lasts for several minutes.

8. **If prompted, provide information about the system.**

   After booting, the Solaris installation program may prompt you to provide configuration information about the system. If you have preconfigured all the system configuration information, you can skip this step and go to Step 9 on page 18.

9. **Wait as the Solaris software installs on the system.**

   After the installation is finished, a log of how the Solaris software was installed on the system is saved to a file, as shown in Table 3–2.

TABLE 3–2 Installation Log Locations

If the System Was Installed Using The ...	Then the Log Is Saved To ...
Initial installation option	■ Before the system reboots: `/a/var/sadm/system/logs/install_log` ■ After system reboots: `/var/sadm/system/logs/install_log`
Upgrade option	■ Before the system reboots: `/a/var/sadm/system/logs/upgrade_log` ■ After system reboots: `/var/sadm/system/logs/upgrade_log`

10. **If the system's profile allocated space for diskless clients or AutoClient systems, use Solstice Host Manager to complete the setup of those clients.**

x86: How to Perform a Custom JumpStart Installation

1. **Check Table 3–3 to make sure the system to be installed is set up for a custom JumpStart installation.**

TABLE 3–3 **x86:** Setting Up a System for a Custom JumpStart Installation: Task Map

Task	Description	For Instructions, Go To
Determine if you need to preserve an existing operating system and user data	If the system has an existing operating system that uses the entire disk, you will have to preserve the existing operating system so it can co-exist with the Solaris software. This decision will determine how to specify the `fdisk` keyword in the system's profile.	Appendix F
Check if the system is supported	Check the hardware documentation to see if the system is supported for Solaris 2.6.	"Supported Hardware" in the *Information Library for Solaris 2.6 (Intel Platform Edition)*
Decide how to upgrade the system if it has a previous version of Solaris installed	If the system has a previous Solaris release installed, you need to determine how to upgrade the system. Make sure you know what to do before and after you upgrade a system; this will help you set up your profiles, begin scripts, and finish scripts.	Chapter 4
Check if the system has enough disk space for the Solaris software	*Optional.* There are many considerations when planning disk space, such as deciding which software group you want to install.	Appendix A
Preconfigure system configuration information	*Optional.* You can use the `sysidcfg` file or the name service to preconfigure installation information (for example, `locale`) for a system, so you won't be prompted to supply the information during the installation.	Chapter 6
Prepare system for custom Jumpstart installation	You need to do some initial setup work before you can install a system using custom JumpStart.	Chapter 8

TABLE 3–3 x86: Setting Up a System for a Custom JumpStart Installation: Task Map *(continued)*

Task	Description	For Instructions, Go To
Set up system to install over the network	**For network installations only.** To install a system from a remote Solaris CD image, you need to set up the system to boot and install from an install server or boot server.	Chapter 7

2. **If the system is part of a network, make sure an Ethernet connector or similar network adapter is plugged into your system.**

3. **If you are installing a system connected through a tip line, make sure that your window display is at least 80 columns wide and 24 rows long.**

   Otherwise, the character installation interface will be displayed improperly. You can use the `stty` command to determine the current dimensions of your tip window.

4. **Insert the Configuration Assistant diskette or profile diskette into the system's boot diskette drive (usually the a: drive).**

   Note - If you are using a profile diskette to perform a custom JumpStart installation, you must insert the profile diskette (which is also a copy of the Configuration Assistant diskette) into the system's a: diskette drive.

   ---

5. **If you are using the system's CD-ROM drive to install the Solaris software on the system, insert the CD-ROM into the CD-ROM drive.**

6. **If the system is off, turn it on. If the system is on, reboot it.**

   The Device Configuration Assistant program runs to identify the system's devices.

7. **Boot from the system's CD-ROM drive (CD) or from an install server on the network (NET).**

```
Boot Solaris
```

(continued)

(Continuation)

```
Select one of the identified devices to boot Solaris.

> To make a selection, use the arrow keys, then press Enter to mark it [X].


  Boot Solaris
  ------------------------------------------------------------------
  [ ]  NET : Xircom Pocket Ethernet parallel port card
             Port: 3BC-3BF; IRQ: 7
  [ ]  DISK: IDE(ATA) QUANTUM FIREBALL1080A
             Target: 0; Port: 1F0-1F7, 3F6-3F7; IRQ: 14
  [ ]  CD  : IDE(ATA) IBM-H2344-A4
             Target: 0; Port: 1F0-1F7, 3F6-3F7; IRQ: 14
```

8. **Select the custom JumpStart installation method.**

```
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 the <ENTER> key.

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

   If you have any problems from this point, go to Chapter 5.

9. **If prompted, provide information about the system.**

   After booting, the Solaris installation program may prompt you to provide configuration information about the system. If you have preconfigured all the system configuration information, you can skip this step and go to Step 10 on page 22.

10. **Wait as the Solaris software installs on the system.**

    After the installation is finished, a log of how the Solaris software was installed on the system is saved to a file, as shown in Table 3–4.

TABLE 3–4 Installation Log Locations

If the System Was Installed Using The ...	Then the Log Is Saved To ...
Initial installation option	■ Before the system reboots: `/a/var/sadm/system/logs/install_log` ■ After system reboots: `/var/sadm/system/logs/install_log`
Upgrade option	■ Before the system reboots: `/a/var/sadm/system/logs/upgrade_log` ■ After system reboots: `/var/sadm/system/logs/upgrade_log`

11. **If the system's profile allocated space for diskless clients or AutoClient systems, use Solstice Host Manager to complete the setup of those clients.**

CHAPTER **4**

Upgrading a System

Ways to Upgrade a System

When you try to install a new version of Solaris software on an existing Solaris system, the installation program allows you to choose one of the following options to copy the Solaris software to disk:

- **Upgrade –** This option merges the new version of Solaris software with the existing files on the system's disks. It saves as many modifications that you have made to the previous version of Solaris software as possible. This is the preferred way to upgrade a system.
- **Initial –** This option overwrites the system's disk with the new version of Solaris software. You must back up any local modifications that you have made to the previous version of Solaris software before you begin the installation, and you must restore the local modifications after the installation completes.

This chapter mainly focuses on using the upgrade option.

Note - The upgrade option is not available for 4.1.x systems. Because you must use the initial installation option, backing up your data is critical. See the *Solaris 1.x to 2.x Transition Guide* for information.

Upgrade With Disk Space Reallocation

The upgrade option in the Solaris Interactive Installation program provides the ability to reallocate disk space if the current file systems don't have enough space for the upgrade. By default, an auto-layout feature tries to determine how to reallocate the disk space so the upgrade can succeed. If auto-layout can't determine how to reallocate disk space, you must specify which file systems can be moved or changed and run auto-layout again.

If you're creating an upgrade profile and the current file systems don't have enough space for the upgrade, you can use the `backup_media` and `layout_constraint` keywords to reallocate disk space. See "Reallocating Disk Space for an Upgrade" on page 99 for an example of how to use the `backup_media` and `layout_constraint` keywords in a profile.

Frequently Asked Questions About Upgrading

- Will I be able to use the upgrade option on my system?

  The upgrade option is supported on any system with Solaris 2.4 or later installed. Type the following command to see what version of Solaris software the system is running.

  ```
  $ uname -a
  ```

- Do I have to back out patches before I use the upgrade option?

  No.

- How do I use the upgrade option with custom JumpStart?

  You must specify `install_type upgrade` in your profiles.

- What if the Solaris Interactive Installation program doesn't provide the upgrade option, but the system should be upgradable?

  See Chapter 5 for more details.

- How can I test my profiles that use the upgrade option?

  You can use the `pfinstall -D` command to test profiles that use the upgrade option. This enables you to see if a profile will do what you want before you use it to upgrade a system. This is especially useful with the new "upgrade with disk space reallocation" feature.

To test an upgrade profile, you must run `pfinstall -D` command on the system that you're going to upgrade, because you need to test the profile against the system's disk configuration and its currently installed software. You cannot test an upgrade profile using a disk configuration file. For details, see "Testing a Profile" on page 118.

- Can I automatically upgrade to another software group?

  No. For example, if you previously installed the end user software group on your system, you cannot use the upgrade option to upgrade to the developer software group. However, you can always add software to the system during the upgrade that is not part of the currently installed software group.

- Where does the installation program log any local modifications that the upgrade couldn't preserve?

  - Before the system reboots: `/a/var/sadm/system/data/upgrade_cleanup`

  - After system reboots: `/var/sadm/system/data/upgrade_cleanup`

- Where does the installation program log what happened during the upgrade?

  - Before the system reboots: `/a/var/sadm/system/logs/upgrade_log`

  - After system reboots: `/var/sadm/system/logs/upgrade_log`

What to Do Before Upgrading

1. **Consider the following information before you upgrade an existing system to the new Solaris release.**
   - Check the *Release Notes for Solaris 2.6* to see if:
     - There is any Solaris software that you use that is no longer provided in the new release.
     - Any of the changes or enhancements to the Solaris software will affect anything that you currently do.
     - You need any of the available patches. The most current patch list is provided by SunSolve on the Internet: `http://sunsolve.sun.com/`
   - Make sure the system is supported by the new Solaris release.
   - Make sure the third-party or co-packaged software you are using will still run on the new Solaris release.
   - To avoid upgrade problems or loss of data, check the documentation of the third-party or co-packaged software you are running before upgrading.

Caution - Table 4–1 provides a list of known software requiring additional instructions before upgrading. Because this list is not complete, always check the documentation of the third-party or co-packaged software you are running before upgrading.

TABLE 4–1 Software Requiring Additional Instructions Before Upgrading

Software	Problem Summary
Online: DiskSuite™	Metadevices cannot be upgraded. You must comment out metadevices entries (`/dev/md`) in the `/etc/vfstab` file before starting the upgrade option. See the Online: DiskSuite documentation for instructions.
Prestoserve	If you start the upgrade process by shutting down the system using `init 0`, you can lose data. See the Prestoserve documentation for shutdown instructions.

2. **Back up your system.**

   Always back up an existing system before upgrading a system. The safest backup to perform is a level 0 dump of all the file systems connected to the system being upgraded. If you do not have an established backup procedure, see the *System Administration Guide*.

3. **If the configuration information for your system isn't preconfigured (see Chapter 6), use the following table to find the needed system configuration information that you will be prompted for**:

Configuration Information	Example	Command for Finding Information
System's name (host name)	`crater`	**`/usr/bin/uname -u`**
Primary network interface	`le0`	**`ifconfig -a`**

Configuration Information	Example	Command for Finding Information

IP address	`129.221.2.1`	**`ypmatch`** ***host_name*** **`host`**
		or
		`nismatch` ***host_name*** **`hosts.org_dir`**

Domain name	`chandy.West.Arp.COM`	**`/usr/bin/domainname`**
System part of subnet?		Check for existing subnet in `/etc/netmasks`
Netmask	`255.255.255.0`	**`more /etc/netmasks`**

4. **Make sure you have a backup media ready for the upgrade (if neccessary).**

   If some of the Solaris based files systems (for example, root (/), /usr) on the system to be upgraded don't have much space left, you'll probably need to reallocate disk space during the upgrade. If so, the Solaris Interactive Installation program will prompt you for a backup media, which is required to temporarily back up file systems that need to be reallocated.

   You can use one of the following for the backup media:

   - Local file system - You can use a local file system on the system to be upgraded for the backup media. However, the local file file system can't be used to help reallocate disk space during the upgrade. You'll need to provide the installation program with either the file system's block device path or the file system's mount point name.
   - Local tape - Make sure a tape drive is connected to the system to be upgraded and the system knows about the tape drive. You'll need to provide the installation program with the character (raw) device path for the diskette.
   - Local diskette - Make sure the system to be upgraded has a diskette drive. You'll need to provide the installation program with the character (raw) device path for the diskette drive.
   - Remote file system (NFS) - If you want to use an NFS file system on a remote system, the system to be upgraded must be on the same network as the NFS file system, and the NFS file system must be shared properly. You'll need to provide the installation program with the name or IP address of the NFS server and the absolute path to the NFS file system.
   - Remote system (`rsh`) - If you want to use a directory on a remote system that can be reached by a remote shell (`rsh`), the system to be upgraded must be on the same network as the remote system, and it must have access to the remote system through the remote system's `.rhosts` file. You'll need to provide the

installation program with the name of the remote system, the user login, and the absolute path to the directory.

What to Do After Upgrading

After the system finishes upgrading, the installation program leaves it at the superuser prompt in single-user mode.

1. **Clean up the system if necessary.**

   When you use the upgrade option, the Solaris installation program merges local software modifications of the existing system with the new software; however, in some cases, the merge is not possible. Refer to the following file to see if you need to fix any of the local modifications that the upgrade could not preserve:

   `/a/var/sadm/system/data/upgrade_cleanup`

Caution - Make sure you look at all the information in the `upgrade_cleanup` file. Your system may not boot if you fail to fix the unpreserved local modifications.

2. **Reboot the system.**

```
# reboot
```

3. **If you upgraded an OS server, upgrade clients with different platforms and platform groups.**

   If you've upgraded a heterogeneous OS server, clients of that server are automatically upgraded only if their platform (x86 or SPARC) and platform group (for example, sun4m or i386) is supported by the Solaris CD. For example, if you upgrade a SPARC server using the SPARC Solaris CD, only SPARC clients that share the platform group on the CD are upgraded.

   To upgrade clients with different platforms and platform groups, you must use the `server_upgrade` command. See the `server_upgrade(1M)` man page for more instructions.

CHAPTER **5**

Troubleshooting

This chapter provides a list of specific error messages and generic problems that you may encounter when installing the Solaris software. Start by using the following list to identify where in the installation process the problem is occurring.

- "Setting Up Network Installations" on page 31
- "Booting a System" on page 32
- "Booting a System Over the Network" on page 36
- "Installing Solaris (Initial)" on page 41
- "Installing Solaris (Upgrade)" on page 42

Setting Up Network Installations

```
Error: Unknown client ''host_name''
```

Problem	**How to Fix the Problem**
The *host_name* argument in the `add_install_client` command is not a host in the name service.	Add the host *host_name* to the NIS or NIS+ name service and execute the `add_install_client` command again.

Booting a System

Error Messages

```
le0: No carrier - transceiver cable problem
```

Problem	**How to Fix the Problem**
The system is not connected to the network.	If this is a non-networked system, ignore this message. If this is a networked system, make sure the Ethernet cabling is attached securely.

```
The file just loaded does not appear to be executable
```

Problem	**How to Fix the Problem**
The system cannot find the proper media for booting.	Verify that the system has been set up properly to install over the network from an install server. For example, make sure you specified the right platform group for the system when you set it up. Also, if you did not copy the Solaris CD, make sure the Solaris CD on the install server is mounted and accessible.

```
boot: cannot open /kernel/unix
```

Problem	**How to Fix the Problem**
SPARC based systems only. This error occurs when you override the boot file location by explicitly setting it to `/kernel/unix`. In Solaris 2.6, the kernel no longer resides in `/kernel/unix`, but in `/platform/<`*arch*`>/kernel/unix`.	Reset the boot file in the PROM to “ “ (blank).

```
Can’t boot from file/device
```

Problem	How to Fix the Problem
The installation program can’t find the Solaris CD in the system’s CD-ROM drive.	Make sure the: ■ CD-ROM drive is installed properly or is turned on ■ Solaris CD-ROM is inserted into the CD-ROM drive

```
WARNING: clock gained xxx days -- CHECK AND RESET
DATE!
```

Problem	How to Fix the Problem
SPARC based systems only. This is an informational message.	Ignore the message and continue with the installation.

```
Not a UFS filesystem
```

Problem	How to Fix the Problem
x86 based systems only. When Solaris software was installed (either through the interactive or custom JumpStart method), the default boot drive was not selected. When an alternate boot disk is selected, you must use the Configuration Assistant diskette to boot the system from that point on.	Insert the Configuration Assistant diskette into the system’s boot diskette drive (usually the `a:` drive).

General Problems

Problem	How to Fix the Problem
x86 based systems only. The Solaris root slice must reside within the first 1024 cylinders of the disk. If it does not, the installation fails after booting.	If the first fdisk partition is primary DOS (PRI DOS), use the `fdisk` program to delete space from it. Try booting again. If the first fdisk partition is extended DOS (EXT DOS) or another operating system, use the `fdisk` program to delete it. Try booting again.

Problem	How to Fix the Problem
x86 based systems only. System hangs or panics when non-memory PC cards are inserted.	Non-memory PC cannot use the same memory resources used by other devices. To correct this, use a DOS debugger to identify device memory usage, then manually reserve memory resources for the PC card device using the following instructions. 1. Boot the system using the Configuration Assistant diskette. 2. Go to the Device Tasks menu. 3. Select Review / Edit Devices. 4. Select Add Device. 5. Select Define Device. 6. Enter a unique name following the EISA ID naming conventions (for example, ITD4001), and choose Continue. 7. Select Memory Address from the list of resources, and choose Continue. 8. Enter the address range to reserve (for example, CA800–CFFFF), and choose Continue. 9. Return to the Device Tasks menu and select Save Configuration. 10. Reboot the Solaris operating environment.

Problem	How to Fix the Problem
x86 based systems only. The BIOS primary drive on your system was not detected by the Configuration Assistant program during the pre-booting phase.	■ If you are using old drives, they may be unsupported. Check Supported Hardware in the *Information Library for Solaris 2.6 (Intel Platform Edition)*. ■ Make sure the ribbon and power cables are plugged in correctly. Check the manufacturer's documentation. ■ If only one drive is attached to the controller, designate the drive as the master drive by setting jumpers. Some drives have different jumper settings for a single master, as opposed to a master operating with a slave. Connect the drive to the connector at the end of the cable to reduce signal ringing that occurs when an unused connector is dangling at the end of the cable. ■ If two drives are attached to the controller, jumper one drive as the master (or as a master operating with a slave), and jumper the second drive as a slave. ■ If one drive is a hard disk and the second a CD-ROM drive, designate the drive as the slave drive by setting jumpers. It doesn't matter which drive is plugged into which drive connection on the cable. ■ If there are persistent problems with two drives on a single controller, attach one drive at a time to verify that each works. Jumper the drive as master or single master and use the drive connector at the end of the IDE ribbon cable to attach the drive. Verify that each drive works, then jumper the drives back into a master and slave configuration. ■ If the drive is a disk drive, use the BIOS setup screen to ensure that the drive type (which indicates the number of cylinders, heads, and sectors) is correctly configured. Some BIOS software may have a feature that automatically detects the drive type. ■ If the drive is a CD-ROM drive, use the BIOS setup screen to configure the drive type as a CD-ROM drive, when the BIOS software has this capability. ■ If MS-DOS does not recognize the drive, there is probably a hardware or BIOS configuration problem. For many systems, IDE CD-ROM drives are only recognized by MS-DOS if a MS-DOS CD-ROM driver has been installed.

Problem	How to Fix the Problem
x86 based systems only. The IDE or CD-ROM drive on your system was not found by the Configuration Assistant program in the pre-booting phase.	■ If disks are disabled in the BIOS, use the Configuration Assistant diskette to boot from the hard disk. ■ If the system has no disks, it may be a diskless client.

Problem	How to Fix the Problem
x86 based systems only. System hangs before getting the prompt.	See Configuring Devices in the *Information Library for Solaris 2.6 (Intel Platform Edition).*

Booting a System Over the Network

Error Messages

```
WARNING: getfile:
RPC failed: error 5 (RPC Timed out).
```

Problem	How to Fix the Problem
This error occurs when you have two or more servers on a network responding to an install client's boot request. The install client connects to the wrong boot server, and the installation hangs. The following specific reasons may cause this error:	
Reason 1: There may be `/etc/bootparams` files on different servers with an entry for this install client.	*Solution for Reason 1:* Make sure that servers on the network do not have multiple `/etc/bootparams` entries for the install client. If they do, remove duplicate client entries in the `/etc/bootparams` file on all install and boot servers except the one you want the install client to use.

Problem	How to Fix the Problem
Reason 2: There may be multiple `/tftpboot` or `/rplboot` directory entries for this install client.	*Solution for Reason 2:* Make sure that servers on the network do not have multiple `/tftpboot` or `/rplboot` directory entries for the install client. If they do, remove duplicate client entries from the `/tftpboot` or `/rplboot` directories on all install and boot servers except the one you want the install client to use.
Reason 3: There may be an install client entry in the `/etc/bootparams` file on a server and an entry in another `/etc/bootparams` file enabling all systems to access the profile server. Such an entry would look like this: `* install_config=`*profile_server*`:`*path* A line like this in the NIS or NIS+ `bootparams` table would also cause this error.	*Solution for Reason 3:* If there's a wildcard entry in the name service `bootparams` map or table (for example, `* install_config=`), delete it and add it to the `/etc/bootparams` file on the boot server.

```
No network boot server. Unable to install the system.
 See installation instructions.
```

Problem	How to Fix the Problem
SPARC based systems only. This error occurs on a system that you are attempting to install over the network. The system is not set up properly.	Make sure you set up the system to install over the network (see "Setting Up Systems to Be Installed Over the Network" on page 66).

```
prom_panic: Could not mount filesystem
```

Problem	How to Fix the Problem
SPARC based systems only. This error occurs when you are doing a network installation, but the boot software cannot locate the Solaris installation image (either the Solaris CD or a copy of the Solaris CD on the install server).	Make sure that the installation software is mounted and shared. If installing from the install server's CD-ROM drive, make sure the Solaris CD is inserted in the CD-ROM drive, is mounted, and is shared in the `/etc/dfs/dfstab` file. If installing from a copy of the Solaris CD on the install server's disk, make sure the directory path to the copy is shared in the `/etc/dfs/dfstab` file.

```
Timeout waiting for ARP/RARP packet...
```

Problem	How to Fix the Problem
SPARC based systems only. The client is trying to boot over the network, but it cannot find a system that knows about it.	Verify the system's host name is in the NIS or NIS+ name service. Also, verify the `bootparams` search order in the boot server's `/etc/nsswitch.conf` file. For example, the following line in the `/etc/nsswitch.conf` file indicates the software will first look in the NIS maps for `bootparams` information. If not found there, software will look in the boot server's `/etc/bootparams` file. `bootparams: nis files`

```
ip: joining multicasts failed on tr0 - will use link layer broadcasts for multicast
```

Problem	How to Fix the Problem
x86 based systems only. You will see this error message when you boot a system with a token ring card. Ethernet multicast and token ring multicast do not work the same way. The driver returns this error message to indicate that an invalid multicast address was given.	Ignore this error message. If multicast doesn't work, IP uses layer broadcasts instead and it won't cause the installation to fail.

```
Requesting Internet address for Ethernet_Address
```

Problem	How to Fix the Problem
x86 based systems only. The client is trying to boot over the network, but it cannot find a system that knows about it.	Verify the system’s host name is in the NIS or NIS+ name service. If the system’s host name is in the NIS or NIS+ name service, and the system continues to print this error message, try rebooting.

```
RPC: Timed out
No bootparams (whoami) server responding; still trying...
```

Problem	How to Fix the Problem
x86 based systems only. The client is trying to boot over the network, but it cannot find a system with an entry in the `/etc/bootparams` file on the install server.	Use `add_install_client` on the install server. This will add the proper entry in the `/etc/bootparams` file, enabling the client to boot over the network.

```
Still trying to find a RPL server...
```

Problem	How to Fix the Problem
x86 based systems only. The system is trying to boot over the network, but the server is not set up to boot this system.	On the install server, execute `add_install_client` for the system to be installed. The `add_install_client` command sets up an `/rplboot` directory, which contains the necessary network boot program.

General Problems

Problem	How to Fix the Problem
The system boots over the network, but from a system other than the specified install server.	On the name server, update the `/etc/bootparams` entry for the system being installed. The entry should conform to the following syntax: *install_system* `root=`*boot_server*`:`*path* `install=`*install_server*`:`*path* Also, ensure there is only one `bootparams` entry on the subnet for the install client.

Problem	How to Fix the Problem

SPARC based systems only.
After you set up an install server and configure the system to install over the network, the system still does not boot.
	Be sure the `tftpd` daemon is running on the install server. Type the following command and press Return:
	`#` **`ps -ef | grep tftpd`**
	If this command does not return a line indicating the `tftpd` daemon is running, edit the `/etc/inetd.conf` file and remove the comment (#) character from the following line:
	`# tftp dgram udp`
	`wait root /usr/sbin/in.tftpd`
	`in.tftpd -s /tftpboot`
	After making this change, try booting the system again.

Problem	How to Fix the Problem
x86 based systems only. After setting up an install server and configuring the system to install over the network, the system still does not boot.	Be sure the `tftpd` daemon is running on the install server. Type the following command and press Return: `#` **`ps -ef | grep rpld`** If this command does not return a line indicating the `rpld` daemon is running, execute the following command: `#` **`/usr/sbin/rpld`** After making this change, try booting the system again.

Installing Solaris (Initial)

```
/cdrom/Solaris_2.x/SUNWxxxx/reloc.cpio: Broken pipe
```

Problem	How to Fix the Problem
Bug ID: 1212370 This error message does not affect the installation.	Ignore the message and continue with the installation.

Problem	How to Fix the Problem
x86 based systems only. IDE disk drives do not automatically map out bad blocks like other drives supported by Solaris software. Before installing Solaris on an IDE disk, you may want to perform a surface analysis on the disk.	To perform surface analysis on an IDE disk, follow this procedure: 1. Start the Solaris Interactive Installation program, as described in “SPARC: How to Perform an Interactive Installation” on page 5. The Solaris Interactive Installation program will start either a graphical user interface (GUI) or a character user interface (CUI), depending on whether you have a graphics or non-graphics monitor. 2. When either the GUI or CUI program starts, enter information and select the Continue option on the first few screens. 3. When you see the *Installing Solaris - Initial* screen, select the Exit option and exit the installation. 4. If you are using the GUI installation program, open a command tool window for the remaining steps in this procedure. If you are using the CUI installation program, use the system shell for the remaining steps in this procedure. 5. Start the `format` program by typing **`format`** 6. Specify the IDE disk drive on which you want to perform a surface analysis. **Note -** IDE drives do not include a target number. The IDE drive naming convention is `c`*x*`d`*y*, where `c`*x* is the controller number and `d`*y* is the device number. 7. At the `format>` prompt, type **`fdisk`**. Use the `fdisk` program to create a Solaris partition on the disk. (If a Solaris fdisk partition already exists, leave it alone.) 8. At the `format>` prompt, type **`analyze`**.

Problem	How to Fix the Problem
	9. At the `analyze>` prompt, type **`config`**. This will show you the current settings for a surface analysis. If you want to change any settings, type `setup`. 10. At the `analyze>` prompt, type **`read`**, **`write`**, or **`compare`** for the type of surface analysis to be performed. If `format` finds bad blocks, it will re-map them. 11. At the `analyze>` prompt, type **`quit`**. 12. You may want to specify blocks to re-map. If so, at the `format>` prompt, type **`repair`**. 13. Type **`quit`** to quit the `format` program. 14. Choose `Restart Install` from the Workspace menu to resume the GUI installation or type **`suninstall`** to resume the CUI installation.

Installing Solaris (Upgrade)

Error Messages

