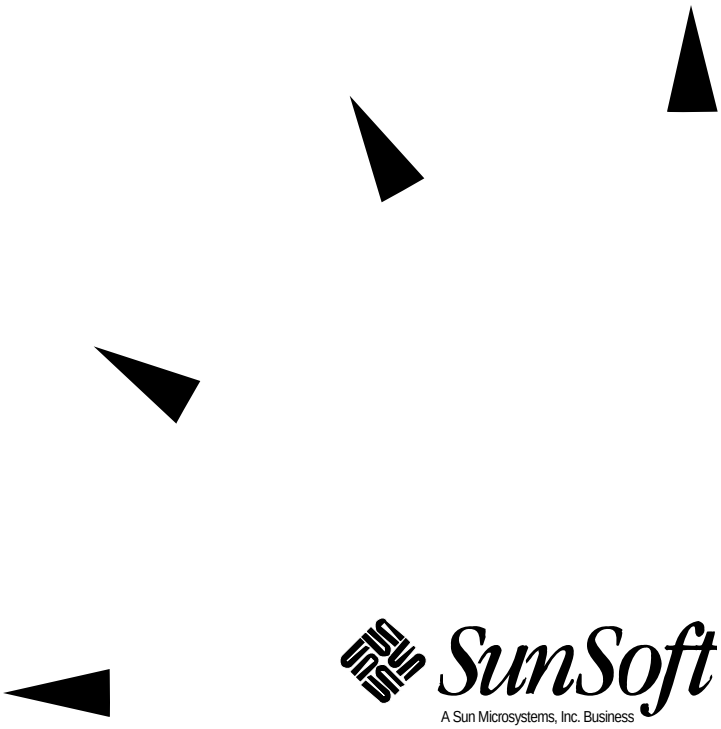
x86: Installing Solaris Software



2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

SunSoft
A Sun Microsystems, Inc. Business

Contents

About This Book

Who Can Use This Book

This book is for anyone installing the Solaris™ operating environment on networked or non-networked systems. Site policy and/or level of expertise will determine who can perform the tasks required to install Solaris software.

Don't Read the Entire Book!

Because this book covers different ways of installing the Solaris software to accommodate a variety of site needs, you do not need to read the entire book.

Read chapters 1 and 2 — they'll help you decide which method of installing is best for your site, and point you to a task map that tells you exactly what sections of the book to use.

How This Book Is Organized

This book is organized by tasks in the categories of before, during, and after installing Solaris software.

Note – This book does not include instructions for setting up system hardware or other peripherals. Setting up hardware and peripherals is described in the *x86 Device Configuration Guide*.

Before Installing Solaris

1 About Installing Solaris

2 Preparing to Install Solaris

3 Preparing to Install Solaris Over a Network

4 Preparing for Custom JumpStart™ Installations

5 Using Optional Custom JumpStart Features

6 Preparing a System for Upgrade

Installing Solaris

7 Installing Solaris: Interactive

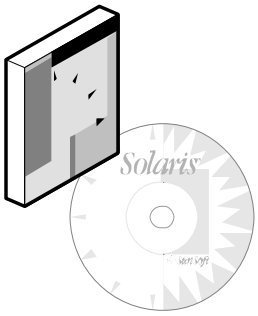
8 Installing Solaris: Custom JumpStart

After Installing Solaris

9 Completing an Upgrade

10 Where To Go After Installing Solaris

Related Information



You may need to refer to the following books or online information when installing Solaris software:

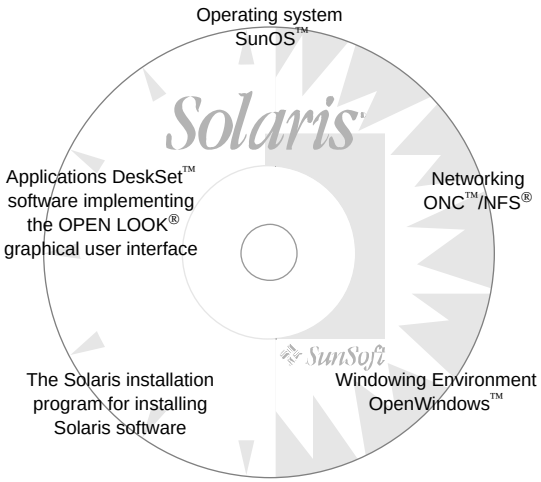
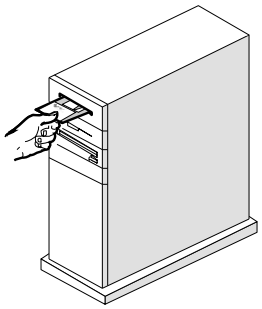
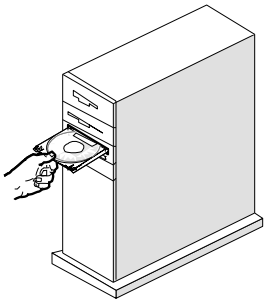
- *x86 Device Configuration Guide*
  Describes how to configure x86 devices before and after installing Solaris software.
- *x86 Hardware Compatibility Guide*
  Contains x86 hardware compatibility information.
- *x86 Notebook Supplement Guide*
  Contains configuration requirements and instruction for installing Solaris on a Notebook computer.
- *Installation Notes*
  Describes any late-breaking news about installing Solaris software including known problems.
- *Solstice AdminSuite 2.1 User's Guide*
  Describes applications such as Solstice™ Host Manager, which you can use if you're setting up network installations.
- *System Administration Guide, Volume I*
  Describes how to back up system files.