```
No upgradeable disks
```

Problem	How to Fix the Problem
Bug ID: 1191792 A swap entry in the `/etc/vfstab` file is causing the upgrade to fail.	Comment out the following lines in the `/etc/vfstab` file: ■ All swap files and slices on disks not being upgraded ■ Swap files that are no longer present ■ Any unused swap slices

General Problems

Problem	How to Fix the Problem
The upgrade fails because the Solaris installation program could not mount metadevices on the system.	Comment out all metadevices in the system’s `/etc/vfstab` file. The upgrade option does not support metadevices.

Problem	**How to Fix the Problem**
Bug ID: 1170953 The upgrade option is not presented even though there is a version of Solaris software that's upgradable on the system. The following specific reasons may cause this problem:	
Reason 1: The `/var/sadm` directory is a symlink or it is mounted from another file system.	*Solution for Reason 1:* Move the `/var/sadm` directory into the root (`/`) or `/var` file system.

Reason 2: The `/var/sadm/softinfo/INST_RELEASE` file is missing.	*Solution for Reason 2:* Create a new `INST_RELEASE` file by using the following template:
	`OS=Solaris`
	`VERSION=2.`*x*
	`REV=0`
	where *x* is the version of Solaris software on the system.

Problem	**How to Fix the Problem**
The upgrade fails for reasons beyond your control, such as a power failure or a network connection failure, and the system is left in an unbootable state.	1. Reboot the system from the Solaris CD or from the network. 2. Choose the upgrade option for installation. The Solaris Interactive Installation program will determine if the system has been partially upgraded and will continue the upgrade.

Problem	**How to Fix the Problem**
The upgrade fails because the Solaris installation program could not mount a file system. During an upgrade, the installation program attempts to mount all the file systems listed in the system's `/etc/vfstab` file on the root file system being upgraded. If the installation program cannot mount a file system, it fails and exits.	Make sure all file systems in the system's `/etc/vfstab` file can be mounted. Comment out any file systems in the `/etc/vfstab` file that can't be mounted or that may cause the problem, so the installation program doesn't try to mount them during the upgrade. **Note -** Any system-based file systems that contain software to be upgraded (for example, `/usr`) cannot be commented out.

Problem	How to Fix the Problem
There is not enough space on the system for the upgrade. Check the following reasons for the space problem and see if you can fix it without using auto-layout to reallocate space:	
Reason 1: Since the automounter is not active during an upgrade, the Solaris installation program installs any package's files or directories that are symbolic links to automounted file systems. If a symbolic link is overwritten, the upgrade may fail because of insufficient disk space.	*Solution for Reason 1:* During the upgrade, delete software packages in the Customize Software screen that will create files or directories on the automounted file systems. Then the Solaris installation program will not overwrite the symbolic link with a package's files or directories.
Note - The `/var/mail` and `/var/news` directories, which usually reside on an automounted file system, are not affected by an upgrade.	
Reason 2: New software has been added to the software group that you are upgrading or some of the existing software has increased in size. During an upgrade, the Solaris installation program installs any new software that is part of the software group previously installed on the system, and it also upgrades any existing packages on the system.	*Solution for Reason 2:* During the upgrade, delete software packages in the Customize Software screen that install into the file systems that need more space. Especially look for any new packages that have been added to the Solaris release that the system doesn't need.

CHAPTER **6**

Preconfiguring System Configuration Information

The Solaris installation program needs to obtain configuration information about a system (such as the system's peripheral devices, host name, IP address, name service) before it can install the system. Before prompting the user for the configuration information, the installation program looks for the information in either the specified `sysidcfg` file or the name service databases (in that order).

This section describes how to preconfigure the information in a `sysidcfg` file or in the name service databases, so you can avoid being prompted for the information every time you install a system. For example, if you have a large number of systems and you don't want to be prompted for the time zone every time you install one of the systems, you can preconfigure the time zone and have it automatically set during each installation. Preconfiguring system configuration information is one of the most important steps to completely automate custom JumpStart installations.

Choose a Method for Preconfiguring

There are two ways to preconfigure system configuration information. You can add the information to:

- A `sysidcfg` file (on a remote system or a diskette).
- The name service available at your site.

Use Table 6–1 to determine which method to use to preconfigure system configuration information.

TABLE 6–1 Methods to Preconfigure System Configuration Information

If You Want to Preconfigure ...	And Your Platform Is ...	Can You Preconfigure With the `sysidcfg` File?	Can You Preconfigure With the Name Service?
Name service	All	Yes	Yes
Domain name	All	Yes	No
Name server	All	Yes	No
Network interface	All	Yes	No
Host name	All	Yes[1]	Yes[2]
IP address	All	Yes[1]	Yes[2]
Netmask	All	Yes	No
Root password	All	Yes	No
Language (locale) in which to display the install program and desktop	All	Yes	Yes
Terminal type	All	Yes	No
Time zone	All	Yes	Yes[2]
Date and time	All	Yes	Yes
Monitor type	x86	Yes	No
Keyboard language, keyboard layout	x86	Yes	No
Graphics card, color depth, display resolution, screen size	x86	Yes	No
Pointing device, number of buttons, IRQ level	x86	Yes	No
Power Management™ (autoshutdown)[3]	SPARC	No	No

[1] Because this information is system specific, edit the name service rather than create a different `sysidcfg` file for each system.

TABLE 6–1 Methods to Preconfigure System Configuration Information *(continued)*

2. This information is automatically preconfigured if you've added clients using Solstice AdminSuite™ Host Manager.
3. This system configuration information cannot be preconfigured through the sysidcfg file or the name service. See "Preconfiguring Power Management Information" on page 55 for more details.

Preconfiguring With the sysidcfg File

The sysidcfg file preconfigures the information through a set of keywords that specify the pieces of information you want to preconfigure. The keywords are described in Table 6–2. You can choose to provide one or more of the keywords to preconfigure as much information as you want.

Every system that requires different configuration information must have a different sysidcfg file. For example, you could use the same sysidcfg file to preconfigure the time zone for multiple systems if you wanted all the systems to have the same time zone. However, if you wanted to preconfigure a different root password for each of those systems, then each system would need to have its own sysidcfg file.

The sysidcfg file can reside on a shared NFS network directory or the root directory on a UFS or PCFS diskette in the system's diskette drive. If you put the sysidcfg file on a shared NFS network directory, you must use the –p option of the add_install_client command (when you set up the system to install over the network) to specify where the system can find the sysidcfg file when it installs.

If you put the sysidcfg file on a diskette, you must make sure the diskette is in the system's diskette drive when the system boots (on x86 based systems, the sysidcfg file should reside on the Configuration Assistant diskette). Also, if you are using a profile diskette, the sysidcfg file should reside on the profile diskette.

Note - Only one sysidcfg file can reside in a directory or diskette. If you are creating more than one sysidcfg file, each file must reside in a different directory or diskette.

Syntax Rules

Syntax Rules	For Example ...

Keywords can be listed in any order.	`pointer=MS-S`
	`display=ati {size=15-inch}`

Keywords are not case sensitive.	`TIMEZONE=US/Central`
	`terminal=PC Console`

Values can optionally be enclosed in single (‘) or double quotes (“).	`network_interface='none'`

Only one instance of a keyword is valid; however, if you specify more than one keyword, only the first one will be used.	`network_interface=none`
	`network_interface=le0`

SPARC: Example sysidcfg File

The following example is a sysidcfg file for a group of SPARC based systems. (The host names, IP addresses, and netmask of these systems have been preconfigured by editing the name service.) Because all the system configuration information has been preconfigured, an automated installation can be created by using a custom JumpStart profile.

```
system_locale=en_US
timezone=US/Central
terminal=sun-cmd
timeserver=localhost
name_service=NIS {domain_name=marquee.central.sun.com
                  name_server=connor(129.152.112.3)}
root_password=m4QPOWNY
system_locale=C
```

x86: Example sysidcfg File

The following example is a sysidcfg file created for a group of x86 based systems that all have the same keyboard, graphics cards, and pointing devices. The device information (keyboard, display, and pointer)was captured from running kdmconfig -d. In this example, users would see the prompt to select a language (system_locale) for displaying the rest of the Solaris installation program.

```
keyboard=ATKBD {layout=US-English}
display=ati {size=15-inch}
pointer=MS-S
timezone=US/Central
timeserver=connor
terminal=AT386
name_service=NIS {domain_name=marquee.central.sun.com
                  name_server=connor(129.152.112.3)}
root_password=URFUni9
```

▼ How to Create a `sysidcfg` Configuration File

1. **Open a text file (it must be named `sysidcfg`) using the editor of your choice.**

> **Note -** If you create multiple `sysidcfg` files, each one must be in a separate directory or diskette.

2. **Enter the `sysidcfg` keywords shown in Table 6–2 for the system configuration information you want to preconfigure.**

TABLE 6–2 `sysidcfg` Keywords

Configuration Information	Platform	Keywords	Where to Find Values/Example

Name service, domain name, name server	All	`name_service=NIS, NIS+,`
		`OTHER, NONE`
		`{domain_name=`*domain_name*
		`name_server=`*hostname*`(`*ip_address*`)}`
			`name_service=NIS`
			`{domain_name=chandy.West.Arp.COM`
			`name_server=timber(129.221.2.1)}`

Network interface, host name, IP address, netmask	All	`network_interface=NONE,`
		`PRIMARY,` *value*
		`{hostname=`*host_name*
		`ip_address=`*ip_address*
		`netmask=`*netmask*`}`
			`network_interface=le0`
			`{hostname=feron`
			`ip_address=129.222.2.1`
			`netmask=255.255.0.0}`

Root password	All	root_password=*root_password*	Encrypted from `/etc/shadow`.

TABLE 6–2 sysidcfg Keywords *(continued)*

Configuration Information	Platform	Keywords	Where to Find Values/Example
Language in which to display the install program and desktop	All	system_locale=*locale*	The /usr/lib/locale directory or Appendix E provides the valid language and locale values.
Teminal type	All	terminal=*terminal_type*	The subdirectories in the /usr/share/lib/terminfo directory provide the valid terminal values.
Time zone	All	timezone=*timezone*	The directories and files in the /usr/share/lib/zoneinfo directory provide the valid time zone values. The time zone value is the pathname relative to the /usr/share/lib/zoneinfo directory. For example, the time zone value for the Mountain Standard Time in the United States is US/Mountain; the time zone value for Japan is Japan.
Time and date	All	timeserver=localhost, *hostname*, *ip_address*	If you specify localhost as the time server, the system's time is assumed to be correct. If you specify the *hostname* or *ip_address* (if you are not running a name service) of a system, that system's time is used to set the time.
Monitor type	x86	monitor=*monitor_type*	Run kdmconfig -d *filename*; append output to sysidcfg file.
Keyboard language, keyboard layout	x86	keyboard=*keyboard_language* {layout=*value*}	Run kdmconfig -d *filename*; append output to sysidcfg file.

TABLE 6–2 sysidcfg Keywords *(continued)*

Configuration Information	Platform	Keywords	Where to Find Values/Example

Graphics card, color depth, display resolution, screen size	x86	display=*graphics_card*
		{size=*screen_size*
		depth=*color_depth*
		resolution=*screen_resolution*}
			Run kdmconfig -d *filename*; append output to sysidcfg file.

Pointing device, number of buttons, IRQ level	x86	pointer=*pointing_device*
		{nbuttons=*number_buttons*
		irq=*value*}
			Run kdmconfig -d *filename*; append output to sysidcfg file.

3. **Save the** sysidcfg **file.**

4. **Make the** sysidcfg **file available to clients through:**
   - A shared NFS network directory (add_install_client command using the –p option)
   - The root directory on a PCFS or UFS diskette

Preconfiguring With the Name Service

Preconfiguring system configuration information by editing the name service (NIS or NIS+) is recommended for SPARC based systems. The following table gives a high-level overview of what you need to do.

To Preconfigure ...	You Must Edit and Populate These Name Service Databases...
Host name and IP address	hosts
Date and time	hosts (specify the timehost alias next to the host name of the system that will provide the date and time for the systems being installed)

To Preconfigure ...	You Must Edit and Populate These Name Service Databases...
Time zone	`timezone`
Netmask	`netmasks`

The procedure to preconfigure the language or locale for a system is different for each name service. See “How to Preconfigure Language or Locale Using NIS” on page 52 or “Preconfigure Language or Locale Using NIS+” on page 53.

▼ How to Preconfigure Language or Locale Using NIS

1. **As root on the name server, edit the** `/var/yp/Makefile` **file.**

   Add the following text after the other `*.time` entries.

```
locale.time:  $(DIR)/locale
        -@if [ -f $(DIR)/locale ]; then \
               sed -e "/^#/d" -e s/#.*$$// $(DIR)/locale \
               | awk '{for (i = 2; i<=NF; i++) print $$i, $$0}' \
               | $(MAKEDBM) - $(YPDBDIR)/$(DOM)/locale.byname; \
               touch locale.time; \
               echo "updated locale"; \
               if [ ! $(NOPUSH) ]; then \
                       $(YPPUSH) locale.byname; \
                       echo "pushed locale"; \
               else \
               : ; \
               fi \
        else \
               echo "couldn't find $(DIR)/locale"; \
        fi
```

2. **Add** `locale` **to the line starting with the word** `all` **and add** `locale: locale.time` **on a new line.**

```
all: passwd group host ethers networks rpc services protocols netgroup
bootparams aliases \
timezone locale
locale: locale.time
```

3. **Create the file** `/etc/locale` **and make one entry for each domain or a specific system.**

```
locale      domain_name
```

or

```
locale      system_name
```

For example, the following line specifies French to be the default language for the `worknet.com` domain:

```
fr worknet.com
```

For example, the following line specifies French Belgium to be the default locale for a system named `sherlock`:

```
fr_BE sherlock
```

See Appendix E for a list of valid language and locale values.

Note - Not all languages or locales are available on all Solaris CDs. The language or locale you select will be used for installation if it is on the Solaris CD.

---

4. **Make the maps.**

```
# cd /var/yp; make
```

Systems specified by domain or individually in the `locale` map are now set up to use the default language or locale. The default language or locale you've specified will be used during the installation and for the desktop after the system is rebooted.

Preconfigure Language or Locale Using NIS+

This procedure assumes the NIS+ domain is set up. Setting up the NIS+ domain is documented in the *Solaris Naming Administration Guide.*

1. **Log in to a name server as root or as a user in the NIS+ admin group.**

2. **Use the following `nistbladm` command to create the `locale` table.**

```
# nistbladm -D access=og=rmcd,nw=r -c locale_tbl name=SI,nogw= locale=,nogw=
comment=,nogw= locale.org_dir.'nisdefaults -d'
```

3. **Add an entry to the `locale` table by typing the following `nistbladm` command.**

```
# nistbladm -a name=domain_name locale=locale comment=comment
locale.org_dir.'nisdefaults -d'
```

domain_name — Is either the domain name or a specific system name for which you want to preconfigure a default language or locale.

locale — Is the language or locale you want installed on the system and for displaying the desktop after the system is rebooted. See Appendix E for a list of valid language and locale values.

comment — Is the comment field. Use double quotation marks to begin and end comments that are longer than one word.

Note - Not all languages or locales are available on all Solaris CDs. The language or locale you select will be used for installation if it is on the Solaris CD.

Systems specified by domain or individually in the `locale` table are now set up to use the default language or locale. The default language or locale you've specified will be used during the installation and for the desktop after the system is rebooted.

Preconfiguring Power Management Information

The Power Management software provided by Solaris can be used to automatically save the state of a system and power it off after it is idle for 30 minutes. When you install the Solaris software on SPARC based systems of the sun4u platform group (and any other systems that are compliant with Version 2 of the EPA's Energy Star guidelines), the Power Management software is installed by default, and you are prompted after the subsequent reboot to enable or disable the Power Management software.

If you are performing interactive installations, there is no way to preconfigure the Power Management Information and avoid the prompt. However, with custom JumpStart installations, you can preconfigure the Power Management information by using a finish script to create an `/autoshutdown` or `/noautoshutdown` file on the system. When the system reboots, the `/autoshutdown` file enables Power Management and the `/noautoshutdown` file disables Power Management.

For example, the following line in a finish script enables the Power Management software and avoids the prompt after the system reboots.

```
touch /a/autoshutdown
```

For more information about finish scripts, see "Creating Finish Scripts" on page 127.

CHAPTER 7

Preparing to Install Solaris Software Over the Network

The typical way to install the Solaris software on a system is to use a system's CD-ROM drive. However, if you have systems on a network, you can also install Solaris software on systems over the network instead of using the systems' CD-ROM drives.

Network installations enable you to install the Solaris software from a system that has access to a Solaris CD image, called an install server, to other systems on the network. An install server can either have the Solaris CD image in its CD-ROM drive, or you can copy the Solaris CD to the install server's hard disk. Network installations that use a Solaris CD image copied on an install server's hard disk are usually faster than installations from a CD-ROM drive.

Task Map: Preparing to Install Solaris Software Over the Network

TABLE 7–1 Task Map: Preparing to Install Solaris Over the Network

Task	Description	For Instructions, Go To
Create an Install Server	You can create an install server by copying the Solaris CD to the server's hard disk (`setup_install_server` command) or by mounting the Solaris CD from its CD-ROM drive.	"How to Create an Install Server" on page 62
Create Boot Servers	If you want to install systems over the network that are not on the same subnet as the install server, you must create a boot server on the subnet to boot the systems. Use the `setup_install_server -b` command to create a boot server.	"How to Create a Boot Server on a Subnet " on page 64
Set Up Systems to Be Installed Over the Network	**Using Solstice Host Manager** Host Manager provides an easy-to-use, graphical user interface that enables you to add network installation information about a system to the name service, so the system can be installed over the network. You should use this tool if you want to use the NIS or NIS+ name service to store the network installation information.	"How to Set Up Systems to Be Installed Over the Network With Host Manager" on page 66
	Using the `add_install_client` Command The `add_install_client` command provides a command-line way to add network installation information about a system to an install or boot server's `/etc` files, so the system can install over the network.	"How to Set Up Systems to be Installed Over the Network With `add_install_client`" on page 68

Servers Required for Network Installation

As shown in Figure 7–1, systems that install Solaris software over the network require:

- **Install server –** A networked system that provides a Solaris CD image for systems on the network to install from. You can create an install server by copying the Solaris CD to the server's hard disk or by mounting the Solaris CD from its CD-ROM drive.

  By copying Solaris CDs, you enable a single install server to provide Solaris CD images for multiple releases, including Solaris CD images for different platforms. For example, a SPARC install server could provide the Solaris 2.5.1 and 2.6 CD images for SPARC based systems, and the same SPARC install server could also provide the Solaris 2.6 CD image for x86 based systems.

- **Name server –** A system that manages a distributed network database (such as NIS or NIS+) containing information about users and other systems on the network.

Note - The install server and name server may be the same or separate systems.

- **Boot server –** A system used to boot the system to be installed over the network. A boot server and install server are typically the same system. However, if the system to be installed is on a *different* subnet than the install server, a boot server is required on that subnet.

  A single boot server can provide Solaris boot software for multiple releases, including the Solaris boot software for different platforms. For example, a SPARC boot server could provide the Solaris 2.5.1 and 2.6 boot software for SPARC based systems, and the same SPARC boot server could also provide the Solaris 2.6 boot software for x86 based systems.

- **OS server –** A system that provides Solaris operating environment software including services and/or file systems.

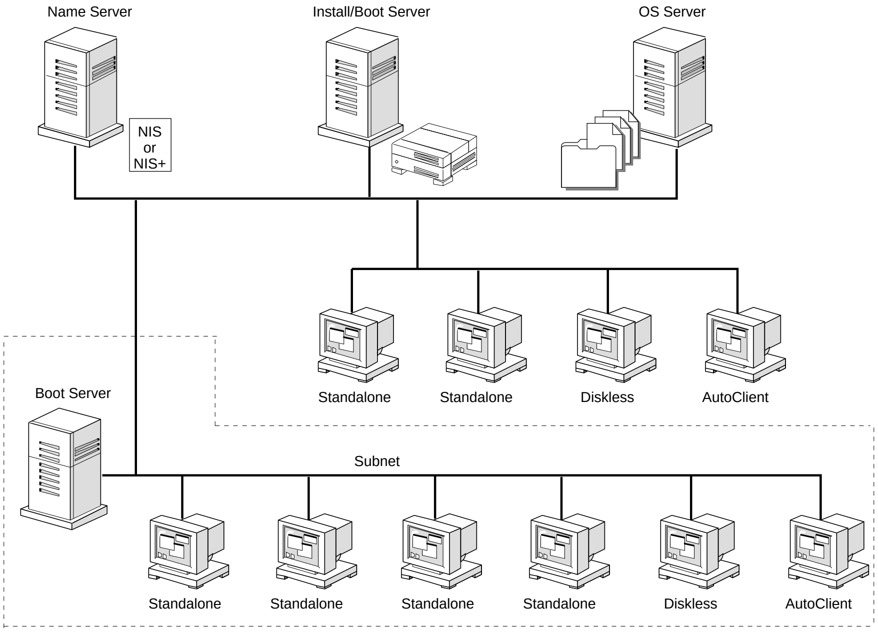


Figure 7–1 Network Installation Servers

Network Installation Commands

Table 7–2 lists the commands that you need to use for setting up network installations.

TABLE 7–2 Network Installation Commands

Command	Platform	Description
`add_install_client`	All	A command that adds network installation information about a system to an install or boot server's `/etc` files, so the system can install over the network.
`setup_install_server`	All	A script that copies the Solaris CD to an install server's local disk or copies the boot software to a boot server. See the `setup_install_server(1M)` man page for more information.

TABLE 7–2 Network Installation Commands *(continued)*

Command	**Platform**	**Description**
Host Manager	All	A graphical user interface that is available from within Solstice AdminSuite (`solstice`). You can use it to add network installation information about a system to the name service, so the system can install over the network.
`mount`	All	A command that shows mounted file systems, including the Solaris CD file system. See the `mount(1M)` page for more information.
`uname -i`	All	A command for determining a system's platform name (for example, `SUNW,SPARCstation-5`). This information is sometimes required during installation. See the `uname(1)` man page for more information.
`patchadd -C`	All	A command to patch a Solaris CD image on an install server's hard disk, so clients can install Solaris software that's already patched. See the `patchadd(1M)` man page for more information.
`reset`	SPARC	A command for resetting the terminal settings and display. It is sometimes useful to use `reset` before booting. Or, if you boot and see a series of error messages about I/O interrupts, press the L1 or STOP and A keys at the same time, and then enter `reset` at the `ok` or > PROM prompt.
`banner`	SPARC	A command for displaying system information, such as model name, Ethernet address, or memory installed. Available only from the `ok` or > PROM prompt.

Creating an Install Server and Boot Servers

You must create an install server, and possibly a boot server, to install the Solaris software on a system over the network. This section describes how to:

- Create an install server by copying the Solaris CD to the server's hard disk or by mounting the Solaris CD from its CD-ROM drive.

Note - If you intend to do frequent installations over the network, you should copy the Solaris CD to an install server's hard disk. Network installations from an install server's hard disk are faster than from its CD-ROM drive, and using the hard disk also frees the CD-ROM drive for other uses.

Note - You cannot use a SunOS 4.1.x system as an install server for Solaris-based systems.

- Create separate boot servers (required *only* if systems are not on same subnet as the install server) for each subnet. Instead of creating separate boot servers, you can create an install server for each subnet; however, this requires more disk space or CD-ROM resources.

▼ How to Create an Install Server

1. **On the system that is going to be the install server, log in and become root.**

   This system must have a CD-ROM drive and be part of the site's network and name service. The system must also be in the NIS or NIS+ name service. (If your site doesn't use the NIS or NIS+ name service, you must distribute information about this system by following your site's policies.)

   Note - This procedure assumes that the system is running Volume Management. For detailed information about managing CDs without Volume Management, see the *System Administration Guide.*

2. **Insert the Solaris CD into the CD-ROM drive.**

   This is the Solaris CD that you want to provide to the systems over the network.

3. **Mount the Solaris CD (if needed).**

   Volume management automatically mounts the Solaris CD on `/cdrom/cdrom0/s0` or `/cdrom/cdrom0/s2`.

4. **Determine your next step based on whether or not you want to copy the Solaris CD to the install server's hard disk.**

If You ...	Then ...
Want to copy the Solaris CD	Go to Step 5 on page 63.
Do not want to copy the Solaris CD	Go to Step 7 on page 63.

5. **Change directory to the `Tools` directory on the mounted Solaris CD.**

```
# cd Solaris_2.6/Tools
```

6. **Copy the Solaris CD to the install server's hard disk by using the `setup_install_server` command.**

```
# ./setup_install_server install_dir_path
```

install_dir_path   Specifies the directory where the Solaris CD will be copied. The directory must be empty.

Note - The `setup_install_server` command will indicate if you do not have enough disk space for the Solaris CD. Use the `df -kl` command to determine available disk space.

Note - After copying the Solaris CD, you can use the `patchadd -c` command to patch the Solaris CD image on the install server's hard disk. So, instead of patching every client after it installs, clients can install Solaris software that is already patched.

---

7. **Determine your next step based on whether or not the install server is on the same subnet as the system to be installed.**

If Install Server Is ...	Then ...
On same subnet as the system to be installed	You don't need to create a boot server. Go to "Setting Up Systems to Be Installed Over the Network" on page 66.
Not on the same subnet as the system to be installed	Follow the steps in "How to Create a Boot Server on a Subnet " on page 64. You must complete this procedure when the install server is *not* on the same subnet as the system to be installed.

Example–Creating an Install Server

The following example creates an install server by copying the Solaris CD to the install server’s /export/install directory.

```
# cd /cdrom/cdrom0/s0/Solaris_2.6/Tools
# ./setup_install_server /export/install
```

Where to Go Next

The install server (and boot server, if needed) is now created. To set up systems to be installed from the install server, go to “Setting Up Systems to Be Installed Over the Network” on page 66.

▼ How to Create a Boot Server on a Subnet

You can install Solaris software over the network from any install server on the network. However, a system that needs to use an install server on another subnet *requires* a separate boot server on its own subnet. A boot server contains enough of the boot software to boot systems over the network, and then the install server takes over to install the Solaris software.

1. **On the system that will be the boot server for the subnet, log in and become root.**

   This system must have a local CD-ROM drive or have access to a remote Solaris CD image. The system must also be in the NIS or NIS+ name service. (If your site doesn’t use the NIS or NIS+ name service, you must distribute information about this system by following your site’s policies.)

   Note - This procedure assumes that the system is running Volume Management. For detailed information about managing CDs without Volume Management, see the *System Administration Guide.*

   ---

2. **Determine your next step based on where you want to access the Solaris CD image from.**

   You must use the Solaris CD image of the same release and platform type that is being used to install the systems.

If You Want To...	**Then ...**
Mount the Solaris CD on the boot server's CD-ROM drive	1. Insert the Solaris CD into the CD-ROM drive. 2. Mount the Solaris CD (if needed). **Note -** Volume management automatically mounts the Solaris CD on `/cdrom/cdrom0/s0` or `/cdrom/cdrom0/s2`.
NFS-mount a Solaris CD image from a remote install server	1. Mount the Solaris CD image. # **`mount -F nfs -o ro`** *server_name*`:`*path* **`/mnt`** where *server_name*`:`*path* is the host name and absolute path to the Solaris CD image. 2. Change directory to the mounted Solaris CD image. # **`cd /mnt`**

3. **Change directory to the `Tools` directory on the Solaris CD image.**

```
# cd Solaris_2.6/Tools
```

4. **Copy the boot software to the boot server by using the `setup_install_server` command.**

```
# ./setup_install_server -b boot_dir_path
```

`–b` Specifies that the system will be set up as a boot server.

boot_dir_path Specifies the directory where the boot software will be copied. The directory must be empty.

Note - The `setup_install_server` command will indicate if you do not have enough disk space to copy the platform dependent information. Use the `df -kl` command to determine available disk space.

Example–Creating a Boot Server on a Subnet

The following example creates a boot server on a subnet by copying the boot software from the Solaris CD image to `/export/install/boot` on the system's local disk.

```
# cd /cdrom/cdrom0/s0/Solaris_2.6/Tools
# ./setup_install_server -b /export/install/boot
```

Where to Go Next

The boot server is now set up to boot supported systems on a subnet. To continue, go to “Setting Up Systems to Be Installed Over the Network” on page 66.

Setting Up Systems to Be Installed Over the Network

After you’ve created an install server and possibly a boot server, you are ready to install the Solaris software on other systems over the network. However, to be installed over the network, a system needs to know where it is going to:

- Install from (install server)
- Boot from (install server or boot server)
- Find its profile during a custom JumpStart installation (JumpStart directory on profile server)

Because a system looks for this information in the name service (`bootparams` database in the `/etc` files, NIS, or NIS+) when it installs over the network, you must add this information into the name service for every system that is going to be installed over the network. You can do this by using Solstice Host Manager or the `add_install_client` command.

Note - If you use the `/etc` files to store network installation information, the information must reside on the install server or the boot server (if a boot server is required).

▼ How to Set Up Systems to Be Installed Over the Network With Host Manager

You should use Host Manager if you want:

- A centralized way to set up systems to be installed over the network
- An easy-to-use graphical user interface
- To use the NIS or NIS+ name service to store network installation information

Note - If you want to preconfigure system configuration information by using a remote `sysidcfg` file, you must use the `add_install_client` command to set up systems to be installed over the network.

This procedure assumes that you have already installed Host Manager at your site and have the required permissions to use it. For detailed information about setting up Host Manager, refer to the Host Manager documentation.

1. **Log in to a system that has access to Host Manager.**

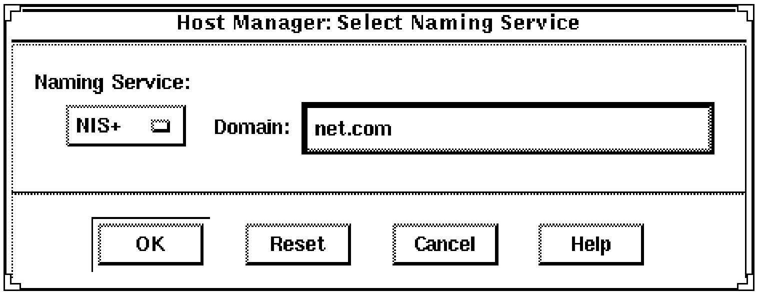
2. **Start the Solstice AdminSuite with the following command.**

```
$ /usr/bin/solstice &
```

3. **After the Solstice AdminSuite main window appears, click on the Host Manager icon.**

4. **On the Host Manager: Select Naming Service screen, select a naming service and click on the Apply button. See the sample Naming Service screen below.**

If the Name Service Is ...	Then Select ...
NIS+	NIS+. Host Manager will update the NIS+ tables.
NIS	NIS. Host Manager will update the NIS maps.
None	None. Host Manager will store the information you provide in the `/etc` files on the appropriate systems. This will provide enough information for systems to boot and to install Solaris software over the network.



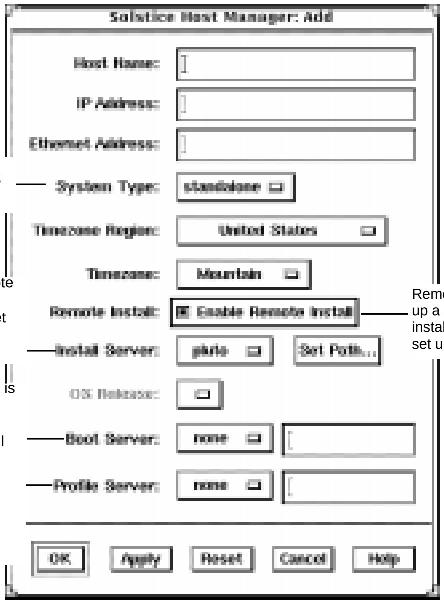
Host Manager: Select Naming Service

Naming Service: NIS+

Domain: net.com

5. **On the Host Manager main window, choose Add... from the Edit menu.**

6. **On the Host Manager: Add screen, complete all fields and click on the OK or Apply button.**



Solstice Host Manager: Add

Host Name:

IP Address:

Ethernet Address:

System Type: standalone

Timezone Region: United States

Timezone: Mountain

Remote Install: ☒ Enable Remote Install

Install Server: pluto

OS Release:

Boot Server: none

Profile Server: none

Type - Select standalone or OS server.

Install Server - If enabling remote install for a network installation, specify the install server and set the path to the location of the Solaris CD image.

Boot Server - If the install client is on a different subnet than the install server, specify the boot server that resides on the install client's subnet.

Profile Server - If using custom JumpStart installations, specify the system with the custom Jumpstart profiles on it.

Remote Install - Select if setting up a network installation. The install server should have been set up already

7. **On the Host Manager main window, choose Save Changes from the File menu.**

Where to Go Next

The system is now ready to be installed over the network. To install the system, see Chapter 3.

▼ How to Set Up Systems to be Installed Over the Network With `add_install_client`

If Host Manager is not available, you can use the `add_install_client` command to set up systems to be installed over the network. The `add_install_client` updates only the `/etc` files.

1. **Become root on the install server (or the boot server if a system requires one).**

2. **Make sure the following information about the system to be installed has been added to the name service (`/etc` files, NIS, or NIS+).**

- Host name
- IP address
- Ethernet address

3. **Change directory to the** `Tools` **directory on the install server's Solaris CD image or the boot server's boot directory.**

```
# cd Solaris_2.6/Tools
```

4. **Set up a system to be installed over the network with the** `add_install_client` **command.**

```
# ./add_install_client [-c server:jumpstart_dir_path]
  [-s install_server:install_path] [-p server:path] host_name platform_group
```

–c *server*:*jumpstart_dir_path*

Specifies a JumpStart directory for custom JumpStart installations. This option and its arguments are required only for custom JumpStart installations.

server is the host name of the server on which the JumpStart directory is located. *jumpstart_dir_path* is the absolute path to the JumpStart directory.

–s *install_server*:*install_dir_path*

Specifies the install server. This option is required only when you are using `add_install_client` on a boot server.

install_server is the host name of the install server. *install_dir_path* is the absolute path to the Solaris CD image.

–p *server*:*path*

Specifies the `sysidcfg` file for preconfiguring system information. *server* is either a valid host name or IP address for the server that contains the file and *path* is the absolute path to the `sysidcfg` file.

host_name

Is the host name of system to be installed over the network. (This is not the host name of the install server). The host must be in the name service for this command to work.

platform group

Is the platform group of the system to be installed. (For a detailed list of platform groups, see Appendix C.)

SPARC: Example–Adding Systems to Be Installed Over the Network With `add_install_client`

The following example adds a system named `basil`, which is a SPARCstation 10, to be installed over the network. The system requires a boot server, so the command is run on the boot server; the `–s` option is used to specify the install server named `install_server1`, which contains a Solaris CD image in `/export/install`.

```
# cd /export/boot/Solaris_2.6/Tools
# ./add_install_client -s install_server1:/export/install basil sun4m
```

Where to Go Next

The system is now ready to be installed over the network. To install the system, see Chapter 3.

CHAPTER **8**

Preparing Custom JumpStart Installations

This chapter provides the step-by-step instructions on how to prepare your site to perform custom JumpStart installations.

Note - Appendix D provides a detailed example of how you would prepare a fictitious site for custom JumpStart installations.

Overview

The custom JumpStart installation method provides a way to automatically and identically install groups of systems. The first step when preparing custom JumpStart installations is deciding how you want the systems at your site to be installed. For example, the following scenario would be ideal to set up and perform custom JumpStart installations:

- You need to install the Solaris software on 100 new systems.
- The engineering group owns 70 out of the 100 new, SPARC based systems, and its systems must be installed as standalone systems with the developer software group.
- The marketing group owns 30 out of the 100 new, x86 based systems, and its systems must be installed as standalone systems with the end user software group.

After you decide how you want the systems at your site to be installed, the most important step when preparing custom JumpStart installations is to create the essential files that are used during a custom JumpStart installation: the rules.ok file (a validated rules file) and a profile for each group of systems. The rules file is a text file that should contain a rule for each group of systems (or single systems) that you want to automatically install. Each rule distinguishes a group of systems based on one or more system attributes, and it links each group to a profile, which is a text file that defines how the Solaris software will be installed on each system in the group. Both the rules file and the profiles must be located in a JumpStart directory.

In the previous scenario, you would create a rules file with two different rules, one rule for the engineering and another rule for the marketing group. For each rule, you could use the system's platform groups to distinguish the groups from one another: the engineering group has SPARC based systems and the marketing group has x86 based systems. Each rule would also contain a link to an appropriate profile. For example, in the rule for the engineering group, you would add a link to the profile, called eng_profile, that you created for the engineering group. And, in the rule for the marketing group, you would add a link to the profile, called market_profile, that you created for the engineering group.

After creating the rules file and profiles, you have to validate them with the check script. If the check script runs successfully, the rules.ok file is created, which is a generated version of the rules file that the Solaris installation program uses to perform custom JumpStart installations.

What Happens During a Custom JumpStart Installation

During a custom JumpStart installation, the Solaris installation program reads the rules.ok file and tries to find the first rule whose defined system attributes match the system that's installing. If a match occurs, the installation program uses the profile specified in the rule to automatically install the system.

Figure 8–1 is an example of how a custom JumpStart installation works on a standalone, non-networked system using a diskette in the system's diskette drive.

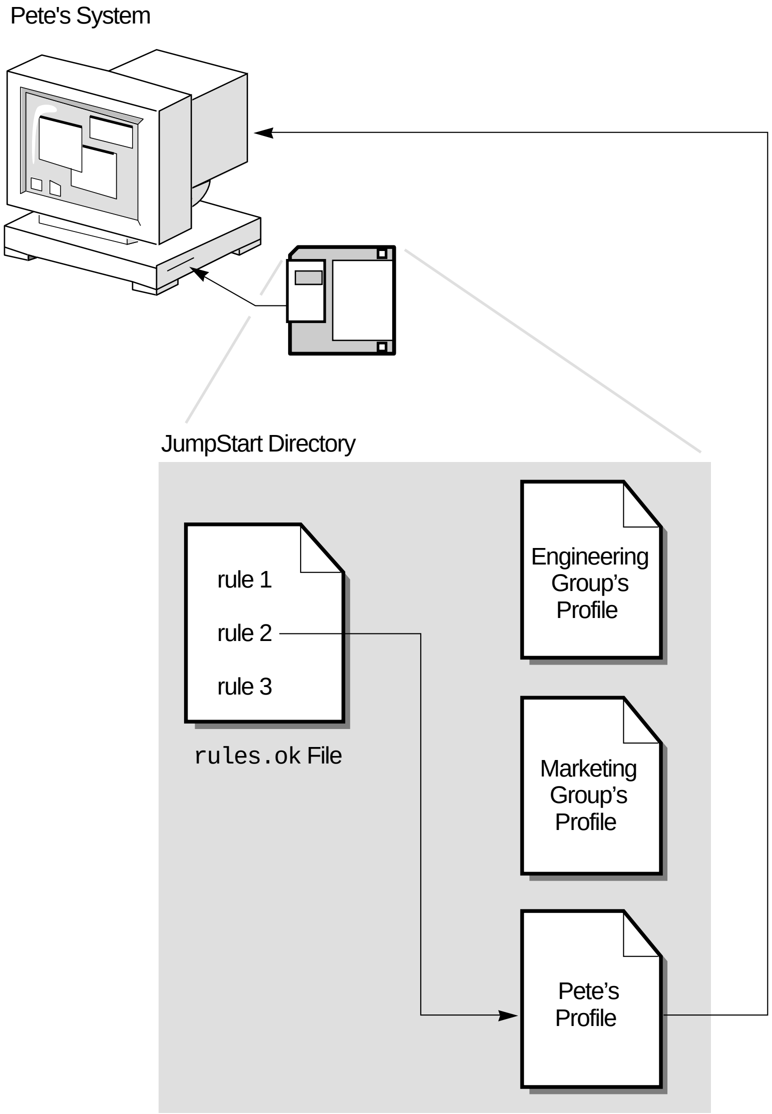


Figure 8–1 How a Custom JumpStart Installation Works: Non-Networked Example

Figure 8–2 is an example of how a custom JumpStart installation works for multiple systems on a network where different profiles are accessed from a single server.

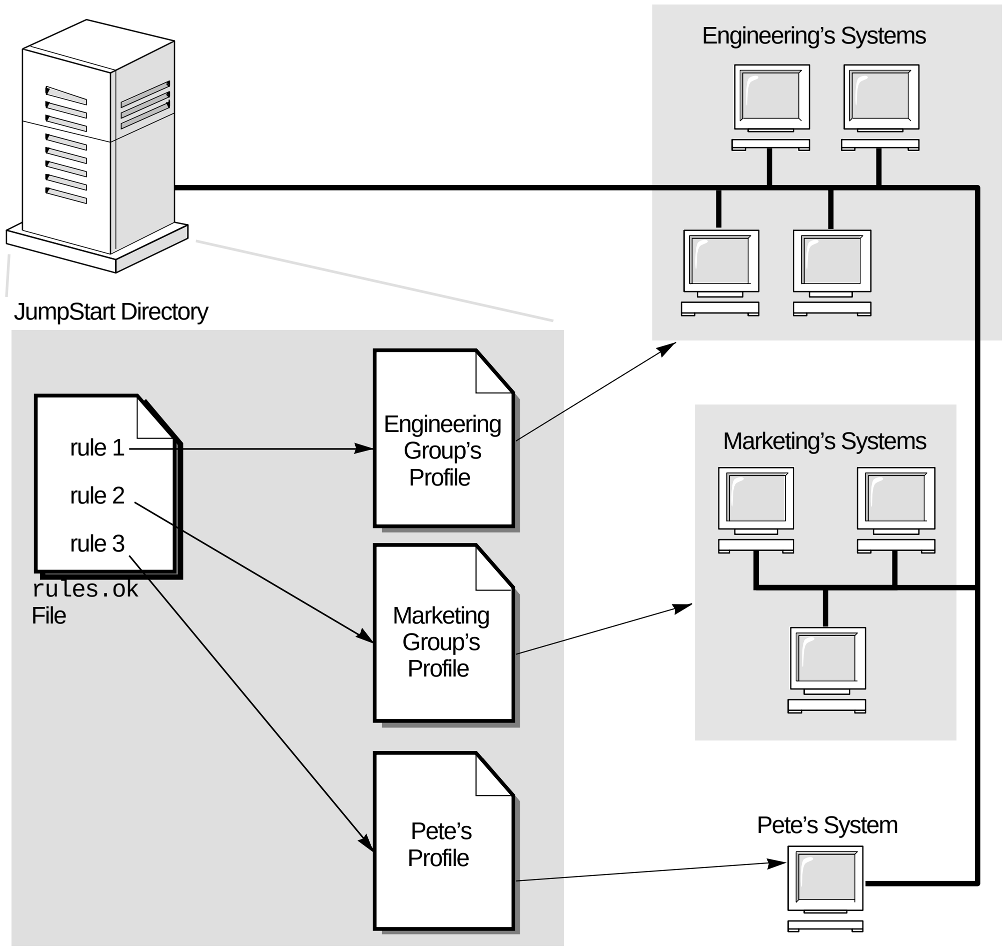


Figure 8–2 How a Custom JumpStart Installation Works: Networked Example

As shown in Figure 8–1 and Figure 8–2, the custom JumpStart files that you need to set up can be located on either a diskette or server (called a profile diskette and profile server, respectively). A profile diskette is required when you want to perform custom JumpStart installations on a non-networked, standalone systems. A profile server should be used when you want to perform custom JumpStart installations on networked systems that have access to the server.

Figure 8–3 describes what happens on a system during a custom JumpStart installation and shows the search order that the Solaris installation program uses to find the custom JumpStart files.

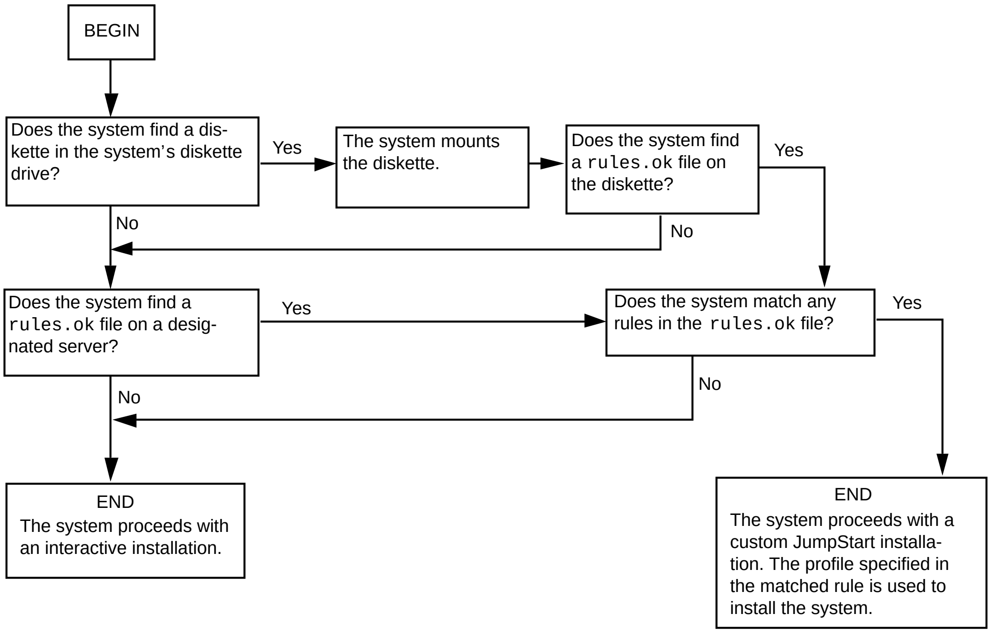

BEGIN
Does the system find a diskette in the system's diskette drive?
Yes
The system mounts the diskette.
Does the system find a rules.ok file on the diskette?
Yes
No
No
Does the system find a rules.ok file on a designated server?
Yes
Does the system match any rules in the rules.ok file?
Yes
No
No
END
The system proceeds with an interactive installation.
END
The system proceeds with a custom JumpStart installation. The profile specified in the matched rule is used to install the system.

Figure 8–3 What Happens During a Custom JumpStart Installation

Task Map: Preparing Custom JumpStart Installations

TABLE 8–1 Task Map: Preparing Custom JumpStart Installations

Task	Description	For Instructions, Go To
Create a JumpStart Directory	**On a Diskette** If you want to perform custom JumpStart installations on systems that are not connected to a network, you must create a profile diskette, which is a diskette that contains the custom JumpStart files.	"Creating a Profile Diskette " on page 81
	On a Server If you want to perform custom JumpStart installations on systems connected to a network, you should create a profile server, which is a server that contains a JumpStart directory for the custom JumpStart files.	"Creating a Profile Server" on page 77
Enable All Systems to Access the Profile Server	Optional. When you use a profile server, you can enable all systems at once to access the profile server. By doing this, you don't have to individually enable every system to access the profiles on the profile server.	"Enabling All Systems to Access the Profile Server" on page 80
Add Rules to the `rules` File	After you decide how you want each group of systems (or single systems) at your site to be installed, you have to create a rule for each specific group that you want to install. Each rule distinguishes a group based on one or more system attributes, and it links each group to a profile.	"Creating the `rules` File" on page 87
Create a Profile for Every Rule	A profile is a text file that defines how to install the Solaris software (for example, which software group to install) on a system. Every rule specifies a profile to define how a system will be installed when the rule is matched. You usually create a different profile for every rule; however, the same profile can be used in more than one rule.	"Creating a Profile" on page 95

TABLE 8–1 Task Map: Preparing Custom JumpStart Installations *(continued)*

Task	Description	For Instructions, Go To
Test the Profiles	Optional. After you create a profile, you should use the `pfinstall(1M)` command to test the profile before using it to install or upgrade a system (called a "dry run" installation).	"Testing a Profile" on page 118
Validate the `rules` File	The `rules.ok` file is a generated version of the `rules` file that the Solaris installation program uses to match the system to be installed with a profile. You must use the `check` script to validate the `rules.ok` file.	"Validating the `rules` File" on page 122

Creating a Profile Server

When setting up custom JumpStart installations for systems on the network, you have to create a directory on a server (called a JumpStart directory). A JumpStart directory contains all the essential custom JumpStart files (for example, the `rules` file, `rules.ok` file, and profiles) at its root level.

The server that contains a JumpStart directory is called a *profile server*. The profile server can be the same system as either the install or boot server, or it can be a completely different server. The JumpStart directory should be owned by root and have permissions equal to 755.

Note - A profile server can provide custom JumpStart files for systems with the same or different platform type as the server. For example, a SPARC server can provide custom JumpStart files for both SPARC and x86 based systems.

▼ How to Create a JumpStart Directory on a Server

Note - This procedure assumes that the system is running Volume Management. If you are not using Volume Management to manage diskettes and CDs, refer to the *System Administration Guide* for detailed information about managing removable media without Volume Management.

1. **Log in as root on the server where you want the JumpStart directory to reside.**

2. **Create the JumpStart directory anywhere on the server.**

   ```
   # mkdir jumpstart_dir_path
   ```

   jumpstart_dir_path Is the absolute path of the JumpStart directory.

   For example, the following command would create the directory called `jumpstart` in the root file system:

   ```
   mkdir /jumpstart
   ```

3. **Edit the** `/etc/dfs/dfstab` **file. Add the following entry.**

   ```
   share -F nfs -o ro,anon=0 jumpstart_dir_path
   ```

   For example, the following entry would share the `/jumpstart` directory:

   ```
   share -F nfs -o ro,anon=0 /jumpstart
   ```

4. **Type** `shareall` **and press Return.**

5. **Determine your next step based on where the Solaris CD is located.**

   You only need to perform the rest of the steps if you want to copy example custom JumpStart files from the Solaris CD. You are already done creating the profile server.

If You Want to Use The ...	Then ...
Solaris CD in the local CD-ROM drive	1. Insert the Solaris CD into the CD-ROM drive. 2. Mount the Solaris CD (if needed). **Note -** Volume management automatically mounts the Solaris CD on `/cdrom/cdrom0/s0` or `/cdrom/cdrom0/s2`.
Solaris CD image on local disk	Change the directory to the Solaris CD image on the local disk. For example: `cd /export/install`

6. **Change directory to the `Misc` directory on the Solaris CD image.**

```
# cd Solaris_2.6/Misc
```

7. **Copy the example custom JumpStart files into the JumpStart directory on the profile server.**

```
# cp -r jumpstart_sample/* jumpstart_dir_path
```

For example, the following command would copy the `jumpstart_sample` directory into the `/jumpstart` directory:

```
cp -r jumpstart_sample/* /jumpstart
```

The files you just copied are only example custom JumpStart files. You must update the files for your own site.

Where to Go Next

You have completed creating a profile server. To continue, see “Enabling All Systems to Access the Profile Server” on page 80.

Enabling All Systems to Access the Profile Server

When you create a profile server, you must make sure systems can access it during a custom JumpStart installation. There are two ways to do this:

- Every time you add a system for network installation, you have to use the `–c` option of the `add_install_client` command or specify the profile server in Host Manager.

  or

- Enable all systems to access the profile server by using a wildcard in the `/etc/bootparams` file.

To save you time when adding systems for network installations, use the following procedure to enable all systems to access the profile server. Otherwise, see "Creating the `rules` File" on page 87.

▼ How to Enable All Systems to Access the Profile Server

This procedure is valid only if you are using the `/etc/bootparams` file to store network installation information. If you are using the NIS or NIS+ `bootparams` database for network installation information, you need to update the `bootparams` database with the entry in Step 2 on page 80.

Note - This procedure is not necessary if you are using a diskette for the JumpStart directory.

---

1. **On the install server or boot server, log in as root.**

2. **Edit the `/etc/bootparams` file. Add the following entry.**

```
* install_config=server:jumpstart_dir_path
```

* Is a wildcard character specifying all systems.

server Is the host name of the profile server where the JumpStart directory is located.

jumpstart_dir_path Is the absolute path of the JumpStart directory.

For example, the following entry would enable all systems to access the `/jumpstart` directory on the profile server named `sherlock`:

```
* install_config=sherlock:/jumpstart
```

Caution - Using this procedure may produce the following error message when an install client is booted:
`WARNING: getfile: RPC failed: error 5: (RPC Timed out).` See "Booting a System Over the Network" on page 36 for more details on this error message.

Where to Go Next

All systems can now access the profile server. You no longer need to specify the profile server in Host Manager or use the `–c` option of the `add_install_client` command when adding systems for network installations. To continue, go to "Creating the `rules` File" on page 87.

Creating a Profile Diskette

You must create a JumpStart directory on a diskette if a system is *not* connected to a network, because the system won't have access to a profile server. However, as expected, the system must have a diskette drive.

When you use a diskette for custom JumpStart installations, the essential custom JumpStart files (for example, the `rules` file, `rules.ok` file, and profiles) must reside in the root directory (JumpStart directory) on the diskette. The diskette that contains a JumpStart directory is called a *profile diskette*. The custom JumpStart files on the diskette should be owned by root and have permissions equal to 755.

The diskette requirements for the profile diskette are different for x86 based systems and SPARC based systems, so there is a different procedure to create a profile diskette for each platform.

▼ x86: How to Create a Profile Diskette

Follow this procedure to create a profile diskette for x86 based systems, which involves:

- Making a copy of the Configuration Assistant diskette (the copied boot diskette has a PCFS file system)
- Copying sample custom JumpStart installation files into the diskette's root directory (JumpStart directory)

Note - This procedure assumes that the system is running Volume Management. For detailed information about managing CDs without Volume Management, see the *System Administration Guide*.

---

1. **Log in as root on an x86 or SPARC based system that has a diskette drive.**

2. **Insert the Configuration Assistant diskette into the diskette drive.**

3. **Make sure Volume Management knows about the diskette.**

```
# volcheck
```

4. **Copy the Configuration Assistant diskette image to the system's hard disk.**

```
# dd if=/vol/dev/aliases/floppy0 of=boot_image
```

boot_image Is the file name where the Configuration Assistant diskette image is copied. You can specify an absolute path name.

For example, the following command would copy the boot diskette to the `boot_save` file.

```
dd if=/vol/dev/aliases/floppy0 of=boot_save
```

5. **Manually eject the Configuration Assistant diskette.**

6. **Find a blank diskette (or a diskette that can be overwritten) that you can use for a profile diskette and insert it into the diskette drive.**

   Any previous information on the diskette will be overwritten when you make it into a profile diskette.

7. **Make sure Volume Management knows about the diskette.**

```
# volcheck
```

8. **Format the diskette.**

Caution - This step will overwrite any data on the diskette.

```
# fdformat -d -U
```

9. **Copy the Configuration Assistant diskette image from the system's hard disk to the formatted diskette.**

```
# dd if=boot_image of=/vol/dev/aliases/floppy0
```

   The *boot_image* variable should be the same as in Step 4 on page 82.

10. **Eject the diskette.**

```
# eject floppy
```

11. **Insert the copied boot diskette back into the diskette drive.**

12. **Make sure Volume Management knows about the diskette.**

```
# volcheck
```

13. **Determine your next step based on where the Solaris CD is located.**

    You only need to perform the rest of the steps if you want to copy example custom JumpStart files from the Solaris CD. You are already done creating the profile diskette.

If You Want to Use The ...	Then ...
Solaris CD in the local CD-ROM drive	1. Insert the Solaris CD into the CD-ROM drive. 2. Mount the Solaris CD (if needed). **Note -** Volume management automatically mounts the Solaris CD on `/cdrom/cdrom0/s0` or `/cdrom/cdrom0/s2`.
Solaris CD image on local disk	Change directory to the Solaris CD image on the local disk. For example: `cd /export/install`

14. Change directory to the `Misc` directory on the Solaris CD image.

```
# cd Solaris_2.6/Misc
```

15. Copy the example custom JumpStart files into the root directory (JumpStart directory) of the profile diskette.

```
# cp -r jumpstart_sample/* /floppy/floppy0/.
```

The files you just copied are only example custom JumpStart files. You must update the files for your own site.

Caution - File names on PCFS file systems can be only 11 characters long (an 8-character file name and a 3-character extension). When copying JumpStart installation files to a diskette for x86 systems, be aware that the file transfer may truncate file names.

Note - When using a profile diskette, all the custom JumpStart installation files must be in the root directory of the diskette.

Where to Go Next

You have completed creating a profile diskette. Now you can update the `rules` file and create profiles on the profile diskette to perform custom JumpStart installations. To continue, go to “Creating the `rules` File” on page 87.

▼ SPARC: How to Create a Profile Diskette

Follow this procedure to create a profile diskette for SPARC based systems, which involves:

- Formatting a diskette (if needed).
- Creating a UFS file system on the diskette (if needed).
- Copying sample custom JumpStart installation files into the diskette's root directory (JumpStart directory).

Note - This procedure assumes that the system is running Volume Management. If you are not using Volume Management to manage diskettes and CDs, refer to the *System Administration Guide* for detailed information about managing removable media without Volume Management.

---

1. **Log in as root on a SPARC based system that has a diskette drive and a CD-ROM drive.**

2. **Find a blank diskette (or a diskette that can be overwritten) that you can use for a profile diskette and insert it into the diskette drive.**

   Any previous information on the diskette will be overwritten when you make it into a profile diskette.

3. **Make sure Volume Management knows about the diskette.**

```
# volcheck
```

4. **If the diskette already has a UFS file system on it, go to Step 10 on page 86.**

   To find out if the diskette has a UFS file system on it, check the `/etc/mnttab` file for an entry similar to this:

```
/floppy/unnamed_floppy ufs
```

5. **Format the diskette.**

Caution - This step will overwrite any data on the disk.