About Installing Solaris 1

Definition: Installing Solaris Software

As shown on the following pages, the process of installing Solaris software means copying it from the Solaris CD to a system's local disk.

1

Stage	Diagram	What Happens
1		You choose a Solaris CD for your hardware platform. The Solaris CD contains the SunOS operating system and other software.
2		You insert the Solaris boot diskette into the system's a: diskette drive.
3		You load the Solaris CD into the CD-ROM drive.

Stage	Diagram	What Happens
4	SunSoft x.x Multiple Device Boot Solaris/x86 Multiple Device Boot Menu Code Device Vendor Model/Desc Rev ========== 10 DISK CONNER CP330-360MB-3.5 3236 11 CD SONY CD-ROM CDU-8012 3.le 12 NET SMC/WD I/O=280 IRQ=10 Enter the boot device code:	After you power on the system, you choose to boot from the system's local CD-ROM drive, or from another system on the network. (Sample screen shown.)
5	Select the type of installation you want to perform 1 interactive 2 custom JumpStart Enter the number of your choice followed by the Enter key.	You choose a method for installing Solaris software, then the systems boots. During the booting phase, checks are performed on the hardware. (Sample screen shown.)
6	The Solaris Installation Program You are now interacting with the Solaris installation program. The program is divided into a series of short sections. At the end of each section, you will see a summary of the choices you've made, and be given the opportunity to make changes. Continue Exit Help	The Solaris installation program copies the Solaris software from the CD to the system's local disk. This is done *interactively* using a graphical (shown) or character interface, or *automatically* without user intervention.

1

Preparing to Install Solaris 2

This chapter guides you step-by-step through making decisions and completing the tasks required to prepare your system to install the Solaris software including:

Why You Should Not Ignore This Chapter

Successfully installing Solaris software requires a clear sense of what you're doing and why, or it can become difficult. This chapter provides all the information you need to determine the best way to install the Solaris software for your site. This chapter also directs you to specific chapters of this book you'll need.

2

Steps to Prepare to Install Solaris

Follow these steps before turning on your system.

1 Make sure your system has supported devices.

Check the *Solaris 2.5 Hardware Compatibility List*.

2 Make sure that the system's peripheral devices are properly configured.

Before installing Solaris software, you must configure peripheral devices so they do not conflict with each other, and so that Solaris software can access them. Configuring peripheral devices involves setting jumpers or running a software program under MS-DOS®. See the *x86 Device Configuration Guide* for detailed instructions.

Caution – If peripheral devices are not correctly configured before installing Solaris software, or if you are using unsupported devices, the Solaris installation program will fail.

3 If your system has the MS-DOS operating system, and you want to preserve the data, see Appendix G, "Preserving MS-DOS and Installing the Solaris Operating Environment."

4 Determine if your system is networked.

If your system is networked (connected to a network), an Ethernet connector or similar network adapter should be plugged into your system.

5 Determine your system type.

Before installing Solaris software, you must determine the *system type* which determines where the system gets important file systems. Using Figure 2-1 on page 8, choose a system type from the following lists:

Networked systems:

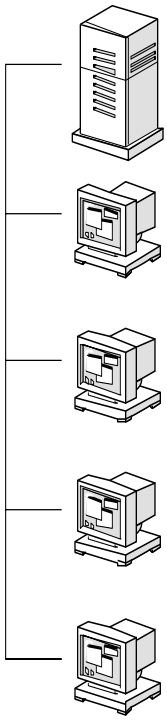
- *OS server*
- *Standalone system*
- *Dataless client*
- *Diskless client*
- *AutoClient™ system*

Non-networked systems:

- *Standalone system*

2

Networked systems



OS Server
Provides Solaris operating environment software including services and/or file systems for systems on the network. For dataless and diskless clients, OS servers provide the `/usr` file system. For diskless clients, an OS server provides root (/) and swap file systems. For AutoClient systems, an OS server provides all system software except the individual root (/) and /usr file system required for local swapping.

Standalone system
Has a local disk and does not require support from an OS server.

Dataless client
Has a local disk for root (/) and swap. Gets `/usr` file system from an OS server. Must mount `/usr` from a server.