```
# fdformat -U
```

6. **Create a UFS file system on the diskette.**

```
# newfs /vol/dev/aliases/floppy0
```

7. **Eject the diskette.**

```
# eject floppy
```

8. **Insert the formatted diskette back into the diskette drive.**

9. **Make sure Volume Management knows about the diskette.**

```
# volcheck
```

10. **Determine your next step based on where the Solaris CD is located.**

    You only need to perform the rest of the steps if you want to copy example custom JumpStart files from the Solaris CD. You are already done creating the profile diskette.

If You Want to Use The ...	Then ...
Solaris CD in the local CD-ROM drive	1. Insert the Solaris CD into the CD-ROM drive. 2. Mount the Solaris CD (if needed). **Note -** Volume management automatically mounts the Solaris CD on `/cdrom/cdrom0/s0` or `/cdrom/cdrom0/s2`.
Solaris CD image on the local disk	Change the directory to the Solaris CD image on the local disk. For example: `cd /export/install`

11. **Change directory to the `Misc` directory on the Solaris CD image.**

```
# cd Solaris_2.6/Misc
```

12. **Copy the example custom JumpStart installation files into the root directory (JumpStart directory) of the profile diskette.**

```
# cp -r jumpstart_sample/* /floppy/floppy0/.
```

The files you just copied are only example custom JumpStart files. You must update the files for your own site.

Note - When using a profile diskette, all the custom JumpStart installation files must be in the root directory of the diskette.

Where to Go Next

You have completed creating a profile diskette. Now you can update the `rules` file and create profiles on the profile diskette to perform custom JumpStart installations. To continue, go to “Creating the `rules` File” on page 87.

Creating the `rules` File

What Is the `rules` File

The `rules` file is a text file that should contain a rule for each group of systems (or single systems) that you want to automatically install. Each rule distinguishes a group of systems based on one or more system attributes, and it links each group to a profile, which is a text file that defines how the Solaris software will be installed on each system in the group. For example, the rule

```
karch sun4c - basic_prof -
```

specifies that the Solaris installation program will automatically install any system with the sun4c platform group based on the information in the `basic_prof` profile. The `rules` file is used to create the `rules.ok` file, which is required for custom JumpStart installations.

Note - If you set up the JumpStart directory by using the procedures on “Creating a Profile Diskette ” on page 81 or “Creating a Profile Server” on page 77, an example `rules` file should already be in the JumpStart directory; the example `rules` file contains documentation and some example rules. If you use the example `rules` file, make sure you comment out the example rules that you will not use.

When Does a System Match a Rule

During a custom JumpStart installation, the Solaris installation program attempts to match the system being installed to the rules in the `rules.ok` file in order: first rule through the last rule. A rule match occurs when the system being installed matches all of the system attributes defined in the rule. As soon as a system matches a rule, the Solaris installation program stops reading the `rules.ok` file and begins to install the system based on the matched rule's profile.

Important Information About the `rules` File

The `rules` file must have:

- At least one rule.
- The file name, `rules`.
- Rules that have at least a rule keyword, a rule value, and a corresponding profile. Rules must also have a minus sign (-) in the *begin* and *finish* fields if there is no entry.

The rules file allows:

- A comment after the pound sign (#) anywhere on a line. If a line begins with a #, the entire line is a comment line. If a # is specified in the middle of a line, everything after the # is considered a comment.
- Blank lines.
- Rules to span multiple lines. You can let a rule wrap to a new line, or you can continue a rule on a new line by using a backslash (\) before the carriage return.

▼ How to Create the `rules` File

1. **Open a new text file (it must be named `rules`) using the editor of your choice.**

   You can create a new `rules` file or edit the sample `rules` file provided in the JumpStart directory you created.

2. **Add a rule in the `rules` file for each group of systems you want to install using custom JumpStart.**

   Refer to the following information as you add rules to the `rules` file:

   - "Important Information About the `rules` File" on page 88
   - "Rule Examples" on page 90
   - Table 8–2

   A rule within the `rules` file must have the following syntax:

```
[!]rule_keyword  rule_value  [&& [!]rule_keyword  rule_value]...  begin   profile   finish
```

!
Is a symbol used before a rule keyword to indicate negation.

[ ]
Is a symbol used to indicate an optional expression or field.

...
Is a symbol used to indicate the preceding expression may be repeated.

rule_keyword
Is a predefined keyword that describes a general system attribute, such as host name (`hostname`) or memory size (`memsize`). It is used with the `rule` value to match a system with the same attribute to a profile. See Table 8–2 for the list of `rule` keywords.

rule_value
Is a value that provides the specific system attribute for the corresponding `rule` keyword. See Table 8–2 for the list of `rule` values.

&&
Is a symbol that must be used to join rule keyword and rule value pairs together in the same rule (a logical AND) . During a custom JumpStart installation, a system must match every pair in the rule before the rule matches.

begin
Is a name of an optional Bourne shell script that can be executed before the installation begins. If no begin script exists, you *must* enter a minus sign (-) in this field. All begin scripts must reside in the JumpStart directory.

See “Creating Begin Scripts” on page 125 for detailed information on how to create begin scripts.

profile

Is a name of a text file that defines how the Solaris software will be installed on the system when a system matches the rule. The information in a profile consists of profile keywords and their corresponding profile values. All profiles must reside in the JumpStart directory.

Note - There are optional ways to use the profile field, which are described in "Using a Site-Specific Installation Program" on page 138 and "Creating Derived Profiles With Begin Scripts" on page 126.

finish

Is a name of an optional Bourne shell script that can be executed after the installation completes. If no finish script exists, you must enter a minus sign (-) in this field. All finish scripts must reside in the JumpStart directory.

See "Creating Finish Scripts" on page 127 for detailed information on how to create finish scripts.

3. **Save the `rules` file into the JumpStart directory.**

   The `rules` file should be owned by root and have permissions equal to 644.

Where to Go Next

This completes the procedure to create a `rules` file. To create profiles, go to "Creating a Profile" on page 95.

Rule Examples

The following illustration shows several example rules in a `rules` file. Each line has a rule keyword and a valid value for that keyword. The Solaris installation program scans the `rules` file from top to bottom. When the Solaris installation program matches a rule keyword and value with a known system, it installs the Solaris software specified by the profile listed in the profile field.

```
#  rule keywords and rule values      begin script       profile       finish script
 #  ----------------------------      ------------       --------      -------------
1 hostname eng-1                       -               basic_prof         -
 2 network 192.43.34.0 && !model \
 'SUNW,Sun 4_50'                       -                net_prof           -
 3 model SUNW,SPARCstation-LX           -                 lx_prof         complete
 4 network 193.144.2.0 && karch i86pc  setup              x86_prof        done
 5 memsize 16-32 && arch ppc             -                prog_prof         -
 6 any  -                                -                generic_prof        -
```

(continued)

1. This rule matches if the system's host name is eng-1. The basic_prof profile is used to install the Solaris software on the system that matches this rule.

2. The rule matches if the system is on subnet 192.43.34.0 and it is *not* a SPARCstation IPX™ (SUNW,Sun 4_50). The net_prof profile is used to install the Solaris software on systems that match this rule.

3. The rule matches if the system is a SPARCstation LX. The lx_prof profile and the complete finish script are used to install the Solaris software on systems that match this rule. This rule also provides an example of rule wrap, which is defined on "Important Information About the rules File" on page 88.

4. This rule matches if the system is on subnet 193.144.2.0 and the system is an x86 system. The setup begin script, the x86_prof profile, and the done finish script are used to install the Solaris software on systems that match this rule.

5. This rule matches if the system has 16-32 Mbytes of memory and its a PowerPC. The prog_prof profile is used to install the Solaris software on systems that match this rule.

6. This rule matches any system that did not match the previous rules. The generic_prof profile is used to install the Solaris software on systems that match this rule. If used, –any should always be in the last rule.

Rule Keyword and Rule Value Descriptions

Table 8–2 describes the rule keywords and rule values that you can use in the rules file.

TABLE 8–2 *Rule Keyword* and *Rule Value* Descriptions

Rule Keyword	Rule Values	Description
`any`	minus sign (-)	Match always succeeds.
`arch`	*processor_type* The following table lists the valid values for *processor_type*. **platform** *processor_type* SPARC `sparc` x86 `i386`	Matches a system's processor type. The `uname -p` command reports the system's processor type.
`domainname`	*domain_name*	Matches a system's domain name, which controls how a name service determines information. If you have a system already installed, the `domainname` command reports the system's domain name.
`disksize`	*disk_name size_range* *disk_name* - A disk name in the form `c`*x*`t`*y*`d`*z*, such as `c0t3d0`, or the special word `rootdisk`. If `rootdisk` is used, the disk to be matched is determined in the following order: ■ The disk that contains the pre-installed boot image (new SPARC based system with factory JumpStart installed) ■ The `c0t3d0s0` disk, if it exists ■ The first available disk (searched in kernel probe order) *size_range* - The size of the disk, which must be specified as a range of Mbytes (*xx*-*xx*).	Matches a system's disk (in Mbytes). Example: `disksize c0t3d0 250-300` The example tries to match a system with a `c0t3d0` disk that is between 250 and 300 Mbytes. **Note -** When calculating *size_range*, remember that a Mbyte equals 1,048,576 bytes. A disk may be advertised as a "535 Mbyte" disk, but it may have only 510 million bytes of disk space. The Solaris installation program will actually view the "535 Mbyte" disk as a 510 Mbyte disk because 535,000,000 / 1,048,576 = 510. So, a "535 Mbyte" disk would not match a *size_range* equal to 530-550.
`hostaddress`	*IP_address*	Matches a system's IP address.
`hostname`	*host_name*	Matches a system's host name. If you have a system already installed, the `uname -n` command reports the system's host name.

TABLE 8–2 *Rule Keyword* and *Rule Value* Descriptions *(continued)*

Rule Keyword	Rule Values	Description
`installed`	*slice version* *slice* - A disk slice name in the form `c`*w*`t`*x*`d`*y*`s`*z*, such as `c0t3d0s5`, or the special words `any` or `rootdisk`. If `any` is used, all of the system's disks will try to be matched (in kernel probe order). If `rootdisk` is used, the disk to be matched is determined in the following order: ■ The disk that contains the pre-installed boot image (new SPARC based system with factory JumpStart installed) ■ The `c0t3d0s0` disk, if it exists ■ The first available disk (searched in kernel probe order) *version* - A version name, `Solaris_2.`*x*, or the special words `any` or `upgrade`. If `any` is used, any Solaris or SunOS release is matched. If `upgrade` is used, any upgradable Solaris 2.1 or greater release is matched.	Matches a disk that has a root file system corresponding to a particular version of Solaris software. Example: `installed c0t3d0s1 Solaris_2.5` The example tries to match a system that has a Solaris 2.5 root files system on `c0t3d0s1`.
`karch`	*platform_group* Valid values are sun4d, sun4c, sun4m, sun4u, i86pc, or prep. (See Appendix C for a detailed list of systems and their corresponding platform group.)	Matches a system's platform group. If you have a system already installed, the `arch -k` command or the `uname -m` command reports the system's platform group.
`memsize`	*physical_mem* The value must be a range of Mbytes (*xx*-*xx*) or a single Mbyte value.	Matches a system's physical memory size (in Mbytes). Example: `memsize 16-32` The example tries to match a system with a physical memory size between 16 and 32 Mbytes. If you have a system already installed, the output of the `prtconf` command (line 2) reports the system's physical memory size.

TABLE 8–2 *Rule Keyword* and *Rule Value* Descriptions *(continued)*

Rule Keyword	Rule Values	Description
`model`	*platform_name*	Matches a system's platform name. See Appendix C for a list of valid platform names. To find the platform name of an installed system, use the `uname -i` command or the output of the `prtconf` command (line 5). **Note -** If the *platform_name* contains spaces, you must enclose it in single quotes ('). For example: `'SUNW,Sun 4_50'`
`network`	*network_num*	Matches a system's network number, which the Solaris installation program determines by performing a logical AND between the system's IP address and the subnet mask. Example: `network 193.144.2.0` The example tries to match a system with a 193.144.2.8 IP address (if the subnet mask were 255.255.255.0).

TABLE 8–2 *Rule Keyword* and *Rule Value* Descriptions *(continued)*

Rule Keyword	Rule Values	Description
`osname`	`Solaris_2.`*x*	Matches a version of Solaris software already installed on a system. Example: `osname Solaris_2.5` The example tries to match a system with Solaris 2.5 already installed.
`totaldisk`	*size_range* The value must be specified as a range of Mbytes (*xx-xx*).	Matches the total disk space on a system (in Mbytes). The total disk space includes all the operational disks attached to a system. Example: `totaldisk 300-500` The example tries to match a system with a total disk space between 300 and 500 Mbytes. **Note -** When calculating *size_range*, remember that a Mbyte equals 1,048,576 bytes. A disk may be advertised as a "535 Mbyte" disk, but it may have only 510 million bytes of disk space. The Solaris installation program will actually view the "535 Mbyte" disk as a 510 Mbyte disk because 535,000,000 / 1,048,576 = 510. So, a "535 Mbyte" disk would not match a *size_range* equal to 530-550.

Creating a Profile

What Is a Profile

A profile is a text file that defines how to install the Solaris software (for example, which software group to install) on a system. Every rule specifies a profile to define how a system will be installed when the rule is matched. You usually create a different profile for every rule; however, the same profile can be used in more than one rule.

A profile consists of one or more profile keywords and their values. Each profile keyword is a command that controls one aspect of how the Solaris installation program will install the Solaris software on a system. For example, the profile keyword and value

```
system_type  server
```

tells the Solaris installation program to install the system as a server.

Note - If you created the JumpStart directory by using the procedures on "Creating a Profile Diskette " on page 81 or "Creating a Profile Server" on page 77, example profiles should already be in the JumpStart directory.

Important Information about Creating Profiles

A profile must have:

- The `install_type` profile keyword as the first entry.
- Only one profile keyword on a line.
- The `root_device` keyword if the systems being upgraded by the profile have more than one root file system that can be upgraded (if you are creating a profile to perform upgrade installations).

A profile allows:

- A comment after the pound sign (#) anywhere on a line. If a line begins with a #, the entire line is a comment line. If a # is specified in the middle of a line, everything after the # is considered a comment.
- Blank lines.

▼ How to Create a Profile

1. **Open a new text file (with a descriptive name) using the editor of your choice.**

   You can create a new profile or edit one of the sample profiles in the JumpStart directory you created.

   The name of a profile should reflect how it will install the Solaris software on a system (for example, `basic_install`, `eng_profile`, or `user_profile`).

2. **Add profile keywords and profile values to the profile.**

   Refer to the following information as you edit the profile:

- “Important Information about Creating Profiles” on page 96
- “Profile Examples” on page 97
- “Profile Keyword and Profile Value Descriptions” on page 100

3. **Save the profile into the JumpStart directory.**

   A profile should be owned by root and have permissions equal to 644.

4. **Test the profile (optional).**

   See “Testing a Profile” on page 118 for detailed information.

Where to Go Next

This completes the procedure to create a profile. After you’ve created all your profiles, go to “Validating the `rules` File” on page 122.

Profile Examples

The following profile examples describe how you can use different profile keywords and profile values to control how the Solaris software is installed on a system. See “Profile Keyword and Profile Value Descriptions” on page 100 for the list of profile keywords and profile values.

Mounting Remote File Systems and Adding/Deleting Packages

```
# profile keywords       profile values
# -----------------      -----------------
1 install_type            initial_install
2 system_type             standalone
3 partitioning            default
 filesys                 any 60 swap   # specify size of /swap
 filesys                 s_ref:/usr/share/man - /usr/share/man ro
 filesys                 s_ref:/usr/openwin/share/man -
                              /usr/openwin/share/man ro,quota
4 cluster                 SUNWCprog
5 package                 SUNWman delete
 package                 SUNWolman delete
 package                 SUNWxwman delete
 package                 SUNWoldem add
 package                 SUNWxwdem add
 package                 SUNWoldim add
 package                 SUNWxwdim add
```

(continued)

(Continuation)

1. This profile keyword is required in every profile.

2. This profile keyword defines that the system will be installed as a standalone system.

3. The file system slices are determined by the software to be installed (`default` value); however, the size of swap is set to 60 Mbytes and it is installed on any disk (`any` value). The standard and OpenWindows man pages are mounted from the file server, `s_ref`, on the network.

4. The developer software group (`SUNWCprog`) is installed on the system.

5. Because the man pages are being mounted remotely, those packages are selected *not* to be installed on the system; however, the packages containing the OpenLook and X Windows demo programs and images are selected to be installed on the system.

Specifying Where to Install File Systems

```
# profile keywords         profile values
# ----------------          -------------------
 install_type              initial_install
  system_type                standalone

1 partitioning               explicit
  filesys                    c0t0d0s0 auto /
 filesys                    c0t3d0s1 32 swap
  filesys                    any auto usr
2 cluster                    SUNWCall
```

1. The file system slices are determined by the `filesys` keywords (`explicit` value). The size of root is based on the selected software (`auto` value) and it is installed on `c0t0d0s0`; the size of `swap` is set to 32 Mbytes and it is installed on `c0t3d0s1`; and `usr` is based on the selected software and the installation program determines where it is installed (`any` value).

2. The entire distribution software group (`SUNWCall`) is installed on the system.

x86: Using the fdisk Keyword

```
# profile keywords     profile values
# ----------------      -------------------
  install_type          initial_install
  system_type           standalone

1 fdisk                   c0t0d0 0x04 delete
2 fdisk                   c0t0d0 solaris maxfree
3 cluster                 SUNWCall
4 cluster                 SUNWCacc delete
```

1. All fdisk partitions of type DOSOS16 (04 hexadecimal) are deleted from the `c0t0d0` disk.

2. A Solaris fdisk partition is created on the largest contiguous free space on the `c0t0d0` disk.

3. The entire distribution software group (`SUNWCall`) is installed on the system.

4. The system accounting utilities (`SUNWCacc`) are selected *not* to be installed on the system.

Reallocating Disk Space for an Upgrade

```
# profile keywords       profile values
# ----------------       -------------------
1 install_type             upgrade

2 root_device              c0t3d0s2

3 backup_media             remote_filesystem timber:/export/scratch
4 layout_constraint        c0t3d0s2 changeable 100
  layout_constraint        c0t3d0s4 changeable
  layout_constraint        c0t3d0s5 movable

5 package                  SUNWbcp delete
6 package                  SUNWolman add
  package                  SUNWxwman add
   cluster                  SUNWCumux add

7 locale                   de
```

1. This profile upgrades a system by reallocating disk space. In this example, disk space must be reallocated because some file systems on the system did not have enough room for the upgrade.

2. The root file system on `c0t3d0s2` is upgraded.

3. A remote system named `timber` will be used to back up data during the disk space reallocation.

(continued)

4. The `layout_constraint` keywords designate that auto-layout can change slice 2 and 5 (the slices can be moved to another location and their size can be changed) and it can move slice 5 (the slice can be moved to another location but its size stays the same) when it tries to reallocate disk space for the upgrade.

5. The binary compatibility package (`SUNWbcp`) will not be installed on the system after the upgrade.

6. This code ensures that the OpenLook and X Windows man pages and the universal multiplexor software are selected to be installed if they are not installed on the system. (All packages already on the system are automatically upgraded.)

7. The German localization packages are selected to be installed on the system.

Profile Keyword and Profile Value Descriptions

The following sections describe the profile keywords and profile values that you can use in a profile. Profile keywords and their values *are* case sensitive.

Table 8–3 provides a quick way to determine which keywords you can use based on your installation scenario. Unless otherwise noted in the profile keyword descriptions, the profile keyword can only be used with the initial installation option.

TABLE 8–3 Profile Keyword Overview

	Installation Scenarios				
Profile Keywords	**Standalone System (Non-Networked)**	**Standalone System (Networked) or Server**	**OS Server**	**Upgrade**	**Upgrade with Disk Space Reallocation**
`backup_media`					x
`boot_device`	x	x	x		
`client_arch`			x		
`client_root`			x		

TABLE 8–3 Profile Keyword Overview *(continued)*

Profile Keywords	Installation Scenarios				
	Standalone System (Non-Networked)	**Standalone System (Networked) or Server**	**OS Server**	**Upgrade**	**Upgrade with Disk Space Reallocation**
`client_swap`			x		
`cluster` (adding software groups)	x	x	x		
`cluster` (adding/ deleting clusters)	x	x	x	x	x
`dontuse`	x	x	x		
`fdisk`	x	x	x		
`filesys` (mounting remote filesystems)		x	x		
`filesys` (creating local filesystems)	x	x	x		
`install_type`	x	x	x	x	x
`layout_constraint`					x
`locale`	x	x	x	x	x
`num_clients`			x		
`package`	x	x	x	x	x
`paritioning`	x	x	x		
`root_device`	x	x	x	x	x

TABLE 8–3 Profile Keyword Overview *(continued)*

Profile Keywords	Installation Scenarios				
	Standalone System (Non-Networked)	**Standalone System (Networked) or Server**	**OS Server**	**Upgrade**	**Upgrade with Disk Space Reallocation**
`system_type`	x	x	x		
`usedisk`	x	x	x		

`backup_media` Profile Keyword

`backup_media` *type* *path*

Note - `backup_media` must be used only with the upgrade option when disk space reallocation is required.

`backup_media` defines the media that will be used to back up file systems if space needs to be reallocated during an upgrade because of space problems. If multiple tapes or diskettes are required for the backup, you will be prompted to insert tapes or diskettes during the upgrade.

Valid *type* Values	Valid *path* Values	Description
`local_tape`	`/dev/rmt/`*n*	Specifies a local tape drive on the system being upgraded. *path* must be the character (raw) device path for the tape drive, where *n* is the number of the tape drive.
`local_diskette`	`/dev/rdiskette`*n*	Specifies a local diskette drive on the system being upgraded. *path* must be the character (raw) device path for the diskette drive, where *n* is the number of the diskette drive. **Note -** Diskettes used for the backup must be formatted.

Valid *type* Values	**Valid *path* Values**	**Description**
`local_filesystem`	`/dev/dsk/c`*w*`t`*x*`d`*y*`s`*z* /*file_system*	Specifies a local file system on the system being upgraded. You cannot specify a local file system that is being changed by the upgrade. *path* can be a block device path for a disk slice (`t`*x* may not be needed) or the absolute path to a file system mounted by the `/etc/vfstab` file.
`remote_filesystem`	*host*`:/`*file_system*	Specifies an NFS file system on a remote system. *path* must include the name or IP address of the remote system (*host*) and the absolute path to the NFS file system (*file_system*). The NFS file system must have read / write access.
`remote_system`	*user@host*`:/`*directory*	Specifies a directory on a remote system that can be reached by a remote shell (`rsh`). The system being upgraded must have access to the remote system through the remote system's `.rhosts` file. *path* must include the name of the remote system (host) and the absolute path to the directory (*directory*). If a user login (*user*) is not specified, the login will be tried as root.

Examples:

```
backup_media local_tape /dev/rmt/0

backup_media local_diskette /dev/rdiskette1

backup_media local_filesystem /dev/dsk/c0t3d0s4

backup_media local_filesystem /export

backup_media remote_filesystem system1:/export/temp

backup_media remote_system user1@system1:/export/temp
```

`boot_device` Profile Keyword

`boot_device` *device* *eeprom*

`boot_device` designates the device where the installation program will install the root file system and consequently what the system's boot device will be. The *eeprom* value also enables you to update the system's EEPROM if you change the system's current boot device, so the system can automatically boot from the new boot device (SPARC systems only).

If you don't specify the `boot_device` keyword in a profile, the following `boot_device` keyword is specified by default during the installation: `boot_device any update`.

device - Choose what the boot device will be.

- c*w*t*x*d*y*s*z* or c*x*d*y*s*z* - The disk slice where the installation program places the root file system, for example, `c0t0d0s0`. (SPARC based systems only).
- c*w*t*x*d*y* or c*x*d*y* - The disk where the installation program places the root file system, for example, `c0t0d0`. (x86 based systems only).
- `existing` - The installation program places the root file system on the system's existing boot device.
- `any` - The installation program chooses where to place the root file system. It will try to use the system's existing boot device; however, it can choose a different boot device if it needs to.

eeprom - Choose if you want to update the system's EEPROM to the specified boot device (SPARC based systems only). For x86 based systems, you must always specify the `preserve` value.

- `update` - The installation program updates the system's EEPROM to the specified boot device, so the installed system will automatically boot from it.
- `preserve` - The boot device value in the system's EEPROM is not changed. If you specify a new boot device without changing the system's EEPROM, you will have to manually change the system's EEPROM, so it can automatically boot from the new boot device.

Example:

```
boot_device c0t0d0s2 update
```

Note - `boot_device` must match any `filesys` keywords that specify the root file system and the `root_device` keyword (if specified).

`client_arch` Profile Keyword

```
client_arch karch_value[karch_value...]
```

`client_arch` defines that the OS server will support a different platform group than it uses. If you do not specify `client_arch`, any diskless client or Solstice AutoClient system that uses the OS server must have the same platform group as the server. You must specify each platform group that you want the OS server to support.

Valid values for *karch_value* are sun4d, sun4c, sun4m, sun4u, or i86pc. (See Appendix C for a detailed list of the platform names of various systems.)

Note - `client_arch` can be used only when `system_type` is specified as `server`.

`client_root` Profile Keyword

```
client_root root_size
```

`client_root` defines the amount of root space (*root_size* in Mbytes) to allocate for each client. If you do not specify `client_root` in a server's profile, the installation software will automatically allocate 15 Mbytes of root space per client. The size of the client root area is used in combination with the `num_clients` keyword to determine how much space to reserve for the `/export/root` file system.

Note - `client_root` can be used only when `system_type` is specified as `server`.

`client_swap` Profile Keyword

```
client_swap swap_size
```

`client_swap` defines the amount of swap space (*swap_size* in Mbytes) to allocate for each diskless client. If you do not specify `client_swap`, 32 Mbytes of swap space is allocated.

Example:

```
client_swap 64
```

The example defines that each diskless client will have a swap space of 64 Mbytes.

Note - `client_swap` can be used only when `system_type` is specified as `server`.

`cluster` Profile Keyword (Adding Software Groups)

```
cluster group_name
```

`cluster` designates what software group to add to the system. The cluster names for the software groups are:

Software Group	*group_name*
Core	`SUNWCreq`
End user system support	`SUNWCuser`
Developer system support	`SUNWCprog`
Entire distribution	`SUNWCall`
Entire distribution plus OEM support (SPARC based systems only)	`SUNWCXall`

You can specify only one software group in a profile, and it must be specified before other `cluster` and `package` entries. If you do not specify a software group with `cluster`, the end user software group (`SUNWCuser`) is installed on the system by default.

`cluster` Profile Keyword (Adding/Deleting Clusters)

```
cluster cluster_name [add | delete]
```

Note - `cluster` (adding/deleting clusters) can be used with both the initial installation and upgrade options.

`cluster` designates whether a cluster should be added or deleted from the software group that will be installed on the system. `add` or `delete` indicates whether the cluster should be added or deleted. If you do not specify `add` or `delete`, `add` is set by default.

cluster_name must be in the form `SUNWC`*name*. To view detailed information about clusters and their names, start Admintool on an installed system and choose Software from the Browse menu. Table A–3 and Table A–4 also provide a list of the clusters included in each software group.

For Upgrade:

- All clusters already on the system are automatically upgraded.
- If you specify *cluster_name* `add`, and *cluster_name* is not installed on the system, the cluster is installed.
- If you specify *cluster_name* `delete`, and *cluster_name* is installed on the system, the package is deleted *before* the upgrade begins.

`dontuse` Profile Keyword

```
dontuse disk_name [disk_name...]
```

`dontuse` designates one or more disks that you don't want the Solaris installation program to use when `partitioning default` is specified (by default, the installation program uses all the operational disks on the system). *disk_name* must be specified in the form `c`*x*`t`*y*`d`*z* or `c`*y*`d`*z*, for example, `c0t0d0`.

Note - You cannot specify the `dontuse` keyword and the `usedisk` keyword in the same profile.

`fdisk` Profile Keyword

```
fdisk disk_name   type   size
```

`fdisk` defines how the fdisk partitions are set up on an x86 based system, and you can specify `fdisk` more than once. This is what happens by default with fdisk partitions on x86 based systems:

- All fdisk partitions on the disk are preserved unless you specifically delete them with the `fdisk` keyword (if *size* is `delete` or `0`). Also, all existing fdisk partitions are deleted when *size* is `all`.
- A Solaris fdisk partition that contains a root file system is always designated as the active partition on the disk (an x86 based system boots from the active partition by default).
- If no `fdisk` keyword is specified in a profile, the following `fdisk` keyword is specified during the installation:

```
fdisk all solaris maxfree
```

- `fdisk` entries are processed in the order they are listed in the profile.

disk_name - Choose where the fdisk partition will be created or deleted:

- `c`*x*`t`*y*`d`*z* or `c`*y*`d`*z* - A specific disk, for example, `c0t3d0`.
- `rootdisk` - The variable that contains the value for the system's root disk, which is determined by the Solaris installation program (described on "How the System's Root Disk Is Determined" on page 117).
- `all` - All the selected disks.

type - Choose what type of fdisk partition will be created or deleted on the specified disk:

- `solaris` - A Solaris fdisk partition (SUNIXOS fdisk type).

- `dosprimary` - An alias for primary DOS fdisk partitions (not for extended or data DOS fdisk partitions). When deleting fdisk partitions (*size* is `delete`), `dosprimary` is an alias for the DOSHUGE, DOSOS12, and DOSOS16 fdisk types (they are all deleted). When creating an fdisk partition, `dosprimary` is an alias for the DOSHUGE fdisk partition (a DOSHUGE fdisk partition is created).

- *DDD* - An integer fdisk partition. *DDD* is an integer number (valid values are 1 through 255).

Note - This value can be specified only if *size* is `delete`.

- 0x*HH* - A hexadecimal fdisk partition. *HH* is a hexadecimal number (valid values are 01 through FF).

Note - This value can be specified only if *size* is `delete`.

The following table shows the integer and hexadecimal numbers for some of the fdisk types:

fdisk Type	***DDD***	***HH***
DOSOS12	`1`	`01`
PCIXOS	`2`	`02`
DOSOS16	`4`	`04`
EXTDOS	`5`	`05`
DOSHUGE	`6`	`06`
DOSDATA	`86`	`56`
OTHEROS	`98`	`62`
UNIXOS	`99`	`63`

size - Choose one of the following:

- *DDD* - An fdisk partition of size *DDD* (in Mbytes) is created on the specified disk. *DDD* must be an integer number, and the Solaris installation program automatically rounds the number up to the nearest cylinder boundary. If `0` is specified, it is the same as specifying `delete`.

- `all` - An fdisk partition is created on the entire disk (all existing fdisk partitions will be deleted).

Note - This value can be specified only if *type* is `solaris`.

- `maxfree` - An fdisk partition is created in the largest contiguous free space on the specified disk. If an fdisk partition of the specified *type* already exists on the disk, the existing fdisk partition is used (a new fdisk partition is *not* created on the disk).

Note - There must be at least one unused fdisk partition on the disk and the disk must have free space or the installation will fail. This value can be specified only if *type* is `solaris` or `dosprimary`.

- `delete` - All fdisk partitions of the specified *type* are deleted on the specified disk.

`filesys` Profile Keyword (Mounting Remote File Systems)

```
filesys server:path    server_address    mount_pt_name  [mount_options]
```

This instance of `filesys` sets up the installed system to automatically mount remote file systems when it boots. You can specify `filesys` more than once.

Example:

```
filesys sherlock:/export/home/user2 - /home
```

server: - The name of the server where the remote file system resides (followed by a colon).

path - The remote file system's mount point name, for example, `/usr` or `/export/home`.

server_address - The IP address of the server specified in *server*:*path*. If you don't have a name service running on the network, this value can be used to populate the `/etc/hosts` file with the server's host name and IP address. If you don't want to specify the server's IP address (if you have a name service running on the network), you must specify a minus sign (`-`).

mount_pt_name - The name of the mount point that the remote file system will be mounted on.

mount_options - One or more mount options (`-o` option of the `mount(1M)` command) that are added to the `/etc/vfstab` entry for the specified *mount_pt_name*.

Note - If you need to specify more than one mount option, the mount options must be separated by commas and no spaces. For example: `ro,quota`

---

filesys Profile Keyword (Creating Local File Systems)

```
filesys slice    size    [file_system]    [optional_parameters]
```

This instance of filesys creates local file systems during the installation. You can specify filesys more than once.

slice - Choose one of the following:

- any - The Solaris installation program places the file system on any disk.

Note - any cannot be specified when size is existing, all, free, *start*:*size*, or ignore.

- c*w*t*x*d*y*s*z* or c*x*d*y*s*z* - The disk slice where the Solaris installation program places the file system, for example, c0t0d0s0.
- rootdisk.s*n* - The variable that contains the value for the system's root disk, which is determined by the Solaris installation program (described on "How the System's Root Disk Is Determined" on page 117). The s*n* suffix indicates a specific slice on the disk.

size - Choose one of the following:

- *num* - The size of the file system is set to *num* (in Mbytes).
- existing - The current size of the existing file system is used.

Note - When using this value, you can change the name of an existing slice by specifying *file_system* as a different *mount_pt_name*.

- auto - The size the file system is automatically determined depending on the selected software.
- all - The specified *slice* uses the entire disk for the file system. When you specify this value, no other file systems can reside on the specified disk.
- free - The remaining unused space on the disk is used for the file system.

Note - If free is used as the value to filesys, it must be the last filesys entry in a profile.

- *start*:*size* - The file system is explicitly partitioned: *start* is the cylinder where the slice begins; *size* is the number of cylinders for the slice.

file_system - You can use this optional value when *slice* is specified as any or c*w*t*x*d*y*s*z*. If *file_system* is not specified, unnamed is set by default, but then you can't specify the *optional_parameters* value. Choose one of the following:

- *mount_pt_name* - The file system's mount point name, for example, /var.
- swap - The specified *slice* is used as swap.

- overlap - The specified *slice* is defined as a representation of a disk region (VTOC value is V_BACKUP). By default, slice 2 is an overlap slice that is a representation of the whole disk.

Note - overlap can be specified only when *size* is existing, all, or *start*:*size*.

- unnamed - The specified *slice* is defined as a raw slice, so *slice* will not have a mount point name. If *file_system* is not specified, unnamed is set by default.
- ignore - The specified *slice* is not used or recognized by the Solaris installation program. This can be used to ignore a file system on a disk during an installation, so the Solaris installation program can create a new file system on the same disk with the same name. ignore can be used only when partitioning existing is specified.

optional_parameters - Choose one of the following:

- preserve - The file system on the specified *slice* is preserved.

Note - preserve can be specified only when *size* is existing and *slice* is c*w*t*x*d*y*s*z*.

- *mount_options* - One or more mount options (–o option of the mount(1M) command) that are added to the /etc/vfstab entry for the specified *mount_pt_name*.

Note - If you need to specify more than one mount option, the mount options must be separated by commas and no spaces. For example: ro,quota

install_type Profile Keyword

```
install_type initial_install | upgrade
```

install_type defines whether to perform the initial installation option or upgrade option on the system.

Note - install_type must be the first profile keyword in every profile.

Note - Some profile keywords can only be used with the initial_install option, and this also applies to the upgrade option.

layout_constraint Profile Keyword

```
layout_constraint slice constraint [minimum_size]
```

Note - layout constraint can be used only for the upgrade option when disk space reallocation is required.

layout_constraint designates the constraint auto-layout has on a file system if it needs to reallocate space during an upgrade because of space problems.

If you don't specify the layout_constraint keyword, the:

- File systems requiring more space for the upgrade are marked changeable
- File systems on the same disk as the file system requiring more space (mounted by the /etc/vfstab file) are marked changeable
- Remaining file systems are marked fixed (auto-layout can't change them)

If you specify one or more layout_constraint keywords, the

- File systems requiring more space for the upgrade are marked changeable
- File systems for which you specified a layout_constraint keyword are marked with the specified constraint
- Remaining file systems are marked fixed

Even though you can't change the constraint on file systems requiring more space for the upgrade (they must be marked changeable), you can use layout_constraint on those file systems to change their *minimum_size* values.

Note - To help auto-layout reallocate space, select more file systems to be changeable or moveable, especially those that reside on the same disks as the file systems that require more space for the upgrade.

slice - This is the file system's disk slice on which to specify the constraint. It must be specified in the form c*w*t*x*d*y*s*z* or c*x*d*y*s*z*.

constraint - Choose one the following constraints for the specified file system.

- changeable - Auto-layout can move the file system to another location and it can change its size. This constraint can only be specified on file systems that are mounted by the /etc/vfstab file. You can change the file system's size by specifying the *minimum_size* value.

  When you mark a file system as changeable and *minimum_size* is not specified, the file system's minimum size will be set to 10% greater than the minimum size required. For example, if the minimum size for a file system is 100 Mbytes, the changed size would be 110 Mbytes. If *minimum_size* is specified, any free space left (original size minus minimum size) will be used for other file systems.

- `movable` - Auto-layout can move the file system to another slice (on the same disk or different disk) and its size stays the same.
- `available` - Auto-layout can use all of the space on the file system to reallocate space. All the data in the file system will be lost. This constraint can only be specified on file systems that are not mounted by the `/etc/vfstab` file.
- `collapse` - Auto-layout will move (collapse) the specified file system into its parent file system. This can be used to reduce the number of file systems on a system as part of the upgrade. For example, if the system has the `/usr` and `/usr/openwin` file systems, collapsing the `/usr/openwin` file system would move it into `/usr` (its parent). This constraint can only be specified on file systems that are mounted by the `/etc/vfstab` file.
- *minimum_size* - This value specifies the size that you want the file system to be when auto-layout reallocates space (it basically lets you change the size of a file system). The size of the file system may end up being more if unallocated space is added to it, but the size will never be less than the value you specify. You can use this optional value only if you have marked a file system as changeable, and the minimum size cannot be less than what the file system needs for its existing contents.

Examples:

```
layout_constraint c0t3d0s1 changeable 200

layout_constraint c0d0s4 movable

layout_constraint c0t3d1s3 availiable

layout_constraint c0t2d0s1 collapse
```

`locale` *locale_name* Profile Keyword

`locale` *locale_name*

Note - `locale` can be used with both the initial installation and upgrade options.

`locale` designates which language or locale packages should be installed (or added for upgrade) for the specified *locale_name*. The *locale_name* values are the same used for the `$LANG` environment variable. See Appendix E for a list of valid language and locale values.

Note - If you’ve preconfigured a default language or locale, it is automatically installed. The English language packages are installed by default.

Note - You can specify a `locale` keyword for each language or locale you need to add to a system.

num_clients Profile Keyword

```
num_clients client_num
```

When a server is installed, space is allocated for each diskless client's root (/) and swap file systems. num_clients defines the number of diskless clients (*client_num*) that a server will support. If you do not specify num_clients, five diskless clients are allocated.

Note - num_clients can be used only when system_type is specified as server.

package Profile Keyword

```
package package_name [add | delete]
```

Note - package can be used with both the initial installation and upgrade options.

package designates whether a package should be added to or deleted from the software group that will be installed on the system. add or delete indicates whether the package should be added or deleted. If you do not specify add | delete, add is set by default.

package_name must be in the form SUNW*name*. Use the pkginfo -l command or Admintool (choose Software from the Browse menu) on an installed system to view detailed information about packages and their names. Table A–3 and Table A–4 also provide a list of the packages included in each software group.

For Upgrade:

- All packages already on the system are automatically upgraded.
- If you specify *package_name* add, and *package_name* is not installed on the system, the package is installed.
- If you specify *package_name* delete, and *package_name* is installed on the system, the package is deleted *before* the upgrade begins.
- If you specify *package_name* delete, and *package_name* is not installed on the system, the package is prevented from being installed if it is part of a cluster that is designated to be installed.

partitioning Profile Keyword

```
partitioning default | existing | explicit
```

`partitioning` defines how the disks are divided into slices for file systems during the installation. If you do not specify `partitioning`, `default` is set.

`default` - The Solaris installation program selects the disks and creates the file systems on which to install the specified software, except for any file systems specified by the `filesys` keyword. `rootdisk` is selected first; additional disks are used if the specified software does not fit on `rootdisk`.

`existing` - The Solaris installation program uses the existing file systems on the system's disks. All file systems except `/`, `/usr`, `/usr/openwin`, `/opt`, and `/var` are preserved. The installation program uses the last mount point field from the file system superblock to determine which file system mount point the slice represents.

Note - When specifying the `filesys` profile keyword with `partitioning existing`, *size* must be `existing`.

`explicit` - The Solaris installation program uses the disks and creates the file systems specified by the `filesys` keywords. If you specify only the root (`/`) file system with the `filesys` keyword, all the Solaris software will be installed in the root file system.

Note - When you use the `explicit` profile value, you must use the `filesys` profile keyword to specify which disks to use and what file systems to create.

`root_device` Profile Keyword

```
root_device slice
```

Note - `root_device` can be used with both the initial installation and upgrade options.

`root_device` designates the system's root disk. See "How the System's Root Disk Is Determined" on page 117 for more information.

For Upgrade:

`root_device` designates the root file system (and the file systems mounted by its `/etc/vfstab` file) to be upgraded. You must specify `root_device` if more than one root file system can be upgraded on a system. *slice* must be specified in the form `c`*w*`t`*x*`d`*y*`s`*z* or `c`*x*`d`*y*`s`*z*.

Example:

```
root_device c0t0d0s2
```

Note - If you specify root_device on a system with only one disk (not required), the root_device and the disk must match. Also, any filesys keywords that specify the root file system must match root_device.

system_type Program Keyword

```
system_type standalone | server
```

system_type defines the type of system being installed. If you do not specify system_type in a profile, standalone is set by default.

usedisk Profile Keyword

```
usedisk disk_name [disk_name...]
```

usedisk designates one or more disks that you want the Solaris installation program to use when partitioning default is specified (by default, the installation program uses all the operational disks on the system). *disk_name* must be specified in the form c*x*t*y*d*z* or c*y*d*z*, for example, c0t0d0.

If you specify the usedisk profile keyword in a profile, the Solaris installation program will only use the disks that you specify with the usedisk profile keyword.

Note - You cannot specify the usedisk keyword and the dontuse keyword in the same profile.

How the Size of Swap Is Determined

If a profile does not explicitly specify the size of swap, the Solaris installation program determines the maximum size that swap can be, based on the system’s physical memory. Table 8–4 shows how the maximum size of swap is determined during a custom JumpStart installation.

TABLE 8–4 How the Maximum Size of Swap Is Determined

Physical Memory (in Mbytes)	Maximum Size of Swap (in Mbytes)
16 - 64	32
64 - 128	64
128 - 512	128
Greater than 512	256

The Solaris installation program will make the size of swap no more than 20% of the disk where it resides, unless there is free space left on the disk after laying out the other file systems. If free space exists, the Solaris installation program will allocate the free space to swap up to the maximum size shown in Table 8–4.

Note - Physical memory plus swap space must be a minimum of 32 Mbytes.

How the System's Root Disk Is Determined

A system's root disk is the disk on the system that contains the root file system. In a profile, you can use the `rootdisk` variable in place of a disk name, which the Solaris installation program sets to the system's root disk. Table 8–5 describes how the installation program determines the system's root disk for the installation. This only applies during an initial installation; a system's root disk cannot change during an upgrade.

TABLE 8–5 How the Installation Program Determines the System's Root Disk (Initial Installation Only)

Stage	Action
1	If the `root_device` keyword is specified in the profile, the installation program sets `rootdisk` to the root device.
2	If `rootdisk` is not set and the `boot_device` keyword is specified in the profile, the installation program sets `rootdisk` to the boot device.
3	If `rootdisk` is not set and a `filesys` `c`*w*`t`*x*`d`*y*`s`*z* *size* `/` entry is specified in the profile, the installation program sets `rootdisk` to the disk specified in the entry.

TABLE 8–5 How the Installation Program Determines the System's Root Disk (Initial Installation Only) *(continued)*

Stage	Action
4	If `rootdisk` is not set and a `rootdisk.s`*n* entry is specified in the profile, the installation program searches the system's disks (in kernel probe order) for an existing root file system on the specified slice. If a disk is found, the installation program sets `rootdisk` to the found disk.
5	If `rootdisk` is not set and `partitioning existing` is specified in the profile, the installation program searches the system's disks (in kernel probe order) for an existing root file system. If a root file system is not found or more than one is found, an error will occur. If a root file system is found, the installation program sets `rootdisk` to the found disk.
6	If `rootdisk` is not set, the installation program sets `rootdisk` to the disk where the root file system will be installed.

Testing a Profile

After you create a profile, you can use the `pfinstall(1M)` command to test the profile and see if it does what you want before using it to install or upgrade a system (called a "dry run" installation). This is especially useful when you are creating upgrade profiles that reallocate disk space.

By looking at the installation output generated by `pfinstall`, you can quickly find out if a profile is going to do what you expect and if the installation is going to be successful. For example, you can see if a system will have enough disk space to upgrade to a new release of Solaris before you actually perform the upgrade on the system.

Ways to Test a Profile

`pfinstall` enables you to test a profile against:

- The system's disk configuration where `pfinstall` is being run.
- Other disk configurations by using a *disk configuration file* that represents a structure of a disk (for example, a disk's bytes/sector, flags, slices). See "SPARC: Creating Disk Configuration Files" on page 132 for detailed information. You cannot use disk configuration files to test an upgrade profile; must test the profile

against the system that you're going to upgrade, because you need to test the profile against the system's disk configuration and its currently installed software.

To successfully and accurately test a profile for a particular Solaris release, you must test a profile within the Solaris environment of the same release. For example, if you want to test a profile for Solaris 2.6, you have to run the `pfinstall` command on a system running Solaris 2.6.

So, on a system running Solaris 2.6, you can test Solaris 2.6 initial installation profiles. However, if you want to test a Solaris 2.6 upgrade profile on a system running a previous version of Solaris, or if you don't have a Solaris 2.6 system installed yet to test Solaris 2.6 initial installation profiles, you have to boot a system from a Solaris 2.6 CD image and temporarily create a Solaris 2.6 install environment. Then, you can run `pfinstall` in the Solaris 2.6 install environment to test the profiles you've created.

Creating a temporary Solaris 2.6 install environment involves booting a system from a Solaris 2.6 CD image (just as you would to install), answering any system identification questions, choosing the Solaris interactive installation program, and exiting out of the first screen that's presented. Then, from the shell, you can execute the `pfinstall` command.

▼ How to Test a Profile

1. **Locate a system to test the profile that has the same platform type (x86 or SPARC) for which the profile was created.**

   If you are testing an upgrade profile, you must use the system that you are going to upgrade.

2. **Determine the next step based on your situation.**

If You ...	Then ...
Need to test an initial installation profile and have a system running Solaris 2.6	Become superuser on the system and go to Step 9 on page 120.
Need to test an upgrade profile, or you don't have a system running Solaris 2.6 to test an initial installation profile	Go to Step 3 on page 119.

3. **Boot the system from a Solaris 2.6 image (just as you would to install), which can be located in the system's local CD-ROM or on an install server.**

   See Chapter 2 for details on booting.

   Note - If you are testing an upgrade profile, boot the system that you are going to upgrade.

4. **Answer the system identification questions, if prompted.**

5. **If you are presented with a choice of installation options, choose the Solaris Interactive Installation program.**

6. **Exit from the first screen of the Solaris interactive installation program.**

   After the Solaris interactive installation program exits, a shell prompt displays.

7. **Create a temporary mount point.**

```
# mkdir /tmp/mnt
```

8. **Mount the directory that contains the profile(s) you want to test.**

If You Want To ...	Then Type ...
Mount a remote NFS file system (for systems on the network)	**mount -F nfs** ***server_name*:*path*** **/tmp/mnt**
Mount a UFS-formatted diskette	**mount -F ufs /dev/diskette /tmp/mnt**
Mount a PCFS-formatted diskette	**mount -F pcfs /dev/diskette /tmp/mnt**

9. **To test the profile with a specific system memory size, set** SYS_MEMSIZE **to the specific memory size in Mbytes.**

```
# SYS_MEMSIZE=memory_size
# export SYS_MEMSIZE
```

10. **Change directory to where the profile resides, which is usually the JumpStart directory.**

    If you mounted a directory in Step 8 on page 120, change directory to /tmp/mnt.

```
# cd jumpstart_dir_path
```

11. Test the profile with the `pfinstall -d` **or** `pfinstall -D` **command.**

Caution - Without the –`d` or –`D` option, `pfinstall` will perform an actual installation of the Solaris software on the system by using the specified profile, and the data on the system will be overwritten.

```
# /usr/sbin/install.d/pfinstall -D | -d disk_config [-c path] profile
```

–`D`

Tells `pfinstall` to use the current system's disk configuration to test the profile against.

–`d` *disk_config*

Tells `pfinstall` to use a disk configuration file, *disk_config*, to test the profile against. If *disk_config* file is not in the directory where `pfinstall` is run, you must specify the path.

This option cannot be used with an upgrade profile (`install-type upgrade`). You must always test an upgrade profile against a system's disk configuration (–`D` option).

–`c` *path*

Is the path to the Solaris CD image. This is required if the Solaris CD is not mounted on `/cdrom`. For example, use this option if the system is using Volume Management to mount the Solaris CD.

Note - This option is not required if you have booted from a Solaris CD image, because the Solaris CD image is mounted on `/cdrom` as part of the booting process.

---

profile

Is the name of the profile to test. If *profile* is not in the directory where `pfinstall` is being run, you must specify the path.

Where to Go Next

You have completed testing the profile. To continue, see "Validating the `rules` File" on page 122.

Example–Testing a Profile

The following example tests the basic_prof profile against the disk configuration on a Solaris 2.6 system where pfinstall is being run. The basic_prof profile is located in the /jumpstart directory and the path to the Solaris CD image is specified because Volume Management is being used.

```
# cd /jumpstart
# /usr/sbin/install.d/pfinstall -D -c /cdrom/cdrom0/s0 basic_prof
```

The following example tests the basic_prof profile against the 535_test disk configuration file and 64 Mbytes of system memory. This example uses a Solaris CD image located in the /export/install directory, and pfinstall is being run on a Solaris 2.6 system.

```
# SYS_MEMSIZE=64
# export SYS_MEMSIZE
# /usr/sbin/install.d/pfinstall -d 535_test -c /export/install basic_prof
```

Validating the rules File

Before the rules file and profiles can be used, you must run the check script to validate that these files are set up correctly. If all the rules and profiles are valid, the rules.ok file is created, which is required by the custom JumpStart installation software to match a system to a profile. Table 8–6 shows what the check script does.

TABLE 8–6 What Happens When You Use `check`

Stage	Description
1	The `rules` file is checked for syntax. `check` makes sure that the rule keywords are legitimate, and the *begin*, *class*, and *finish* fields are specified for each rule (the *begin* and *finish* fields may be a minus sign [`-`] instead of a file name).
2	If no errors are found in the `rules` file, each profile specified in the rules is checked for syntax.
3	If no errors are found, `check` creates the `rules.ok` file from the `rules` file, removing all comments and blank lines, retaining all the rules, and adding the following comment line to the end: `# version=2 checksum=num`

Note - The `rules.ok` file should be owned by root and have permissions equal to 644.

▼ How to Validate the `rules` File

1. **Make sure that the `check` script resides in the JumpStart directory.**

   Note - The `check` script is provided in the `Solaris_2.6/Misc/jumpstart_sample` directory on the Solaris CD.

2. **Change the directory to the JumpStart directory.**

3. **Run the `check` script to validate the `rules` file.**

```
$ ./check [-p path] [-r file_name]
```

–p *path*

Validates the rules file by using the check script from a specified Solaris 2.6 CD image, instead of the check script from the system you are using. *path* is a Solaris installation image on a local disk or a mounted Solaris CD.

Use this option to run the most recent version of `check` if your system is running a previous version of Solaris.

–r *file_name*

Specifies a rules file other than the one named `rules`. Using this option, you can test the validity of a rule before integrating it into the `rules` file.

As the `check` script runs, it reports that it is checking the validity of the `rules` file and the validity of each profile. If no errors are encountered, it reports: `The custom JumpStart configuration is ok`.

Where to Go Next

The rules files is now validated. To read about the optional features available for custom JumpStart installations, see Chapter 9. To perform a custom JumpStart installation on a system, see Chapter 3.

CHAPTER **9**

Using Optional Custom JumpStart Features

This chapter describes the optional features available for custom JumpStart installations, and it is a supplement to Chapter 8.

Note - Instructions in this chapter are valid for either an x86 or SPARC server that is being used to provide custom JumpStart files (called a *profile server*). A profile server can provide custom JumpStart files for systems with the same or different platform type as the server. For example, a SPARC server could provide custom JumpStart files for both SPARC and x86 based systems.

Creating Begin Scripts

What Is a Begin Script

A *begin script* is a user-defined Bourne shell script, specified within the `rules` file, that performs tasks before the Solaris software is installed on the system. Begin scripts can be used only with custom JumpStart installations.

Important Information About Begin Scripts

The following information is important to know about begin scripts:

- Be careful that you do not specify something in the script that would prevent the mounting of file systems onto /a during an initial or upgrade installation. If the Solaris installation program cannot mount the file systems onto /a, an error will occur and the installation will fail.
- Output from the begin script goes to /var/sadm/begin.log.
- Begin scripts should be owned by root and have permissions equal to 644.

Ideas for Begin Scripts

You could set up begin scripts to perform the following tasks:

- Creating derived profiles
- Backing up files before upgrading

Creating Derived Profiles With Begin Scripts

A *derived profile* is a profile that is dynamically created by a begin script during a custom JumpStart installation. Derived profiles are needed when you cannot set up the rules file to match specific systems to a profile (when you need more flexibility than the rules file can provide). For example, you may need to use derived profiles for identical system models that have different hardware components (for example, systems that have different frame buffers).

To set up a rule to use a derived profile, you must:

- Set the profile field to an equal sign (=) instead of a profile.
- Set the begin field to a begin script that will create a derived profile depending on which system is being installed.

When a system matches a rule with the profile field equal to an equal sign (=), the begin script creates the derived profile that is used to install the Solaris software on the system.

An example of a begin script that creates the same derived profile every time is shown below; however, you could add code to this example that would create a different derived profile depending on certain command’s output.

```
#!/bin/sh
echo "install_type        initial_install"    > ${SI_PROFILE}
echo "system_type         standalone"        >> ${SI_PROFILE}
```

(continued)

(Continuation)

```
echo "partitioning        default"           >> ${SI_PROFILE}
echo "cluster             SUNWCprog"         >> ${SI_PROFILE}
echo "package       SUNWman     delete"      >> ${SI_PROFILE}
echo "package       SUNWolman   delete"      >> ${SI_PROFILE}
echo "package       SUNWxwman   delete"      >> ${SI_PROFILE}
```

As shown above, the begin script must use the `SI_PROFILE` environment variable for the name of the derived profile, which is set to `/tmp/install.input` by default.

Note - If a begin script is used to create a derived profile, make sure there are no errors in it. A derived profile is not verified by the `check` script, because it is not created until the execution of the begin script.

Creating Finish Scripts

What Is a Finish Script

A *finish script* is a user-defined Bourne shell script, specified within the `rules` file, that performs tasks after the Solaris software is installed on the system, but before the system reboots. Finish scripts can be used only with custom JumpStart installations.

Important Information About Finish Scripts

The following information is important to know about finish scripts:

- The Solaris installation program mounts the system's file systems onto `/a`. The file systems remain mounted on `/a` until the system reboots. Therefore, you can use the finish script to add, change, or remove files from the newly installed file system hierarchy by modifying the file systems respective to `/a`.
- Output from the finish script goes to `/var/sadm/finish.log`.
- Finish scripts should be owned by root and have permissions equal to 644.

Ideas for Finish Scripts

You could set up finish scripts to perform the following tasks:

- Adding files
- Adding packages or patches
- Customizing the root environment
- Setting the system's root password

This section provides finish script examples for all of these tasks.

Adding Files With a Finish Script

Through a finish script, you can add files from the JumpStart directory to the already installed system. This is possible because the JumpStart directory is mounted on the directory specified by the `SI_CONFIG_DIR` variable (which is set to `/tmp/install_config` by default).

Note - You can also replace files by copying files from the JumpStart directory to already existing files on the installed system.

The following procedure enables you to create a finish script to add files to a system after the Solaris software is installed on it:

▼ How to Add Files With a Finish Script

1. **Copy all the files you want added to the installed system into the JumpStart directory.**

2. **Insert the following line into the finish script for each file you want copied into the newly installed file system hierarchy.**

   ```
   cp ${SI_CONFIG_DIR}/file_name /a/path_name
   ```

For example, assume you have a special application, `site_prog`, developed for all users at your site. If you place a copy of `site_prog` into the JumpStart directory, the following line in a finish script would copy the `site_prog` from the JumpStart directory into a system's `/usr/bin` directory during a custom JumpStart installation:

```
cp ${SI_CONFIG_DIR}/site_prog  /a/usr/bin
```

Adding Packages and Patches

You can create a finish script to automatically add packages and patches after Solaris is installed on a system. This will not only save you time, but it can ensure consistency on what packages and patches are installed on various systems at your site. When using the `pkgadd(1M)` or `patchadd(1M)` commands in your finish scripts, you should use the –`R` option to specify `/a` as the root path.

Code Example 9–1 provides an example finish script to add packages.

CODE EXAMPLE 9–1 Adding Packages With a Finish Script

```
 #!/bin/sh

  BASE=/a
  MNT=/a/mnt
  ADMIN_FILE=/a/tmp/admin

  mkdir ${MNT}
1 mount -f nfs sherlock:/export/package ${MNT}
2 cat >${ADMIN_FILE} <<DONT_ASK
  mail=root
  instance=overwrite
  partial=nocheck
  runlevel=nocheck
  idepend=nocheck
  rdepend=nocheck
  space=ask
  setuid=nocheck
  conflict=nocheck
  action=nocheck
  basedir=default
  DONT_ASK

3 /usr/sbin/pkgadd -a ${ADMIN_FILE} -d ${MNT} -R ${BASE} SUNWxyz

  umount ${MNT}
  rmdir ${MNT}
```

1. Mounts a directory on a server that contains the package to install.

2. Creates a temporary package administration file, `patchadd(1M)`, to force the `pkgadd(1M)` command not to perform checks (and prompt for questions) when installing a package. This enables you to maintain a hand-off installation when you are adding packages.

(continued)

(Continuation)

3. Adds the package by using the –a option (specifying the package administration file) and the –R option (specifying the root path).

In the past, the chroot(1M) command was used with the pkgadd and patchadd commands in the finish script environment. Although this is not recommended, there may be some packages or patches that will not work with the –R option. In those instances, you must create a fake /etc/mnttab file in the /a root path before using the chroot command. The easiest way to do this is to add the following line to your finish script.

```
cp /etc/mnttab /a/etc/mnttab
```

Customizing the Root Environment With a Finish Script

Through a finish script, you can customize files already installed on the system. For example, the finish script in Code Example 9–2 customizes the root environment by appending information to the .cshrc file in the root directory.

CODE EXAMPLE 9–2 Customizing the Root Environment With a Finish Script

```
#!/bin/sh
#
# Customize root’s environment
#
echo "***adding customizations in /.cshrc"
test -f a/.cshrc || {
cat >> a/.cshrc <<EOF
set history=100 savehist=200 filec ignoreeof prompt="\$user@‘uname -n‘> "
alias cp cp -i
alias mv mv -i
alias rm rm -i
alias ls ls -FC
alias h history
alias c clear
unset autologout
EOF
}
```

Setting the System’s Root Password With a Finish Script

After Solaris software is installed on a system, the system reboots. Before the boot process is completed, the system prompts for the root password. This means that until someone enters a password, the system cannot finish booting.

The `auto_install_sample` directory provides a finish script called `set_root_pw` that sets the root password for you, which is shown in Code Example 9–3. This allows the initial reboot of the system to be completed without prompting for a root password.

CODE EXAMPLE 9–3 Setting the System’s Root Password With a Finish Script

```
  #!/bin/sh
  #
  #       @(#)set_root_pw 1.4 93/12/23 SMI
  #
  # This is an example bourne shell script to be run after installation.
  # It sets the system’s root password to the entry defined in PASSWD.
  # The encrypted password is obtained from an existing root password entry
  # in /etc/shadow from an installed machine.

  echo "setting password for root"

  # set the root password
1 PASSWD=dKO5IBkSF42lw
  #create a temporary input file
2 cp /a/etc/shadow /a/etc/shadow.orig

  mv /a/etc/shadow /a/etc/shadow.orig
  nawk -F: ’{
3          if ( $1 == "root" )
           printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n",$1,passwd,$3,$4,$5,$6,$7,$8,$9
      else
          printf"%s:%s:%s:%s:%s:%s:%s:%s:%s\n",$1,$2,$3,$4,$5,$6,$7,$8,$9
      }’ passwd="$PASSWD" /a/etc/shadow.orig > /a/etc/shadow
4 #remove the temporary file
  rm -f /a/etc/shadow.orig
5 # set the flag so sysidroot won’t prompt for the root password
  sed -e ’s/0 # root/1 # root/’ ${SI_SYS_STATE} > /tmp/state.$$
  mv /tmp/state.$$ ${SI_SYS_STATE}
```

1. Sets the variable `PASSWD` to an encrypted root password obtained from an existing entry in a system’s `/etc/shadow` file.

2. Creates a temporary input file of `/a/etc/shadow.`

3. Changes the root entry in the `/etc/shadow` file for the newly installed system using `$PASSWD` as the password field.

(continued)

4. Removes the temporary /a/etc/shadow file.

5. Changes the entry from 0 to a 1 in the state file, so that the user will not be prompted for the root password. The state file is accessed using the variable SI_SYS_STATE, whose value currently is /a/etc/.sysIDtool.state. (To avoid problems with your scripts if this value changes, always reference this file using $SI_SYS_STATE.) The sed command shown here contains a tab character after the 0 and after the 1.

Note - If you set your root password by using a finish script, be sure to safeguard against those who will try to discover the root password from the encrypted password in the finish script.

SPARC: Creating Disk Configuration Files

This section describes how to create single and multiple disk configuration files for a SPARC based system. Disk configuration files enable you to test profiles against different disk configurations before actually installing Solaris software.

▼ SPARC: How to Create Disk Configuration Files

Creating disk configuration files enable you to use pfinstall from a single system to test profiles against different disk configurations. Follow this procedure to create single or multiple disk configuration files for a SPARC based system:

1. **Locate a SPARC based system with a disk that you want to test.**

2. **Become root.**

3. **Create a single disk configuration file by redirecting the output of the** prtvtoc **command to a file:**

```
# prtvtoc /dev/rdsk/device_name > disk_config
```

/dev/rdsk/*device_name* Is the device name of the system's disk. *device_name* must be in the form c*w*t*x*d*y*s2 or c*x*d*y*s2.

disk_config Is the name of the disk configuration file.

4. **If you want to test installing Solaris software on multiple disks, concatenate single disk configuration files together and save the output to a new file:**

```
# cat disk_file1 disk_file2 > multi_disk_config
```

The new file becomes the multiple disk configuration file. For example:

```
# cat 104_disk2 104_disk3 104_disk5 > multi_disk_test
```

5. **If you've created a multiple disk configuration file, and the target numbers in the disk device names are not unique, you must edit this file and make them unique.**

For example, if you concatenated two disk configuration files together that each had target numbers of t0, you would have to change the second target number to t2 as shown:

```
* /dev/rdsk/c0t0d0s2 partition map
...
* /dev/rdsk/c0t2d0s2 partition map
```

Where to Go Next

You have completed creating disk configuration files for a SPARC based system. To use disk configuration files to test profiles, see "Testing a Profile" on page 118.

Example

The following example creates a single disk configuration file, 104_test, on a SPARC based system with a 104-Mbyte disk.

You would redirect the output of the prtvtoc command to a single disk configuration file named 104_test.

```
# prtvtoc /dev/rdsk/c0t3d0s2 > 104_test
```

The `104_test` file would look like this:

```
* /dev/rdsk/c0t3d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      72 sectors/track
*      14 tracks/cylinder
*    1008 sectors/cylinder
*    2038 cylinders*    2036 accessible cylinders
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       1      2    00          0     164304   164303   /
       2      5    00          0    2052288  2052287
       3      0    00     164304     823536   987839   /disk2/b298
       5      0    00     987840     614880  1602719   /install/298/sparc/work
       7      0    00    1602720     449568  2052287   /space
```

x86: Creating Disk Configuration Files

This section describes how to create single and multiple disk configuration files for an x86 based system. Disk configuration files enable you to test profiles against different disk configurations before actually installing Solaris software.

x86: How to Create Disk Configuration Files

Disk configuration files enable you to use `pfinstall` from a single system to test profiles against different disk configurations. Follow this procedure to create single and multiple disk configuration files for an x86 based system:

1. **Locate an x86 based system with a disk that you want to test.**

2. **Become root.**

3. **Create part of the single disk configuration file by saving the output of the** `fdisk` **command to a file**:

```
# fdisk -R -W disk_config -h /dev/rdsk/device_name
```

disk_config Is the name of a disk configuration file.

`/dev/rdsk/`*device_name* Is the device name of the fdisk layout of the entire disk. *device_name* must be in the form `c`*w*`t`*x*`d`*y*`p0` or `c`*x*`d`*y*`p0`.

4. **Append the output of the `prtvtoc` command to the disk configuration file:**

```
# prtvtoc /dev/rdsk/device_name >> disk_config
```

`/dev/rdsk/`*device_name* Is the device name of the system's disk. *device_name* must be in the form `c`*w*`t`*x*`d`*y*`s2` or `c`*x*`d`*y*`s2`.

disk_config Is the name of the disk configuration file.

5. **If you want to test installing Solaris software on multiple disks, concatenate single disk configuration files together and save the output to a new file**

```
# cat disk_file1 disk_file2 > multi_disk_config
```

The new file becomes the multiple disk configuration file. For example:

```
# cat 104_disk2 104_disk3 104_disk5 > multi_disk_test
```

6. **If you've created a multiple disk configuration file, and the target numbers in the disk device names are not unique, you must edit this file and make them unique.**

For example, if you concatenated two disk configuration files together that each had target numbers of `t0`, you would have to change the second target number to `t2` as shown:

```
* /dev/rdsk/c0t0d0p0 default fdisk table
...
* /dev/rdsk/c0t2d0p0 default fdisk table
```

(continued)

(Continuation)

Where to Go Next

You have completed creating disk configuration files for an x86 based system. To use disk configuration files to test profiles, see "Testing a Profile" on page 118.

Example

The following example creates a single disk configuration file, `500_test`, on an x86 based system with a 500-Mbyte disk.

First, you would save the output of the `fdisk` command to a file named `500_test`:

```
# fdisk -R -W 500_test -h /dev/rdsk/c0t0d0p0
```

The `500_test` file would look like this:

```
 * /dev/rdsk/c0t0d0p0 default fdisk table
* Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
*  HBA Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
* systid:
*  1:    DOSOS12
*  2:    PCIXOS
*  4:    DOSOS16
*  5:    EXTDOS
*  6:    DOSBIG
*  86:   DOSDATA
*  98:   OTHEROS
*  99:   UNIXOS
* 130:   SUNIXOS
*
* Id  Act Bhead Bsect   Bcyl  Ehead  Esect  Ecyl Rsect  Numsect
 130  128 44    3       0     46    30     1001 1410   2050140
```

(continued)

(Continuation)

Second, you would append the output of the prtvtoc command to the 500_test file:

```
# prtvtoc /dev/rdsk/c0t0d0s2 >> 500_test
```

The 500_test file is now a complete disk configuration file:

```
* /dev/rdsk/c0t0d0p0 default fdisk table
* Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
*  HBA Dimensions:
*     512 bytes/sector
*      94 sectors/track
*      15 tracks/cylinder
*    1455 cylinders
*
* systid:
*  1:    DOSOS12
*  2:    PCIXOS
*  4:    DOSOS16
*  5:    EXTDOS
*  6:    DOSBIG
*  86:   DOSDATA
*  98:   OTHEROS
*  99:   UNIXOS
*  130:  SUNIXOS
*
* Id  Act Bhead Bsect Bcyl  Ehead  Esec  Ecyl Rsect  Numsect
 130  128 44    3     0     46    30    1001 1410   2050140
* /dev/rdsk/c0t0d0s2 partition map
*
* Dimensions:
*      512 bytes/sector
*       94 sectors/track
*       15 tracks/cylinder
*     1110 sectors/cylinder
*     1454 cylinders
*     1452 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*                          First    Sector    Last
```

(continued)

(Continuation)

```
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       2      5    01       1410   2045910   2047319
       7      6    00       4230   2043090   2047319  /space
       8      1    01          0      1410     1409
       9      9    01       1410      2820     422987
```

Using a Site-Specific Installation Program

Through the use of begin and finish scripts, sites with special requirements can install the Solaris software by creating their own installation program. When a minus sign (-) is specified in the profile field, the begin and finish scripts control how the system is installed, instead of the profile and the Solaris installation program.

For example, if the following rule would match, the `x_install.beg` begin script and the `x_install.fin` finish script would install the system named `sherlock` (the Solaris installation program would not be used):

```
hostname sherlock  x_install.beg  -  x_install.fin
```

APPENDIX **A**

Planning Disk Space

Before installing the Solaris software, you can determine if your system will have enough disk space by doing some high-level planning. If you take time to plan, you'll be able to add more disks to your system, if you need them, before you even start installing.

Basic Considerations for Disk Space Planning

Planning disk space is different for everyone; however, here are some basic considerations:

- Allocate additional disk space for each language selected (for example, Chinese, Japanese, Korean).
- Allocate additional disk space in the `/var` file system if printing or mail support is planned.
- Allocate additional disk space on a server if it's going to provide home file systems for users on other systems (by default, home directories are usually located in the `/export` file system).
- Allocate additional disk space on an OS server for diskless clients or Solstice AutoClient systems. The Solaris Interactive Installation program enables you to allocate space for these systems (by default, disk space is allocated in the `/export` file system).
- Make sure you allocate enough swap space. See Table 8–4 for information about how much swap space you should allocate on a system.

- Table A–1 and Table A–2 list the overall contents and sizes of the Solaris software groups that you can select to install. When planning disk space, remember that the Solaris Interactive Installation program enables you to add or remove individual software packages from the software group that you select.
- Create a minimum number of file systems. By default, the Solaris Interactive Installation program creates only root (`/`), `/usr`, and swap (`/export` is also created when space is allocated for OS services). Creating a minimum number of file systems helps with future upgrades and file system expansion, because separate file systems are limited by their slice boundaries.
- Allocate additional disk space for co-packaged or third-party software.

Software Group Contents and Sizes

Table A–1 and Table A–2 provide an overall view of what each Solaris software group contains and how much approximate disk space they require.

TABLE A–1 SPARC: Software Group Contents and Approximate Sizes

Software Group	Size	What It Contains
End User System Support	281 Mbytes	■ Windowing software ■ Common Desktop Environment (CDE) ■ OpenWindows™ ■ Motif runtime libraries ■ Energy saving software (Power Management™) ■ Basic networking support (`telnet`, `rlogin`, `ftp`) ■ Basic language and partial locale support ■ Removable media support (Volume Management) ■ Standard UNIX® utilities (`sed`, `awk`, `nroff`, `troff`, `grep`, `pipes`, `ld`, `ldd`, `spell`) ■ Basic printer support (`lp`, `lpstat`, `lpr`) ■ System support for audio playback and record ■ Java™ VM (ability to run Java applications) ■ Patch utilities ■ Additional hardware support (PCMCIA)
Developer System Support	537 Mbytes	The End User software plus: ■ Development support (`ar`, graphic accelerator support, IPC)

TABLE A–1 SPARC: Software Group Contents and Approximate Sizes *(continued)*

Software Group	Size	What It Contains
		■ CDE/Motif Developer software, runtimes, and manuals ■ Java VM (ability to develop Java applications) ■ OS demo code ■ Power Management GUI tools ■ Online man pages ■ Solaris 1.x compatibility tools ■ Kernel probing support (TNF) ■ Extended language and partial locale support ■ Programming tools and libraries ■ Extended terminal support (`terminfo`) ■ Extended X support (XGL™, XIL™, XCU4) ■ Graphics header (for graphic application development) ■ ISO-8859 required fonts
Entire Distribution	608 Mbytes (SPARC Solaris CD)	The End User and Developer software plus: ■ AnswerBook2™ (online documentation) ■ Full audio tools and demos ■ Enhanced security features (disk quotas, file/directory access monitoring, system accounting) ■ UUCP networking (UNIX-to-UNIX copy) ■ DHCP server (Dynamic Host Configuration Protocol) ■ Additional language and partial locale support (Eastern European) ■ Additional hardware support (Leo, SX/CG14, SunVideo™, SunButtons™, SunDials™, TCX) ■ Enhanced networking support (NIS server, point-to-point protocol) ■ Solstice™ Launcher ■ System recovery tools

TABLE A–1 SPARC: Software Group Contents and Approximate Sizes *(continued)*

Software Group	Size	What It Contains
		■ Additional X features (complete fonts, PEX™) ■ ISO 8559 optional fonts
Entire Distribution Plus OEM System Support	616 Mybtes	The End User, Developer, and Entire Distribution software plus extended hardware support, which includes: ■ Voyager™ drivers and modules ■ sun4u (X server modules, VIS/XIL) ■ SunFastEthernet™/FastWide SCSI adapter drivers ■ PCI drivers ■ M64 graphic accelerator ■ A-10 (PFU) and Fujitsu device drivers and system support

TABLE A–2 x86: Software Group Contents and Approximate Sizes

Software Group	Size	What It Contains
End User System Support	298 Mbytes	■ Windowing software ■ Common Desktop Environment (CDE) ■ OpenWindows™ ■ Motif runtime libraries ■ Energy saving software (Power Management™) ■ Basic networking support (`telnet`, `rlogin`, `ftp`) ■ Basic language and partial locale support ■ Standard UNIX® utilities (`sed`, `awk`, `nroff`, `troff`, `grep`, `pipes`, `ld`, `ldd`, `spell`) ■ Basic printer support (`lp`, `lpstat`, `lpr`) ■ System support for audio playback and record ■ Java™ VM (ability to run Java applications) ■ Patch utilities ■ Removable media support (Volume Management) ■ Platform support for bus-independent, EISA, and ISA drives ■ Additional hardware support (PCMCIA)

TABLE A–2 x86: Software Group Contents and Approximate Sizes *(continued)*

Software Group	Size	What It Contains
Developer System Support	540 Mbytes	The End User software plus: ■ Development support (`ar`, graphic accelerator support, IPC) ■ CDE/Motif Developer software, runtimes, and manuals ■ Java VM (ability to develop Java applications) ■ OS demo code ■ Power Management GUI tools ■ Online man pages ■ Solaris 1.x compatibility tools ■ Kernel probing support (TNF) ■ Extended language and partial locale support ■ Programming tools and libraries ■ Extended terminal support (`terminfo`) ■ Extended X support (XGL™, XIL™, XCU4) ■ Graphics header (for graphic application development) ■ ISO-8859 required fonts
Entire Distribution	599 Mbytes	The End User and Developer software plus: ■ AnswerBook2™ (online documentation) ■ Full audio tools and demos ■ Enhanced security features (disk quotas, file/directory access monitoring, system accounting) ■ UUCP networking (UNIX-to-UNIX copy) ■ DHCP server (Dynamic Host Configuration Protocol) ■ Enhanced networking support (NIS server, point-to-point protocol) ■ Solstice™ Launcher ■ System recovery tools ■ Additional X features (complete fonts, PEX™) ■ Additional language and partial locale support (Eastern European) ■ ISO 8559 optional fonts

Table A–3 and Table A–4 provide a detailed list of the packages and clusters that each software group contains. The table headings correspond to each of the software groups:

- C – Core
- U – End user system support
- D – Developer system support
- E – Entire Distribution
- O – Entire Distribution plus OEM support (SPARC based systems only)

TABLE A–3 **SPARC:** Detailed Software Group Contents (List of Packages)

C	U	D	E	O	Cluster Name	Package Name	Package Description
x	x	x	x	x	SUNWCadm	SUNWadmr	System & Network Administration Root
x	x	x	x	x		SUNWatfsr	AutoFS, (Root)
x	x	x	x	x		SUNWatfsu	AutoFS, (Usr)
x	x	x	x	x		SUNWcar	Core Architecture, (Root)
x	x	x	x	x	SUNWCcg6	SUNWcg6	GX (cg6) Device Driver
x	x	x	x	x	SUNWCcs	SUNWcsd	Core Solaris Devices
x	x	x	x	x	SUNWCcs	SUNWcsr	Core Solaris, (Root)
x	x	x	x	x	SUNWCcs	SUNWcsu	Core Solaris, (Usr)
x	x	x	x	x	SUNWCdfb	SUNWdfb	Dumb Frame Buffer Device Drivers
x	x	x	x	x		SUNWdtcor	Solaris Desktop /usr/dt filesystem anchor
x	x	x	x	x		SUNWesu	Extended System Utilities
x	x	x	x	x		SUNWkey	Keyboard configuration tables
x	x	x	x	x		SUNWkvm	Core Architecture, (Kvm)
x	x	x	x	x	SUNWCptoo	SUNWlibms	Sun WorkShop Bundled shared libm
x	x	x	x	x	SUNWCnis	SUNWnisr	Network Information System, (Root)
x	x	x	x	x	SUNWCnis	SUNWnisu	Network Information System, (Usr)

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
x	x	x	x	x		SUNWos86u	Platform Support, OS Functionality (Usr)
x	x	x	x	x	SUNWCpcmc	SUNWpcmci	PCMCIA Card Services, (Root)
x	x	x	x	x	SUNWCpcmc	SUNWpcmcu	PCMCIA Card Services, (Usr)
x	x	x	x	x	SUNWCpcmc	SUNWpcmem	PCMCIA memory card driver
x	x	x	x	x	SUNWCpcmc	SUNWpcser	PCMCIA serial card driver
x	x	x	x	x	SUNWCpcmc	SUNWpcelx	3COM EtherLink III PCMCIA Ethernet Driver
x	x	x	x	x	SUNWCpcmc	SUNWpsdpr	PCMCIA ATA card driver
x	x	x	x	x		SUNWsolnm	Solaris Naming Enabler
x	x	x	x	x		SUNWswmt	Patch Utilities
x	x	x	x	x	SUNWCown	SUNWxwdv	XWindows Window Drivers
x	x	x	x	x	SUNWCown	SUNWxwmod	OpenWindows kernel modules
	x	x	x	x		SUNWab2m	Solaris Documentation Server Lookup
	x	x	x	x	SUNWCadm	SUNWadmap	System & Network Administration Applications
	x	x	x	x	SUNWCadm	SUNWadmc	System administration core libraries
	x	x	x	x	SUNWCadm	SUNWadmfw	System & Network Administration Framework
	x	x	x	x	SUNWCaud	SUNWaudio	Audio applications
	x	x	x	x		SUNWbcp	Binary Compatibility
	x	x	x	x	SUNWCpm	SUNWcpr	Suspend, Resume package
	x	x	x	x		SUNWdoc	Documentation Tools
	x	x	x	x	SUNWCdtrun	SUNWdtbas	CDE application basic runtime environment

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
	x	x	x	x	SUNWCdtrun	SUNWdtdmn	CDE daemons
	x	x	x	x	SUNWCdtusr	SUNWdtdst	CDE Desktop Applications
	x	x	x	x	SUNWCdtrun	SUNWdtdte	Solaris Desktop Login Environment
	x	x	x	x	SUNWCdtusr	SUNWdthe	CDE HELP RUNTIME
	x	x	x	x	SUNWCdtusr	SUNWdthev	CDE HELP VOLUMES
	x	x	x	x	SUNWCdtusr	SUNWdticn	CDE icons
	x	x	x	x	SUNWCdtrun	SUNWdtlog	System boot for Desktop Login
	x	x	x	x	SUNWCdtusr	SUNWdtwm	CDE DESKTOP WINDOW MANAGER
	x	x	x	x	SUNWCdtusr	SUNWdtim	Solaris CDE Image Viewer
	x	x	x	x	SUNWCdtusr	SUNWdtrme	CDE Release Documentation
	x	x	x	x		SUNWenise	Base Partial Locales
	x	x	x	x		SUNWfns	Federated Naming System
	x	x	x	x		SUNWgfxcf	SMCC Graphics X Window Server Configuration probe modules
	x	x	x	x		SUNWinst	Install Software
	x	x	x	x		SUNWipc	Interprocess Communications
	x	x	x	x		SUNWislcc	XSH4 conversion for Eastern European locales
	x	x	x	x	SUNWCjv	SUNWjvjit	Java JIT compiler
	x	x	x	x	SUNWCjv	SUNWjvrt	JavaVM run time environment
	x	x	x	x		SUNWlibC	SPARCompilers Bundled libC
	x	x	x	x		SUNWlibCf	SPARCompilers Bundled libC (cfront version)
	x	x	x	x		SUNWloc	System Localization

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
	x	x	x	x	SUNWClp	SUNWlpmsg	LP Alerts
	x	x	x	x		SUNWmfrun	Motif RunTime Kit
	x	x	x	x	SUNWCsea	SUNWmibii	Solstice Enterprise Agent SNMP daemon
	x	x	x	x	SUNWCntp	SUNWntpr	NTP, (Root)
	x	x	x	x	SUNWCntp	SUNWntpu	NTP, (Usr)
	x	x	x	x	SUNWCown	SUNWolaud	OPEN LOOK Audio applications
	x	x	x	x	SUNWCown	SUNWolbk	OpenWindows online handbooks
	x	x	x	x	SUNWCown	SUNWoldcv	OPEN LOOK document and help viewer applications
	x	x	x	x	SUNWCown	SUNWoldst	OPEN LOOK deskset tools
	x	x	x	x	SUNWCown	SUNWoldte	OPEN LOOK Desktop Environment
	x	x	x	x	SUNWCown	SUNWolimt	OPEN LOOK imagetool
	x	x	x	x	SUNWCown	SUNWolrte	OPEN LOOK toolkits runtime environment
	x	x	x	x	SUNWCown	SUNWowbcp	OpenWindows binary compatibility
	x	x	x	x	SUNWCown	SUNWowrqd	OpenWindows required core package
	x	x	x	x	SUNWClp	SUNWpcr	SunSoft Print - Client, (root)
	x	x	x	x	SUNWClp	SUNWpcu	SunSoft Print - Client, (usr)
	x	x	x	x		SUNWploc	European Partial Locales
	x	x	x	x		SUNWplow	OpenWindows enabling for Partial Locales
	x	x	x	x	SUNWCpmow	SUNWpmowu	Power Management OW Utilities, (Usr)
	x	x	x	x	SUNWCpmow	SUNWpmowr	Power Management OW Utilities, (Root)

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
	x	x	x	x	SUNWCpm	SUNWpmr	Power Management config file and rc script
	x	x	x	x	SUNWCpm	SUNWpmu	Power Management binaries
	x	x	x	x	SUNWClp	SUNWpsf	PostScript filters - (Usr)
	x	x	x	x	SUNWClp	SUNWpsr	SunSoft Print - LP Server, (root)
	x	x	x	x	SUNWClp	SUNWpsu	SunSoft Print - LP Server, (usr)
	x	x	x	x		SUNWrdm	On-Line Open Issues ReadMe
	x	x	x	x	SUNWCsea	SUNWsacom	Solstice Enterprise Agent files for root file system
	x	x	x	x	SUNWCsea	SUNWsadmi	Solstice Enterprise Agent Desktop Management Interface
	x	x	x	x	SUNWCsea	SUNWsasnm	Solstice Enterprise Agent Simple Network Management Protocol
	x	x	x	x		SUNWscbcp	SPARCompilers Binary Compatibility Libraries
	x	x	x	x	SUNWCscp	SUNWscplp	SunSoft Print - Source Compatibility, (Usr)
	x	x	x	x	SUNWCscp	SUNWscpr	Source Compatibility, (Root)
	x	x	x	x	SUNWCscp	SUNWscpu	Source Compatibility, (Usr)
	x	x	x	x		SUNWspl	Spell Checking Engine - Base Release (English)
	x	x	x	x		SUNWsregu	Solaris User Registration
	x	x	x	x	SUNWCdtusr	SUNWthj	HotJava Browser for Solaris
	x	x	x	x	SUNWCtltk	SUNWtltk	ToolTalk runtime
	x	x	x	x		SUNWtoo	Programming Tools
	x	x	x	x	SUNWCvol	SUNWvolg	Volume Management Graphical User Interface

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
	x	x	x	x	SUNWCvol	SUNWvolr	Volume Management, (Root)
	x	x	x	x	SUNWCvol	SUNWvolu	Volume Management, (Usr)
	x	x	x	x		SUNWxcu4	XCU4 Utilities
	x	x	x	x	SUNWCxgl	SUNWxgldg	XGL Generic Loadable Libraries
	x	x	x	x	SUNWCxgl	SUNWxgler	XGL English Localization
	x	x	x	x	SUNWCxgl	SUNWxglft	XGL Stroke Fonts
	x	x	x	x	SUNWCxgl	SUNWxglrt	XGL Runtime Environment
	x	x	x	x	SUNWCxil	SUNWxildh	XIL Loadable Pipeline Libraries
	x	x	x	x	SUNWCxil	SUNWxilow	XIL Deskset Loadable Pipeline Libraries
	x	x	x	x	SUNWCxil	SUNWxilrl	XIL Runtime Environment
	x	x	x	x		SUNWxwcfg	X Window Server device configuration
	x	x	x	x	SUNWCfs	SUNWxwcft	X Window System common (not required) fonts
	x	x	x	x	SUNWCfs	SUNWxwfnt	X Window System platform required fonts
	x	x	x	x		SUNWxwfs	Font server
	x	x	x	x	SUNWCown	SUNWxwice	ICE components
	x	x	x	x	SUNWCown	SUNWxwopt	nonessential MIT core clients and server extensions
	x	x	x	x	SUNWCown	SUNWxwplt	X Window System platform software
	x	x	x	x		SUNWxwrtl	X Window System & Graphics Runtime LIbrary Links in /usr/lib
		x	x	x		SUNWarc	Archive Libraries
		x	x	x	SUNWCptoo	SUNWbtool	CCS tools bundled with SunOS

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
		x	x	x	SUNWCutf8	SUNWciu8	Chinese/PRC iconv modules for UTF-8
		x	x	x	SUNWCcg6	SUNWcg6h	GX (cg6) Device Driver
		x	x	x	SUNWCdfb	SUNWdfbh	Dumb Frame Buffer Header Files
		x	x	x	SUNWCdtdev	SUNWdtab	CDE DTBUILDER
		x	x	x	SUNWCdtdev	SUNWdtdem	CDE DEMOS
		x	x	x	SUNWCdtdev	SUNWdthed	CDE HELP DEVELOPER ENVIRONMENT
		x	x	x	SUNWCdtdev	SUNWdtinc	CDE Includes
		x	x	x	SUNWCdtdev	SUNWdtma	CDE man pages
		x	x	x	SUNWCdtdev	SUNWdtmad	CDE developer man pages
		x	x	x	SUNWCutf8	SUNWeudba	UTF-8 L10N for CDE Base
		x	x	x	SUNWCutf8	SUNWeudbd	UTF-8 L10N for CDE Dtbuilder
		x	x	x	SUNWCutf8	SUNWeudda	UTF-8 L10N For CDE Desktop Applications
		x	x	x	SUNWCutf8	SUNWeudhr	UTF-8 L10N For CDE Help Runtime
		x	x	x	SUNWCutf8	SUNWeudhs	UTF-8 L10N For CDE Help Runtime
		x	x	x	SUNWCutf8	SUNWeudis	UTF-8 L10N For CDE Icons
		x	x	x	SUNWCutf8	SUNWeudiv	UTF-8 L10N For Desktop Imagetool
		x	x	x	SUNWCutf8	SUNWeudlg	UTF-8 For CDE Desktop Login
		x	x	x	SUNWCutf8	SUNWeudmg	UTF-8 For Desktop Window Manager
		x	x	x	SUNWCutf8	SUNWeuluf	UTF-8 For Language Environment User Files
		x	x	x	SUNWCutf8	SUNWeuodf	UTF-8 Core OPENLOOK Desktop Files

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
		x	x	x	SUNWCutf8	SUNWeuxwe	UTF-8 X Window Environment
		x	x	x		SUNWhea	SunOS Header Files
		x	x	x	SUNWCutf8	SUNWhiu8	Chinese/Taiwan iconv modules for UTF-8
		x	x	x		SUNWi2rf	X11 ISO-8859-2 required fonts
		x	x	x		SUNWi4rf	X11 ISO-8859-4 required fonts
		x	x	x		SUNWi5rf	X11 ISO-8859-5 required fonts
		x	x	x		SUNWi7rf	X Windows ISO-8859-7 required fonts
		x	x	x		SUNWi9rf	X Windows ISO-8859-9 required fonts
		x	x	x	SUNWCutf8	SUNWjiu8	Japanese iconv modules for UTF-8
		x	x	x	SUNWCjv	SUNWjvdem	JavaVM demo programs
		x	x	x	SUNWCjv	SUNWjvdev	JavaVM developers packages, includes javac, javah, and javap
		x	x	x	SUNWCjv	SUNWjvman	JavaVM man pages
		x	x	x	SUNWCkcms	SUNWkcspf	KCMS Optional Profiles
		x	x	x	SUNWCkcms	SUNWkcspg	KCMS Programmers Environment
		x	x	x	SUNWCkcms	SUNWkcsrt	KCMS Runtime Environment
		x	x	x	SUNWCutf8	SUNWkiu8	Korean UTF-8 iconv modules for UTF-8
		x	x	x	SUNWCptoo	SUNWlibm	Sun WorkShop Bundled libm
		x	x	x		SUNWman	On-Line Manual Pages
		x	x	x		SUNWmfdev	Motif UIL Compiler
		x	x	x		SUNWmfman	CDE Motif Manuals
		x	x	x	SUNWCown	SUNWolinc	OPEN LOOK include files

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
		x	x	x	SUNWCown	SUNWolman	OPEN LOOK toolkit/desktop users man pages
		x	x	x	SUNWCown	SUNWolslb	OPEN LOOK toolkit/desktop static/lint libraries
		x	x	x	SUNWCown	SUNWolsrc	OPEN LOOK sample source
		x	x	x		SUNWosdem	OS demo source
		x	x	x	SUNWCgrap	SUNWpexh	PEX Client Developer Files
		x	x	x	SUNWCpmow	SUNWpmowm	Power Management OW Utilities Man Pages
		x	x	x	SUNWCptoo	SUNWscbcp	SPARCompilers Binary Compatibility Libraries
		x	x	x	SUNWCptoo	SUNWsprot	Solaris Bundled tools
		x	x	x	SUNWCscp	SUNWsra	Source Compatibility Archive Libraries
		x	x	x	SUNWCscp	SUNWsrh	Source Compatibility Header Files
		x	x	x		SUNWter	Terminal Information
		x	x	x	SUNWCtltkp	SUNWtltkd	ToolTalk developer support
		x	x	x	SUNWCtltk	SUNWtltkm	ToolTalk manual pages
		x	x	x		SUNWtnfc	TNF Core Components
		x	x	x		SUNWtnfd	TNF Developer Components
		x	x	x	SUNWCutf8	SUNWuiu8	Iconv modules for UTF-8 Locale
		x	x	x	SUNWCutf8	SUNWuium	Iconv Man Pages for UTF-8 Locale
		x	x	x	SUNWCutf8	SUNWulcf	UFE-8 Locale Environment Common Files
		x	x	x	SUNWCutf8	SUNWuxlcf	UTF-8 X Locale Environment Common Files
		x	x	x		SUNWxcu4t	XCU4 make and sccs utilities

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
		x	x	x	SUNWCgrap	SUNWxglh	XGL Include Files
		x	x	x		SUNWxi18n	X Window System I18N Common Package
		x	x	x	SUNWCgrap	SUNWxilh	XIL API Header Files
		x	x	x	SUNWCgrap	SUNWxilmn	XIL man pages and demos
		x	x	x		SUNWxim	X Window System X Input Method Server Package
		x	x	x	SUNWCown	SUNWxwinc	X Window System include files
		x	x	x	SUNWCown	SUNWxwman	X Window System online user man pages
		x	x	x	SUNWCfs	SUNWxwoft	X Window System optional fonts
		x	x	x	SUNWCown	SUNWxwpmn	X Window System online programmers man pages
		x	x	x	SUNWCown	SUNWxwslb	X Window System static/lint libraries
		x	x	x	SUNWCown	SUNWxwsrc	X Window System sample source
			x	x	SUNWCacc	SUNWaccr	System Accounting, (Root)
			x	x	SUNWCacc	SUNWaccu	System Accounting, (Usr)
			x	x	SUNWCpppk	SUNWapppr	PPP/IP Async PPP daemon configuration files
			x	x	SUNWCpppk	SUNWapppu	PPP/IP Async PPP daemon and PPP login service
			x	x		SUNWast	Automated Security Enhancement Tools
			x	x	SUNWCaud	SUNWaudmo	Audio demo programs
			x	x	SUNWCnet	SUNWbnur	Networking UUCP Utilities, (Root)
			x	x	SUNWCnet	SUNWbnuu	Networking UUCP Utilities, (Usr)

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
			x	x	SUNWCdhcp	SUNWdhcsr	BOOTP/DHCP Server Services, (Root)
			x	x	SUNWCdhcp	SUNWdhcsu	BOOTP/DHCP Server Services, (Usr)
			x	x	SUNWCdial	SUNWdial	Buttons/Dials (bd) Streams Module
			x	x	SUNWCdial	SUNWdialh	Buttons/Dials (bd) Header Files
			x	x		SUNWfac	Framed Access Command Environment
			x	x		SUNWfnsx5	FNS Support For X.500 Directory Context
			x	x		SUNWi1of	ISO-8859-1 (Latin-1) Optional Fonts
			x	x		SUNWi2of	X11 ISO-8859-2 optional fonts
			x	x		SUNWi4of	X11 ISO-8859-4 optional fonts
			x	x		SUNWi5of	X11 ISO-8859-5 optional fonts
			x	x		SUNWi7of	X Windows ISO-8859-7 optional fonts
			x	x		SUNWi9of	X Windows ISO-8859-9 optional fonts
			x	x		SUNWisolc	XSH4 conversion for ISO Latin character sets
			x	x	SUNCleo	SUNWleo	ZX System Software (Device Driver)
			x	x	SUNCleo	SUNWleoo	ZX XGL support
			x	x	SUNCleo	SUNWleor	ZX System Software (Root)
			x	x	SUNCleo	SUNWleow	ZX Window System Support
			x	x	SUNClux	SUNWluxal	Sun Enterprise Network Array socal Device Driver
			x	x	SUNClux	SUNWluxdv	Sun Enterprise Network Array sf Device Driver

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
			x	x	SUNClux	SUNWluxop	Sun Enterprise Network Array firmware and utilities
			x	x	SUNWCown	SUNWoladd	OPEN LOOK Alternate Desktop Demos
			x	x	SUNWCown	SUNWoldem	OPEN LOOK demo programs
			x	x	SUNWCown	SUNWoldim	OPEN LOOK demo images
			x	x	SUNWCpex	SUNWpexcl	PEX Runtime Client Library
			x	x	SUNWCpex	SUNWpexsv	PEX Runtime Server Extension
			x	x		SUNWpldte	Eastern European locale support
			x	x		SUNWploc1	Supplementary Partial Locale
			x	x		SUNWplow1	OpenWindows enabling for Supplementary Partial Locales
			x	x	SUNWCpppk	SUNWpppk	PPP/IP and IPdialup Device Drivers
			x	x	SUNWCrtvc	SUNWrtvc	SunVideo Device Driver
			x	x	SUNWCrtvc	SUNWrtvcl	SunVideo XIL library support
			x	x	SUNWCrtvc	SUNWrtvcu	SunVideo Runtime Support Software
			x	x		SUNWsadml	Solstice Launcher
			x	x		SUNWses	SCSI Enclosure Services Device Driver
			x	x		SUNWssadv	SPARCstorage Array Drivers
			x	x		SUNWssaop	SPARCstorage Array Utility
			x	x		SUNWsutl	Static Utilities
			x	x	SUNWCsx	SUNWsx	SX Shareable Library
			x	x	SUNWCsx	SUNWsxow	SX Window System Support
			x	x	SUNWCsx	SUNWsxr	SX Video Subsystem Drivers

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
			x	x	SUNWCsx	SUNWsxxgl	SX XGL Support
			x	x	SUNWCtcx	SUNWtcx	TCX System Software (Device Driver)
			x	x	SUNWCtcx	SUNWtcxow	TCX Window System Support
			x	x	SUNWCtcx	SUNWtcxu	TCX XGL Support
			x	x	SUNWCsx	SUNWxilcg	CG14 XIL Support
			x	x	SUNWCown	SUNWxwacx	AccessX client program
			x	x	SUNWCown	SUNWxwdem	X Window System demo programs
			x	x	SUNWCown	SUNWxwdim	X Window demo images
			x	x		SUNWxwdxm	DPS motif library
			x	x	SUNWCown	SUNWxwfa	X Window System Font Administrator
			x	x	SUNWCfs	SUNWxwfs	Font server
			x	x		SUNWypr	NIS Server for Solaris (root)
			x	x		SUNWypu	NIS Server for Solaris (usr)
				x	FJSVCvpl	FJSVvplr	Fujitsu platform specific symlinks (Root)
				x	FJSVCvpl	FJSVvplu	Fujitsu platform specific symlinks (user)
				x	PFUCaga	PFUaga	AG-10 Device Driver
				x	PFUCaga	PFUagacf	AG-10 System Software(Usr)
				x	PFUCaga	PFUagaow	AG-10 DDX Support(OpenWindows)
				x	PFUCaga	PFUagar	AG-10 System Software(Root)
				x	PFUCaga	PFUagaxgl	AG-10 Runtime Support

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
				x		PFUcar	PFU/Fujistu kernel/unix for Power Control Software
				x		PFUdfb	S-4/Leia LCD Dumb Frame Buffer Driver
				x	PFUCvpl	PFUvplr	PFU/Fujitsu platform links
				x	PFUCvpl	PFUvplu	PFU/Fujitsu usr/platform links
				x	SUNWCffb	SUNWffb	Creator Graphics (FFB) System Software (Device Driver)
				x	SUNWCffb	SUNWffbcf	Creator Graphics (FFB) Configuration Software
				x	SUNWCffb	SUNWffbw	Creator Graphics (FFB) Window System Support
				x	SUNWCffb	SUNWffbxg	Creator Graphics (FFB) XGL Support
				x	SUNWChmd	SUNWhmd	SunSwift SBus Adapter Drivers
				x	SUNWChmd	SUNWhmdu	SunSwift SBus Adapter Headers
				x	SUNWCide	SUNWider	IDE Device Driver, (Root)
				x	SUNWCkmp2	SUNWkmp2r	PS/2 Keyboard and Mouse Device Drivers, (Root)
				x	SUNWCkmp2	SUNWkmp2u	PS/2 keyboard keymap tables, (Usr)
				x	SUNWCm64	SUNWm64	M64 System Software (Device Driver)
				x	SUNWCm64	SUNWm64cf	M64 Graphics Configuration Software
				x	SUNWCm64	SUNWm64w	M64 Window System Support
				x	SUNWCpd	SUNWpd	PCI Drivers
				x	SUNWCpd	SUNWpdu	PCI Drivers Headers
				x		SUNWvygdr	Voyager Drivers and Modules

TABLE A–3 SPARC: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	O	Cluster Name	Package Name	Package Description
				x		SUNWxilvl	VIS/XIL Support
				x	SUNWCxwkey	SUNWxwkey	X Windows sparc platform software, keytables
				x		SUNWxwpsr	Sun4u-platform specific X server auxiliary filter modules

TABLE A–4 x86: Detailed Software Group Contents (List of Packages)

C	U	D	E	Cluster Name	Package Name	Package Description
x	x	x	x	SUNWCadm	SUNWadmr	System & Network Administration Root
x	x	x	x		SUNWatfsr	AutoFS, (Root)
x	x	x	x		SUNWatfsu	AutoFS, (Usr)
x	x	x	x		SUNWcar	Core Architecture, (Root)
x	x	x	x	SUNWCcs	SUNWcsd	Core Solaris Devices
x	x	x	x	SUNWCcs	SUNWcsr	Core Solaris, (Root)
x	x	x	x	SUNWCcs	SUNWcsu	Core Solaris, (Usr)
x	x	x	x		SUNWdtcor	Solaris Desktop /usr/dt filesystem anchor
x	x	x	x		SUNWesu	Extended System Utilities
x	x	x	x		SUNWkey	Keyboard configuration tables
x	x	x	x		SUNWkvm	Core Architecture, (Kvm)
x	x	x	x	SUNWCptoo	SUNWlibms	Sun WorkShop Bundled shared libm
x	x	x	x	SUNWCnis	SUNWnisr	Network Information System, (Root)
x	x	x	x	SUNWCnis	SUNWnisu	Network Information System, (Usr)

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
x	x	x	x	SUNWCos86	SUNWos86r	Platform Support, OS Functionality (Root)
x	x	x	x	SUNWCos86	SUNWos86u	Platform Support, OS Functionality (Usr)
x	x	x	x	SUNWCpcmc	SUNWpcmci	PCMCIA Card Services, (Root)
x	x	x	x	SUNWCpcmc	SUNWpcmcu	PCMCIA Card Services, (Usr)
x	x	x	x	SUNWCpcmc	SUNWpcmem	PCMCIA memory card driver
x	x	x	x	SUNWCpcmc	SUNWpcser	PCMCIA serial card driver
x	x	x	x	SUNWCpcmc	SUNWpcelx	3COM EtherLink III PCMCIA Ethernet Driver
x	x	x	x		SUNWpsdcr	Platform Support, Bus-independent Device Drivers (Root)
x	x	x	x		SUNWpsder	Platform Support, EISA Bus Device Drivers, (Root)
x	x	x	x		SUNWpsdir	Platform Support, ISA Bus Device Drivers, (Root)
x	x	x	x		SUNWpsdmr	Platform Support, MicroChannel Adapter Device Drivers (Root)
x	x	x	x	SUNWCpcmc	SUNWpsdpr	PCMCIA ATA card driver
x	x	x	x		SUNWsolnm	Solaris Naming Enabler
x	x	x	x		SUNWswmt	Patch Utilities
x	x	x	x	SUNWCown	SUNWxwdv	XWindows Window Drivers
x	x	x	x	SUNWCown	SUNWxwmod	OpenWindows kernel modules
x	x	x	x	SUNWCos86	SUNWxwssu	X Window platform specific, config start up software
	x	x	x		SUNWab2m	Solaris Documentation Server Lookup
	x	x	x	SUNWCadm	SUNWadmap	System & Network Administration Applications

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
	x	x	x	SUNWCadm	SUNWadmc	System administration core libraries
	x	x	x	SUNWCadm	SUNWadmfw	System & Network Administration Framework
	x	x	x	SUNWCaud	SUNWaudio	Audio applications
	x	x	x		SUNWdoc	Documentation Tools
	x	x	x	SUNWCdtrun	SUNWdtbas	CDE application basic runtime environment
	x	x	x	SUNWCdtrun	SUNWdtdmn	CDE daemons
	x	x	x	SUNWCdtusr	SUNWdtdst	CDE Desktop Applications
	x	x	x	SUNWCdtrun	SUNWdtdte	Solaris Desktop Login Environment
	x	x	x	SUNWCdtusr	SUNWdthe	CDE HELP RUNTIME
	x	x	x	SUNWCdtusr	SUNWdthev	CDE HELP VOLUMES
	x	x	x	SUNWCdtusr	SUNWdticn	CDE icons
	x	x	x	SUNWCdtrun	SUNWdtlog	System boot for Desktop Login
	x	x	x	SUNWCdtusr	SUNWdtwm	CDE DESKTOP WINDOW MANAGER
	x	x	x	SUNWCdtusr	SUNWdtim	Solaris CDE Image Viewer
	x	x	x	SUNWCdtusr	SUNWdtrme	CDE Release Documentation
	x	x	x		SUNWenise	Base Partial Locales
	x	x	x		SUNWfns	Federated Naming System
	x	x	x		SUNWgfxcf	SMCC Graphics X Window Server Configuration probe modules
	x	x	x		SUNWinst	Install Software
	x	x	x		SUNWipc	Interprocess Communications
	x	x	x		SUNWislcc	XSH4 conversion for Eastern European locales

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
	x	x	x	SUNWCjv	SUNWjvjit	Java JIT compiler
	x	x	x	SUNWCjv	SUNWjvrt	JavaVM run time environment
	x	x	x		SUNWlibC	SPARCompilers Bundled libC
	x	x	x		SUNWlibCf	SPARCompilers Bundled libC (cfront version)
	x	x	x		SUNWloc	System Localization
	x	x	x	SUNWClp	SUNWlpmsg	LP Alerts
	x	x	x		SUNWmfrun	Motif RunTime Kit
	x	x	x	SUNWCsea	SUNWmibii	Solstice Enterprise Agent SNMP daemon
	x	x	x	SUNWCntp	SUNWntpr	NTP, (Root)
	x	x	x	SUNWCntp	SUNWntpu	NTP, (Usr)
	x	x	x	SUNWCown	SUNWolaud	OPEN LOOK Audio applications
	x	x	x	SUNWCown	SUNWolbk	OpenWindows online handbooks
	x	x	x	SUNWCown	SUNWoldcv	OPEN LOOK document and help viewer applications
	x	x	x	SUNWCown	SUNWoldst	OPEN LOOK deskset tools
	x	x	x	SUNWCown	SUNWoldte	OPEN LOOK Desktop Environment
	x	x	x	SUNWCown	SUNWolimt	OPEN LOOK imagetool
	x	x	x	SUNWCown	SUNWolrte	OPEN LOOK toolkits runtime environment
	x	x	x	SUNWCown	SUNWowrqd	OpenWindows required core package
	x	x	x	SUNWClp	SUNWpcr	SunSoft Print - Client, (root)
	x	x	x	SUNWClp	SUNWpcu	SunSoft Print - Client, (usr)
	x	x	x		SUNWploc	European Partial Locales

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
	x	x	x		SUNWplow	OpenWindows enabling for Partial Locales
	x	x	x		SUNWpmi	OpenWindows VESA PMI files
	x	x	x	SUNWCpmow	SUNWpmowu	Power Management OW Utilities, (Usr)
	x	x	x	SUNWCpmow	SUNWpmowr	Power Management OW Utilities, (Root)
	x	x	x	SUNWCpm	SUNWpmr	Power Management config file and rc script
	x	x	x	SUNWCpm	SUNWpmu	Power Management binaries
	x	x	x	SUNWClp	SUNWpsf	PostScript filters - (Usr)
	x	x	x	SUNWClp	SUNWpsr	SunSoft Print - LP Server, (root)
	x	x	x	SUNWClp	SUNWpsu	SunSoft Print - LP Server, (usr)
	x	x	x		SUNWrdm	On-Line Open Issues ReadMe
	x	x	x	SUNWCsea	SUNWsacom	Solstice Enterprise Agent files for root file system
	x	x	x	SUNWCsea	SUNWsadmi	Solstice Enterprise Agent Desktop Management Interface
	x	x	x	SUNWCsea	SUNWsasnm	Solstice Enterprise Agent Simple Network Management Protocol
	x	x	x	SUNWCscp	SUNWscplp	SunSoft Print - Source Compatibility, (Usr)
	x	x	x	SUNWCscp	SUNWscpr	Source Compatibility, (Root)
	x	x	x	SUNWCscp	SUNWscpu	Source Compatibility, (Usr)
	x	x	x		SUNWspl	Spell Checking Engine - Base Release (English)
	x	x	x		SUNWsregu	Solaris User Registration
	x	x	x	SUNWCdtusr	SUNWthj	HotJava Browser for Solaris
	x	x	x	SUNWCtltk	SUNWtltk	ToolTalk runtime

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
	x	x	x		SUNWtoo	Programming Tools
	x	x	x	SUNWCvol	SUNWvolg	Volume Management Graphical User Interface
	x	x	x	SUNWCvol	SUNWvolr	Volume Management, (Root)
	x	x	x	SUNWCvol	SUNWvolu	Volume Management, (Usr)
	x	x	x		SUNWxcu4	XCU4 Utilities
	x	x	x	SUNWCxgl	SUNWxgldg	XGL Generic Loadable Libraries
	x	x	x	SUNWCxgl	SUNWxgler	XGL English Localization
	x	x	x	SUNWCxgl	SUNWxglft	XGL Stroke Fonts
	x	x	x	SUNWCxgl	SUNWxglrt	XGL Runtime Environment
	x	x	x	SUNWCxil	SUNWxildh	XIL Loadable Pipeline Libraries
	x	x	x	SUNWCxil	SUNWxilow	XIL Deskset Loadable Pipeline Libraries
	x	x	x	SUNWCxil	SUNWxilrl	XIL Runtime Environment
	x	x	x		SUNWxwcfg	X Window Server device configuration
	x	x	x	SUNWCfs	SUNWxwcft	X Window System common (not required) fonts
	x	x	x	SUNWCfs	SUNWxwfnt	X Window System platform required fonts
	x	x	x		SUNWxwfs	Font server
	x	x	x	SUNWCown	SUNWxwice	ICE components
	x	x	x	SUNWCown	SUNWxwopt	nonessential MIT core clients and server extensions
	x	x	x	SUNWCown	SUNWxwplt	X Window System platform software
	x	x	x		SUNWxwrtl	X Window System & Graphics Runtime LIbrary Links in /usr/lib
		x	x		SUNWarc	Archive Libraries

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
		x	x	SUNWCptoo	SUNWbtool	CCS tools bundled with SunOS
		x	x	SUNWCutf8	SUNWciu8	Chinese/PRC iconv modules for UTF-8
		x	x	SUNWCcoff	SUNWcoff	COFF Binary Compatibility
		x	x	SUNWCdfb	SUNWdfbh	Dumb Frame Buffer Header Files
		x	x	SUNWCdtdev	SUNWdtab	CDE DTBUILDER
		x	x	SUNWCdtdev	SUNWdtdem	CDE DEMOS
		x	x	SUNWCdtdev	SUNWdthed	CDE HELP DEVELOPER ENVIRONMENT
		x	x	SUNWCdtdev	SUNWdtinc	CDE Includes
		x	x	SUNWCdtdev	SUNWdtma	CDE man pages
		x	x	SUNWCdtdev	SUNWdtmad	CDE developer man pages
		x	x	SUNWCutf8	SUNWeudba	UTF-8 L10N for CDE Base
		x	x	SUNWCutf8	SUNWeudbd	UTF-8 L10N for CDE Dtbuilder
		x	x	SUNWCutf8	SUNWeudda	UTF-8 L10N For CDE Desktop Applications
		x	x	SUNWCutf8	SUNWeudhr	UTF-8 L10N For CDE Help Runtime
		x	x	SUNWCutf8	SUNWeudhs	UTF-8 L10N For CDE Help Runtime
		x	x	SUNWCutf8	SUNWeudis	UTF-8 L10N For CDE Icons
		x	x	SUNWCutf8	SUNWeudiv	UTF-8 L10N For Desktop Imagetool
		x	x	SUNWCutf8	SUNWeudlg	UTF-8 For CDE Desktop Login
		x	x	SUNWCutf8	SUNWeudmg	UTF-8 For Desktop Window Manager
		x	x	SUNWCutf8	SUNWeuluf	UTF-8 For Language Environment User Files
		x	x	SUNWCutf8	SUNWeuodf	UTF-8 Core OPENLOOK Desktop Files

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
		x	x	SUNWCutf8	SUNWeuxwe	UTF-8 X Window Environment
		x	x		SUNWhea	SunOS Header Files
		x	x	SUNWCutf8	SUNWhiu8	Chinese/Taiwan iconv modules for UTF-8
		x	x		SUNWi2rf	X11 ISO-8859-2 required fonts
		x	x		SUNWi4rf	X11 ISO-8859-4 required fonts
		x	x		SUNWi5rf	X11 ISO-8859-5 required fonts
		x	x		SUNWi7rf	X Windows ISO-8859-7 required fonts
		x	x		SUNWi9rf	X Windows ISO-8859-9 required fonts
		x	x	SUNWCutf8	SUNWjiu8	Japanese iconv modules for UTF-8
		x	x	SUNWCjv	SUNWjvdem	JavaVM demo programs
		x	x	SUNWCjv	SUNWjvdev	JavaVM developers packages, includes javac, javah, and javap
		x	x	SUNWCjv	SUNWjvman	JavaVM man pages
		x	x	SUNWCkcms	SUNWkcspf	KCMS Optional Profiles
		x	x	SUNWCkcms	SUNWkcspg	KCMS Programmers Environment
		x	x	SUNWCkcms	SUNWkcsrt	KCMS Runtime Environment
		x	x	SUNWCutf8	SUNWkiu8	Korean UTF-8 iconv modules for UTF-8
		x	x	SUNWCptoo	SUNWlibm	Sun WorkShop Bundled libm
		x	x		SUNWman	On-Line Manual Pages
		x	x		SUNWmfdev	Motif UIL Compiler
		x	x		SUNWmfman	CDE Motif Manuals
		x	x	SUNWCown	SUNWolinc	OPEN LOOK include files

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
		x	x	SUNWCown	SUNWolman	OPEN LOOK toolkit/desktop users man pages
		x	x	SUNWCown	SUNWolslb	OPEN LOOK toolkit/desktop static/lint libraries
		x	x	SUNWCown	SUNWolsrc	OPEN LOOK sample source
		x	x		SUNWosdem	OS demo source
		x	x	SUNWCgrap	SUNWpexh	PEX Client Developer Files
		x	x	SUNWCpmow	SUNWpmowm	Power Management OW Utilities Man Pages
		x	x		SUNWpsh	Platform Support, Driver Header Files
		x	x		SUNWs53	System V Release 3 File System
		x	x	SUNWCptoo	SUNWscbcp	SPARCompilers Binary Compatibility Libraries
		x	x	SUNWCcoff	SUNWshlib	x86 COFF Shared Libraries
		x	x	SUNWCptoo	SUNWsprot	Solaris Bundled tools
		x	x	SUNWCscp	SUNWsra	Source Compatibility Archive Libraries
		x	x	SUNWCscp	SUNWsrh	Source Compatibility Header Files
		x	x		SUNWter	Terminal Information
		x	x	SUNWCtltkp	SUNWtltkd	ToolTalk developer support
		x	x	SUNWCtltk	SUNWtltkm	ToolTalk manual pages
		x	x		SUNWtnfc	TNF Core Components
		x	x		SUNWtnfd	TNF Developer Components
		x	x	SUNWCutf8	SUNWuiu8	Iconv modules for UTF-8 Locale
		x	x	SUNWCutf8	SUNWuium	Iconv Man Pages for UTF-8 Locale
		x	x	SUNWCutf8	SUNWulcf	UFE-8 Locale Environment Common Files

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
		x	x	SUNWCutf8	SUNWuxlcf	UTF-8 X Locale Environment Common Files
		x	x		SUNWxcu4t	XCU4 make and sccs utilities
		x	x	SUNWCgrap	SUNWxglh	XGL Include Files
		x	x		SUNWxglps	XGL Platform Specific Device Pipeline Libraries
		x	x		SUNWxi18n	X Window System I18N Common Package
		x	x	SUNWCgrap	SUNWxilh	XIL API Header Files
		x	x	SUNWCgrap	SUNWxilmn	XIL man pages and demos
		x	x		SUNWxim	X Window System X Input Method Server Package
		x	x	SUNWCown	SUNWxwinc	X Window System include files
		x	x	SUNWCown	SUNWxwman	X Window System online user man pages
		x	x	SUNWCfs	SUNWxwoft	X Window System optional fonts
		x	x		SUNWxwpls	X Windows platform software, specific
		x	x	SUNWCown	SUNWxwpmn	X Window System online programmers man pages
		x	x		SUNWxwscf	X Windows server probe and configuration, platform specific
		x	x	SUNWCown	SUNWxwslb	X Window System static/lint libraries
		x	x	SUNWCown	SUNWxwsrc	X Window System sample source
			x	SUNWCacc	SUNWaccr	System Accounting, (Root)
			x	SUNWCacc	SUNWaccu	System Accounting, (Usr)
			x	SUNWCpppk	SUNWapppr	PPP/IP Async PPP daemon configuration files

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
			x	SUNWCpppk	SUNWapppu	PPP/IP Async PPP daemon and PPP login service
			x		SUNWast	Automated Security Enhancement Tools
			x	SUNWCaud	SUNWaudmo	Audio demo programs
			x	SUNWCnet	SUNWbnur	Networking UUCP Utilities, (Root)
			x	SUNWCnet	SUNWbnuu	Networking UUCP Utilities, (Usr)
			x	SUNWCdhcp	SUNWdhcsr	BOOTP/DHCP Server Services, (Root)
			x	SUNWCdhcp	SUNWdhcsu	BOOTP/DHCP Server Services, (Usr)
			x		SUNWfac	Framed Access Command Environment
			x		SUNWfnsx5	FNS Support For X.500 Directory Context
			x		SUNWi1of	ISO-8859-1 (Latin-1) Optional Fonts
			x		SUNWi2of	X11 ISO-8859-2 optional fonts
			x		SUNWi4of	X11 ISO-8859-4 optional fonts
			x		SUNWi5of	X11 ISO-8859-5 optional fonts
			x		SUNWi7of	X Windows ISO-8859-7 optional fonts
			x		SUNWi9of	X Windows ISO-8859-9 optional fonts
			x		SUNWisolc	XSH4 conversion for ISO Latin character sets
			x	SUNWCown	SUNWoladd	OPEN LOOK Alternate Desktop Demos
			x	SUNWCown	SUNWoldem	OPEN LOOK demo programs
			x	SUNWCown	SUNWoldim	OPEN LOOK demo images
			x	SUNWCpex	SUNWpexcl	PEX Runtime Client Library
			x	SUNWCpex	SUNWpexsv	PEX Runtime Server Extension

TABLE A–4 x86: Detailed Software Group Contents (List of Packages) *(continued)*

C	U	D	E	Cluster Name	Package Name	Package Description
			x		SUNWpldte	Eastern European locale support
			x		SUNWploc1	Supplementary Partial Locale
			x		SUNWplow1	OpenWindows enabling for Supplementary Partial Locales
			x	SUNWCpppk	SUNWpppk	PPP/IP and IPdialup Device Drivers
			x		SUNWsadml	Solstice Launcher
			x		SUNWsutl	Static Utilities
			x	SUNWCown	SUNWxwacx	AccessX client program
			x		SUNWxipsu	XIL Platform Specific Device Pipeline Libraries (/usr)
			x	SUNWCown	SUNWxwdem	X Window System demo programs
			x	SUNWCown	SUNWxwdim	X Window demo images
			x		SUNWxwdxm	DPS motif library
			x	SUNWCown	SUNWxwfa	X Window System Font Administrator
			x	SUNWCfs	SUNWxwfs	Font server
			x		SUNWypr	NIS Server for Solaris (root)
			x		SUNWypu	NIS Server for Solaris (usr)

APPENDIX **B**

Solaris CD Layout

Slice 0 on the Solaris 2.6 CD contains the installation tools to set up Solaris installations, plus it contains the Solaris software itself. Figure B–1 shows the Solaris 2.6 CD layout.

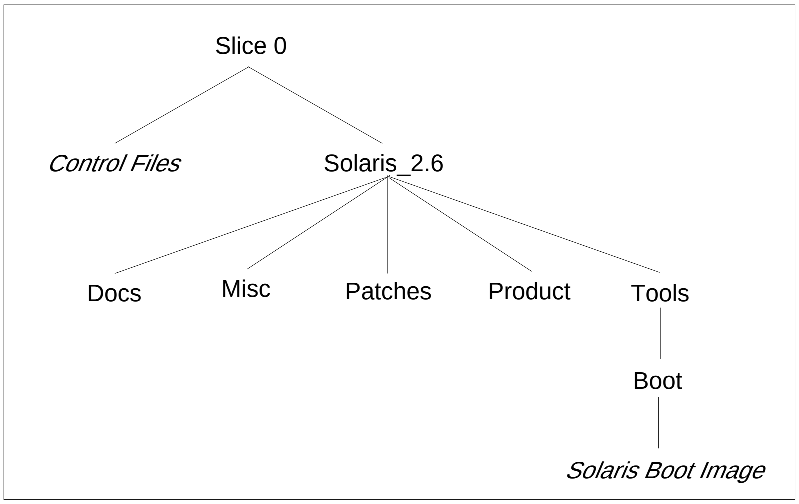


Figure B–1 Solaris 2.6 CD Layout

The control files at the top level on slice 0 are used by the installation software. The control files include `.cdtoc`, `.slicemapfile`, and `.install_config`.

The `Solaris_2.6` directory at the top level on slice 0 contains all the tools, software, and configuration necessary to install, at a minimum, the Solaris 2.6 software product. It contains the following directories:

- `Docs` – Contains release notes for the Solaris software.

- `Misc` – Contains the `jumpstart_sample` directory, which is a sample JumpStart directory that includes a `rules` file, profiles, begin scripts, and finish scripts. This directory has replaced the `auto_install_sample` directory on previous Solaris CDs.

- `Patches` – Contains all of the Solaris patches available at the time the Solaris CD was created.

- `Product` – Contains the Solaris packages and control files. The format of this directory is exactly the same as the product directory (for example, `Solaris_2.5`) on previous Solaris CDs.

- `Tools` – Contains the Solaris installation tools, which include `add_install_client`, `setup_install_server`, and `rm_install_client`. On previous Solaris CDs, these tools were at the top level on slice 0. The `Tools` directory also contains the `Boot` subdirectory that contains the Solaris boot image.

APPENDIX **C**

Platform Names and Groups

Table C–1 lists the platform names and groups of various hardware platforms. You may need this information when preparing a system to install Solaris software.

On a running system, you can also use the `uname -i` command to determine a system's platform name or use the `uname -m` command to determine a system's platform group.

TABLE C–1 Platform Names and Groups

System	Platform Name	Platform Group
x86 based	i86pc	i86pc
SPARCstation™ 1	SUNW,Sun_4_60	sun4c
SPARCstation 1+	SUNW,Sun_4_65	sun4c
SPARCstation SLC™	SUNW,Sun_4_20	sun4c
SPARCstation ELC™	SUNW,Sun_4_25	sun4c
SPARCstation IPC™	SUNW,Sun_4_40	sun4c
SPARCstation IPX™	SUNW,Sun_4_50	sun4c
SPARCstation 2	SUNW,Sun_4_75	sun4c
SPARCserver™ 1000	SUNW,SPARCserver-1000	sun4d

TABLE C–1 Platform Names and Groups *(continued)*

System	Platform Name	Platform Group
SPARCcenter 2000	SUNW,SPARCcenter-2000	sun4d
SPARCstation 5	SUNW,SPARCstation-5	sun4m
SPARCstation 10	SUNW,SPARCstation-10	sun4m
SPARCstation 10SX	SUNW,SPARCstation-10,SX	sun4m
SPARCstation 20	SUNW,SPARCstation-20	sun4m
SPARCstation LX	SUNW,SPARCstation-LX	sun4m
SPARCstation LX+	SUNW,SPARCstation-LX+	sun4m
SPARCclassic™	SUNW,SPARCclassic	sun4m
SPARCclassic X	SUNW,SPARCclassic-X	sun4m
SPARCstation Voyager™	SUNW,S240	sun4m
SPARCstation 4	SUNW,SPARCstation-4	sun4m
Ultra™ 1 systems	SUNW,Ultra-1	sun4u
Ultra Enterprise 1 systems	SUNW,Ultra-1	sun4u
Ultra 30	SUNW,Ultra-30	sun4u
Ultra 2 systems	SUNW,Ultra-2	sun4u
Ultra Enterprise 2 systems	SUNW,Ultra-2	sun4u
Ultra Enterprise 150	SUNW,Ultra-1	sun4u
Ultra 450	SUNW, Ultra-4	sun4u
Ultra Enterprise 450	SUNW,Ultra-4	sun4u
Ultra Enterprise 3000, 4000, 5000, 6000, 10000	SUNW,Ultra-Enterprise	sun4u

APPENDIX **D**

Sample Custom JumpStart Setup

The following example shows how you would set up custom JumpStart installations for a fictitious site. The example includes SPARC and x86 based systems.

Sample Site Setup

Figure D–1 shows the site setup for this example.

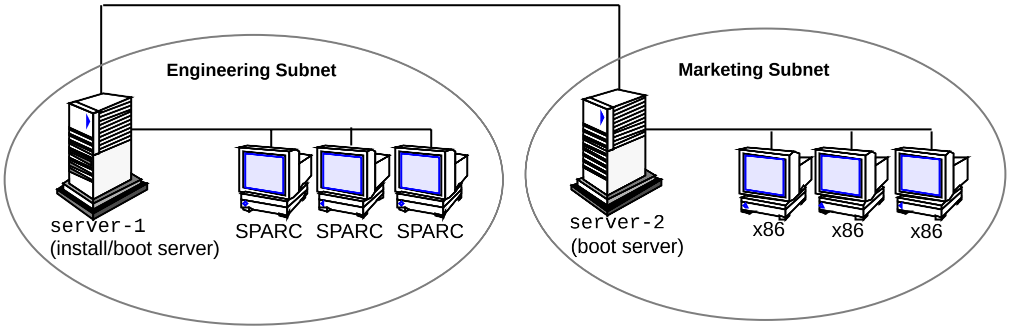


Figure D–1 Sample Site Setup

At this fictitious site:

- The engineering group is on its own subnet. This group uses Sun SPARCstation 5 systems for software development.
- The marketing group is on its own subnet. This group uses Dell® 466 systems for running word processing, spreadsheets, and other office tools.

- The site uses NIS. The Ethernet addresses, IP addresses, and host names of the systems are preconfigured in the NIS maps. The subnet mask, date and time, and geographic region for the site are also preconfigured in the NIS maps.

Note - The peripheral devices for the marketing systems are preconfigured in the `sysidcfg` file located on the Solaris boot diskette.

- Both the engineering and marketing systems will install Solaris software over the network.

Create an Install Server

Because the groups need to install Solaris software over the network, you make `server-1` an install server for both groups. You use the `setup_install_server` command to copy the x86 and SPARC Solaris CDs to `server-1`'s local disk (in the `/export/install` directory).

Also, because you must copy a Solaris CD to an empty directory, you copy the Solaris CD images to separate directories (the `x86_2.6` and `sparc_2.6` directories).

You insert the x86 Solaris CD into `server-1`'s CD-ROM drive.

```
server-1# cd /CD_mount_point/Solaris_2.6/Tools
server-1# ./setup_install_server /export/install/x86_2.6
```

You insert the SPARC Solaris CD into `server-1`'s CD-ROM drive.

```
server-1# cd /CD_mount_point/Solaris_2.6/Tools
server-1# ./setup_install_server /export/install/sparc_2.6
```

Create a Boot Server for Marketing Systems

Systems cannot boot from an install server on a different subnet, so you make `server-2` a boot server on the marketing group's subnet. You use the setup_install_server command to copy the boot software from the x86 Solaris CD to server-2's local disk (in the /export/boot directory).

You insert the x86 Solaris CD into server-2's CD-ROM drive.

```
server-2# cd /CD_mount_point/Solaris_2.6/Tools
server-2# ./setup_install_server -b /export/boot
```

In the setup_install_server command,

–b

Specifies that setup_install_server will to copy the boot information from the Solaris CD to the directory named /export/boot.

Create a JumpStart Directory

Now that you have the install and boot servers set up, you set up a JumpStart directory on server-1 (any system on the network could have been used). This directory will hold files necessary for a custom JumpStart installation of Solaris software. You set up this directory by copying the sample directory from one of the Solaris CD images that has been put in /export/install.

```
server-1# mkdir /jumpstart
server-1# cp -r /export/install/sparc_2.6/Solaris_2.6/Misc/jumpstart_sample
/jumpstart
```

Share the JumpStart Directory

To make the rules file and profiles accessible to systems on the network, you share the /jumpstart directory. This is done by adding the following line to the /etc/dfs/dfstab file:

```
share -F nfs -o ro,anon=0 /jumpstart
```

Then, at the command line, you use the shareall command.

```
server-1# shareall
```

Create the Engineering Group’s Profile

For the engineering systems, you create a file named eng_prof in the /jumpstart directory. The eng_prof file has the following entries, which define the Solaris software to be installed on systems in the engineering group.

```
[1] install_type  initial_install
[2] system_type   standalone
[3] partitioning  default
[4] cluster       SUNWCprog
[5] filesys       any 50 swap
```

Create the Marketing Group’s Profile

For the marketing systems, you create a file named marketing_prof in the /jumpstart directory. The marketing_prof file has the following entries, which define the Solaris software to be installed on systems in the marketing group.

```
[6] install_type  initial_install
[7] system_type   standalone
```

[1] Specifies that the installation will be treated as an initial installation, as opposed to an upgrade.
[2] Specifies that the engineering systems are standalone systems.
[3] Specifies that the JumpStart software uses default disk partitioning for installing Solaris software on the engineering systems.
[4] Specifies that the developer’s software group will be installed.
[5] Specifies that each system in the engineering group will have 50 Mbytes of swap space.
[6] Specifies that the installation will be treated as an initial installation, as opposed to an upgrade.
[7] Specifies that the installation will be treated as an initial installation, as opposed to an upgrade.

(continued)

(Continuation)

```
[8] partitioning  default
[9] cluster       SUNWCuser
[10] package       SUNWaudmo
```

Update the rules File

Now you must add rules to the `rules` file. The Solaris installation program uses the rules to select the correct installation (profile) for each system during a custom JumpStart installation.

At this site, each department is on its own subnet and network address. The engineering department is on subnet 255.222.43.0, and marketing is on 255.222.44.0. You can use this information to control how the engineering and marketing systems are installed. In the `/jumpstart` directory, you edit the `rules` file, delete all of the example rules, and enter:

```
network 255.222.43.0 - eng_prof -
network 255.222.44.0 - marketing_prof -
```

Basically, these rules state that systems on the 255.222.43.0 network will be installed using the `eng_prof` profile, and the systems on the 255.222.44.0 network will be installed using the `marketing_prof` profile.

Note - These are sample rules in which you can use a network address to identify which systems will be installed with the `eng_prof` and `marketing_prof`, respectively. You could also have chosen to use host names, memory size, or model type as the rule keyword. See Table 8–2 for a complete list of keywords you can use in a `rules` file.

[8] Specifies that the JumpStart software will use default disk partitioning for installing Solaris on the marketing systems.

[9] Specifies that the end user software group is to be installed.

[10] Specifies that the audio demo software package is to be added to each system.

Check the rules File

After the rules and profile files are properly set up, you run the check script to verify the files.

```
server-1# cd /jumpstart
server-1# ./check
```

If check doesn't find any errors, it creates the rules.ok file.

Set Up Engineering Systems to Install Over the Network

After setting up the /jumpstart directory and appropriate files, you use the add_install_client command on the install server (server-1) to set up the engineering systems to install Solaris from the install server.

Note - Host Manager could have been used instead of add_install_client to set up the systems.

```
server-1# cd /export/install/sparc_2.6/Solaris_2.6/Tools
server-1# ./add_install_client -c server-1:/jumpstart host_eng1 sun4m
server-1# ./add_install_client -c server-1:/jumpstart host_eng2 sun4m

.
.
.
.
```

In the add_install_client command,

–c Specifies the server (server-1) and path (/jumpstart) to the JumpStart directory.

host_eng1 Is the name of a system in the engineering group.

host_eng2 Is the name of another system in the engineering group.

sun4m Specifies the platform group of the systems that will use server-1 as an install server. (This is the platform group for Sun SPARCstation 5 systems.)

Set Up Marketing Systems to Install Over the Network.

Next, you use the add_install_client command on the boot server (server-2) to set up the marketing systems to boot from the boot server and install Solaris from the install server (server-1).

```
server-2# cd /marketing/boot-dir/Solaris_2.6/Tools
server-2# ./add_install_client -s server-1:/export/install/x86_2.6
-c server-1:/jumpstart host_mkt1 i86pc
server-2# ./add_install_client -s server-1:/export/install/x86_2.6
-c server-1:/jumpstart host_mkt2 i86pc
.
.
.
```

In the add_install_client command,

–s Specifies the install server (server-1) and the path to the Solaris software (/export/install/x86_2.6).

–c Specifies the server (server-1) and path (/jumpstart) to the JumpStart directory.

host_mkt1 Is the name of a system in the marketing group.

`host_mkt` Is the name of another system in the marketing group.

`i86pc` Specifies the platform group of the systems that will use this boot server. (This is the platform name for x86 systems.)

Boot the Engineering Systems and Install Solaris Software

After all the setup is complete, you can boot the engineering systems by using the following `boot` command at the `ok` (PROM) prompt of each system.

```
ok boot net - install
```

The systems will automatically install the Solaris operating environment.

Boot the Marketing Systems and Install Solaris Software

Next, you can boot the marketing systems by using inserting the Solaris boot diskette and powering up the system. The systems will automatically install the Solaris operating environment.

APPENDIX **E**

Language and Locale Values

Table E–1 lists the language and locale values needed to set the `locale` keyword in a profile or when you are preconfiguring a language or locale. A *locale* determines how online information is displayed for a specific *language* or region, for example, date, time, spelling, and monetary value. So, if you want English as your language, but you also want to use the monetary values for Australia, you would choose the Australia locale value (`en_AU`) instead of the English language value (`C`).

TABLE E–1 Language and Locale Values

Language	Locale	Value
English	USA (7-bit)	C
	Australia	en_AU
	Canada	en_CA
	Ireland	en_IE
	United Kingdom	en_UK
	USA (8-bit)	en_US
Spanish		es
	Argentina	es_AR
	Bolivia	es_BO
	Chile	es_CL

TABLE E–1 Language and Locale Values *(continued)*

Language	Locale	Value
	Columbia	es_CO
	Costa Rica	es_CR
	Ecuador	es_EC
	Guatemala	es_GT
	Mexico	es_MX
	Nicaragua	es_NI
	Panama	es_PA
	Peru	es_PE
	Paraguay	es_PY
	El Salvador	es_SV
	Uruguay	es_UY
	Venezuela	es_VE
German		de
	Switzerland	de_CH
	Austria	de_AT
French		fr
	Belgium	fr_BE
	Canada	fr_CA
	Switzerland	fr_CH
Dutch		nl
	Belgium	nl_BE
Norwegian		no
Portuguese		pt

TABLE E–1 Language and Locale Values *(continued)*

Language	Locale	Value
	Brazil	pt_BR
Finnish		su
Danish		da
Italian		it
Japanese		ja
Korean		ko
Swedish		sv
Austrian		at
Czech		cz
Estonian		et
Estonian		et
Greek		el
Hungarian		hu
Israel (Hebrew)		il
Lithuanian		lt
Latvian		lv
Polish		pl
Russian		ru
Turkish		tr

APPENDIX **F**

Preserving Existing Operating Systems and User Data

Many x86 based systems are preinstalled with other operating systems such as MS-DOS, Microsoft Windows, Microsoft Windows NT, OS/2, or some other vendor's UNIX® implementation. It's common that the preinstalled operating system uses the entire disk on the system (on one `fdisk` partition) and contains data that you don't want to lose. Installing the Solaris operating environment on that `fdisk` partition will overwrite the current operating system and its associated user data. If you want to keep an existing operating system on the system and have it co-exist with the Solaris operating environment, you must create multiple `fdisk` partitions on the disk.

The following section describes procedures for preserving existing data on a single-disk system and making the existing operating system co-exist (MS-DOS or other) with the Solaris operating environment.

▼ How to Preserve Existing Operating Systems and User DataHow to Preserve Existing Operating Systems and User Data

1. **Make sure your existing operating system can co-exist with the Solaris operating environment.**

   The following table lists known problems of operating systems co-existing with the Solaris operating environment. This is not a complete list. Always check the existing operating system documentation for problems.

TABLE F–1 Operating Systems That Have Problems Co-Existing With Solaris

Operating System	Problem
Linux	Solaris fdisk partition is the same as Linux swap partition; you must delete the Linux swap partition(s) before you install Solaris software. See Linux documentation for instructions.

2. **Make sure you have media (CD-ROM or diskettes) containing the existing operating system.**

   Some preloaded systems do not automatically come with media for reinstalling the operating system. Create a copy of the operating system on media using vendor-supplied tools.

3. **Back up the existing operating system customizations and/or user data using the backup program of your choice.**

4. **Start the Solaris installation program using the instructions described in Planning Your Installation in this book.**

5. **When prompted, select the Solaris Interactive Installation option.**

   Do not select the Solaris Web Start option; Solaris Web Start does not let you create `fdisk` partitions.

6. **Create a Solaris `fdisk` partition for Solaris, and an `fdisk` partition for the existing operating system.**

   Because the existing operating system takes up the entire disk, you'll have the option to manually or automatically create a Solaris `fdisk` partition.

7. **When prompted during installation, select Manual Reboot.**

8. **After Solaris software is installed, halt the system from the command line.**

9. **Reboot the system using the media for the previously existing operating system.**

10. **If the non-Solaris `fdisk` partition that you created was 'Other', use the `fdisk` utility for that operating system and re-label the `fdisk` partition.**

11. **Reinstall the operating system on the non-Solaris `fdisk` partition using the operating system's installing software.**

Note - MS-DOS Users Only: The MS-DOS setup program will recognize that the MS-DOS partition is unformatted and prompt you for permission to format it. The setup message suggests that the setup program will format the entire disk (and overwrite the Solaris `fdisk` partition). However, the setup program only formats the MS-DOS `fdisk` partition and leaves the Solaris `fdisk` partition intact.

12. **Restore any backed-up data to the non-Solaris `fdisk` partition.**

    Use the appropriate operating system restore program to restore backed-up files.

13. **Reboot the system and change the active partition to the Solaris operating environment.**

    To ensure that the Solaris operating environment automatically runs each time you reboot the system in the future, you must make the Solaris `fdisk` partition the active partition. To do this, follow the instructions on the Boot Solaris screen after rebooting.

Glossary

AutoClient system A system type that caches all of its needed system software from an OS server. Because it contains no permanent data, an AutoClient is a field replaceable unit (FRU). It requires a small local disk for swapping and for caching its individual root (`/`) and `/usr` file systems from an OS server.

begin script A user-defined Bourne shell script, specified within the `rules` file, that performs tasks before the Solaris software is installed on the system. Begin scripts can be used only with custom JumpStart installations.

boot server A server that provides boot services to systems on the same subnet. A boot server is required if the install server is on a different subnet than the systems that need to install the Solaris software from it.

client A system connected to a network.

cluster A logical grouping of software packages. The Solaris software is divided into *software groups*, which are each composed of clusters and *packages*.

core A software group that contains the minimum software required to boot and run the Solaris operating environment on a system. It includes some networking software and the drivers required to run the OpenWindows environment; it does not include the OpenWindows software.

custom JumpStart installation A type of installation in which the Solaris software is automatically installed on a system based on a user-defined profile. You can create customized profiles for different types of users and systems.

derived profile
A profile that is dynamically created by a begin script during a custom JumpStart installation.

developer system support
A software group that contains the End User System Support software group plus the libraries, include files, man pages, and programming tools for developing software.

disk configuration file
A file that represents a structure of a disk (for example, bytes/sector, flags, slices). Disk configuration files enable you to use `pfinstall` from a single system to test profiles on different sized disks.

diskless client
A networked system that does not have its own disk, so it relies completely on an OS server for software and file storage. Diskless clients do not have to use the Solaris installation program, because they use the software that is already installed on an OS server.

domain
A part of the Internet naming hierarchy. It represents a group of systems on a local network that share administrative files.

domain name
The identification of a group of systems on a local network. A domain name consists of a sequence of component names separated by periods (for example: tundra.mpk.ca.us). As you read a domain name from left to right, the component names identify more general (and usually remote) areas of administrative authority.

end user system support
A software group that contains the core software group plus the recommended software for an end user, including OpenWindows and DeskSet software.

entire distribution
A software group that contains the entire Solaris release.

entire distribution plus OEM support
A software group that contains the entire Solaris release, plus additional hardware support for OEMs. This software group is recommended when instaling Solaris software on servers.

EISA
Extended Industry Standard Architecture. A type of bus on x86 based systems. EISA bus standards are “smarter” that ISA bus systems, and attached devices can be automatically detected when they have been configured via the “EISA configurator” program supplied with the system. See ISA.

/etc
A directory that contains critical system configuration files and maintenance commands.

/export

A file system on an OS server that is shared with other systems on a network. For example, the `/export` file system can contain the root file system and swap for diskless clients and the home directories for users on the network. Diskless clients rely on the `/export` file system on an OS server to boot and run.

fdisk partition

A logical partition of a disk drive dedicated to a particular operating system on x86 based systems. During the Solaris installation program, you must set up at least one Solaris fdisk partition on an x86 based system. x86 based systems are designed to support up to four different operating systems on each drive; each operating system must reside on a unique fdisk partition.

file server

A server that provides the software and file storage for systems on a network.

file system

A collection of files and directories that, when set into a logical hierarchy, make up an organized, structured set of information. File systems can be mounted from your local system or a remote system.

finish script

A user-defined Bourne shell script, specified within the `rules` file, that performs tasks after the Solaris software is installed on the system, but before the system reboots. Finish scripts can be used only with custom JumpStart installations.

host name

The name by which a system is known to other systems on a network. This name must be unique among all the systems within a given domain (usually, this means within any single organization). A host name can be any combination of letters, numbers, and minus sign (`-`), but it cannot begin or end with a minus sign.

initial installation option

An option presented during the Solaris installation program that overwrites the disk(s) with the new version of Solaris. The initial installation option is presented for upgradable systems; however, the disk(s) that contain the old version of Solaris software (including the local modifications) will be overwritten if you choose the initial installation option.

install server

A server that provides the Solaris CD image for other systems on a network to install from (also known as a *media server*). You can create an install server by copying the Solaris CD to the server’s hard disk or by mounting the Solaris CD from the server’s CD-ROM drive.

interactive installation

A type of installation where you have full hands-on interaction with the Solaris installation program to install the Solaris software on a system.

IP address

Internet protocol address. A unique number that identifies a networked system so it can communicate via Internet protocols. It consists of four numbers separated by periods. Most often, each part of the IP address is a number between 0 and 225; however, the first number must be less than 224 and the last number cannot be 0.

IP addresses are logically divided into two parts: the network (similar to a telephone area code), and the system on the network (similar to a phone number).

ISA

Industry Standard Architecture. A type of bus found in x86 based systems. ISA bus systems are “dumb” and provide no mechanism the system can use to detect and configure devices automatically. See EISA.

JumpStart directory

When using a profile diskette for custom JumpStart installations, the JumpStart directory is the root directory on the diskette that contains all the essential custom JumpStart files. When using a profile server for custom JumpStart installations, the JumpStart directory is a directory on the server that contains all the essential custom JumpStart files.

JumpStart installation

A type of installation in which the Solaris software is automatically installed on a system by using the factory-installed JumpStart software.

locale

A specific language associated with a region or territory.

MCA

Micro Channel Architecture. A type of bus on x86 based systems. The MCA bus provides fast data transfer within the computer, and attached devices can be automatically detected when they have been configured using the reference disk provided by the manufacturer. The MCA bus is not compatible with devices for other buses.

media server

See *install server*.

mount

The process of making a remote or local file system accessible by executing the `mount` command. To mount a file system, you need a mount point on the local system and the name of the file system to be mounted (for example, `/usr`).

mount point

A directory on a system where you can mount a file system that exists on the local or a remote system.

name server

A server that provides a name service to systems on a network.

name service

A distributed network database that contains key system information about all the systems on a network, so the systems can communicate with each other. With a name service, the system information can be maintained, managed, and accessed on a network-wide basis. Sun supports the following name services: NIS (formerly YP) and NIS+. Without a name service, each system has to maintain its own copy of the system information (in the local `/etc` files).

network installation

A way to install software over the network—from a system with a CD-ROM drive to a system without a CD-ROM drive. Network installations require a *name server* and an *install server*.

networked systems

A group of systems (called hosts) connected through hardware and software, so they can communicate and share information; referred to as a local area network (LAN). One or more servers are usually needed when systems are networked.

NIS

Network Information Service. A type of name service that is standard on SunOS 3.x, 4.x, and Solaris 1.x systems.

NIS+

Network Information Service, Plus. The replacement for NIS that provides automatic information updating and adds security features such as authorization and authentication. NIS+ is the standard on Solaris 2.x systems.

non-networked systems

Systems that are not connected to a network or do not rely on other systems.

/opt

A file system that contains the mount points for third-party and unbundled software.

OS server

A system that provides services to systems on a network. To serve diskless clients, an OS server must have disk space set aside for each diskless client's root file system and swap space (`/export/root`, `/export/swap`). To serve AutoClient systems, an OS server must provide everything except the individual root (/) and /usr file systems required for swapping and caching.

package A functional grouping of files and directories that form a software application. The Solaris software is divided into *software groups*, which are each composed of *clusters* and packages.

platform group A vendor-defined grouping of hardware platforms for the purpose of distributing specific software. Examples of valid platform names are i86pc, sun4c.

platform name The output of the `uname -i` command. For example, the platform name for the SPARCstation IPX is SUNW,Sun_4_50.

profile A text file that defines how to install the Solaris software (for example, which software group to install) on a system. Every rule specifies a profile to define how a system will be installed when the rule is matched. You usually create a different profile for every rule; however, the same profile can be used in more than one rule. See *rules file*.

profile diskette A diskette that contains all the essential custom JumpStart files in its root directory (JumpStart directory).

profile server A server that contains all the essential custom JumpStart files in a JumpStart directory.

/ (root) The file system at the top of the hierarchical file tree on a system. The root directory contains the directories and files critical for system operation, such as the kernel, device drivers, and the programs used to start (boot) a system.

rule A series of values that assigns one or more system attributes to a profile.

rules file A text file that should contain a rule for each group of systems (or single systems) that you want to automatically install. Each rule distinguishes a group of systems based one or more system attributes, and it links each group to a profile, which is a text file that defines how the Solaris software will be installed on each system in the group. See *profile*.

rules.ok file A generated version of the `rules` file. It is required by the custom JumpStart installation software to match a system to a profile. You *must* use the `check` script to create the `rules.ok` file.

slice An area on a disk composed of a single range of contiguous blocks. A slice is a physical subset of a disk (except for slice 2, which by convention represents the entire disk). A disk can be divided into eight slices. Before you can create a file system on a disk, you must format it into slices.

Solaris CD image The Solaris software that is installed on a system. It can be available from the Solaris CD or a copy of the Solaris CD on an install server’s hard disk.

Solaris installation program (1) A menu-driven, interactive program that enables you to set up a system and install the Solaris software on it. (2) Any part of the software that is used to install the Solaris software on a system.

software group A logical grouping of the Solaris software (clusters and packages). During a Solaris installation, you can install one of the following software groups: core, end user system software, developer system support, or entire distribution.

standalone system A system that has its own root (`/`) file system, swap space, and `/usr` file system, which reside on its local disk(s); it does not require boot or software services from an OS server. A standalone system can be connected to a network.

subnet A working scheme that divides a single logical network into smaller physical networks to simplify routing.

subnet mask A bit mask, which is 32 bits long, used to determine important network or system information from an IP address.

swap space Disk space used for virtual memory storage when the system does not have enough system memory to handle current processes. Also known as the `/swap` or `swap` file system.

system types The different ways a system can be set up to run the Solaris software. Valid system types are: standalone system, diskless client, AutoClient system, OS server. However, the only system types that are covered in this document are standalone system and OS server because these are the only system types that can be installed using the Solaris installation program.

time zone Any of the 24 longitudinal divisions of the earth’s surface for which a standard time is kept.

upgrade option An option presented during the Solaris installation program. The upgrade procedure merges the new version of Solaris with existing files on your disk(s), and it saves as many local modifications as possible since the last time Solaris was installed.

/usr A file system on a standalone system or server that contains many of the standard UNIX programs. Sharing the large `/usr` file system with a server rather than maintaining a local copy minimizes the overall disk space required to install and run the Solaris software on a system.

/var A file system or directory (on standalone systems) containing system files that are likely to change or grow over the life of the system. These include system logs, `vi` files, mail files, and uucp files.

Volume Management A program that provides a mechanism to administer and obtain access to the data on CD-ROMs and diskettes.

Index