Diskless client
Does not install Solaris software, but receives file services from an OS server, and does not have a local disk.

AutoClient system
Does not install Solaris software, but receives file services from an OS server. Has a local disk for swapping and caching its root (/) and `/usr` file systems.

Non-networked systems

Standalone systems
Have local disks and file systems.

Figure 2-1 System Types

Note – A standalone system in the Solaris operating environment applies to *both* networked and non-networked systems. Whether networked or non-networked, a standalone system has all of its Solaris software on local hard disk, and does not require services from another system.

6 If you are setting up diskless clients and/or AutoClient systems, and the OS server providing Solaris software is already installed with Solaris, do not go further in this book.

Diskless clients and AutoClient systems do not install Solaris software; instead, they receive file services from an OS server. If the OS server providing Solaris software already has Solaris software installed, refer to the *Solstice AdminSuite 2.1 User's Guide* for information on setting up diskless clients, or the *Solstice AutoClient 1.0 Administration Guide* for setting up AutoClients.

If you do not have an OS server set up with Solaris software, continue in this book.

7 If you are setting up dataless clients, determine if you have required software.

If you are setting up dataless clients to boot over the network from an OS server, you must have the Solstice™ Host Manager, which comes with the server software.

Note – SunSoft™ plans to remove support for the dataless client system type after Solaris 2.5. You can select this system type now, but in future releases you will need to select a different type.

8 Determine if you have required hardware.

For a detailed description of hardware requirements, see the *x86 Device Configuration Guide.*

Table 2-1 Hardware Requirements

CPU	Minimum Memory	Disk Interfaces	Buses	Device for Installing Solaris
An x86 system of the following type: • Intel or compatible 386@33MHz (minimum[1]) • Intel or compatible 486DX, 486DX2, 486SL, 486SX, 486DX4 • Intel Pentium AMD 486DX2-66, AMD 486DX2-80, AMD 486DX4-100	16 Mbytes (including 640K base memory)	SCSI, E-IDE, or IDE hard disk	• ISA • EISA • MCA • PCI • VLB	You must have a diskette drive, and one of the following devices for installing Solaris software: • Local SCSI or IDE/ATAPI CD-ROM drive • Remote SCSI or IDE/ATAPI CD-ROM drive available over a supported network interface[2] • Remote hard disk available over a supported network interface

1. A 486DK, 486DX2, or Pentium CPU is recommended for best performance.

2. See the *Solaris 2.5 Hardware Compatibility List* for information on supported network interfaces.

9 Plan disk space and memory requirements.

There are many variables involved in determining disk space and memory requirements. Determining physical memory is easy—the minimum is 16 Mbytes, but more is better. While a system requires 32 Mbytes of virtual memory (physical and swap file/slices), determining requirements beyond this is not easy; it is depends on the type of applications that are running. Determining disk space is also not so easy; it is depends on the following:

- System type selected (for example, OS server, diskless client, AutoClient system, standalone system, dataless client)
- Language selected (for example, Chinese, Japanese, Korean)
- Software group selected (see following table)

If You Want to Install This Software Group ...	Which Installs ...	Then You'll Need Approximately This Much Disk Space For Solaris Domestic CD ...
Core System Support	The minimum software required to boot and run Solaris software.	70 Mbytes
End User System Support	The core group plus the recommended software for an end user including OpenWindows and the DeskSet software.	160 Mbytes
Developer System Support	The end user software plus software for developing software including libraries, include files, man pages, and programming tools. Compilers and debuggers are not included.	260 Mbytes
Entire Distribution	The entire Solaris release (everything on the CD). Compilers and debuggers are not included.	310 Mbytes

- Software packages that are selected or deselected in the software group (for example, you select Chinese, but deselect the 200-Mbyte font package)
- Disks selected (for example, ten 104-Mbyte disks will waste more space trying to make things fit than a single 1-GByte disk)
- Solaris file system overhead or local file systems (for example, mail, printer spooling, users' personal file systems, swap space)
- Size of other applications that are running (for example, AnswerBooks, x86 compilers, FrameMaker®, IslandDraw®)

Note – If you want to use the DOS operating system for maintenance purposes and utilities, allocate between 5-10 Mbytes of space for the DOS partition (this could include hardware setup utilities, communication software, and even

some games). If you want to install and run Windows in the DOS partition, allocate an additional 20-40 Mbytes, plus whatever is needed for applications and data files.

Depending on which installation method you choose, you can perform a "dry run" to determine if you have enough disk space before actually installing Solaris software.

- Interactive – This method of installation lets you interactively select disks, locales, software, and lay out file systems; you can determine your requirements from a summary at the end of the session, and exit the program prior to installing Solaris software.
- Custom JumpStart – This method of installation lets you run the `pfinstall` command to test specific installation profiles without actually installing the Solaris software on a system.

10 Back up your system.

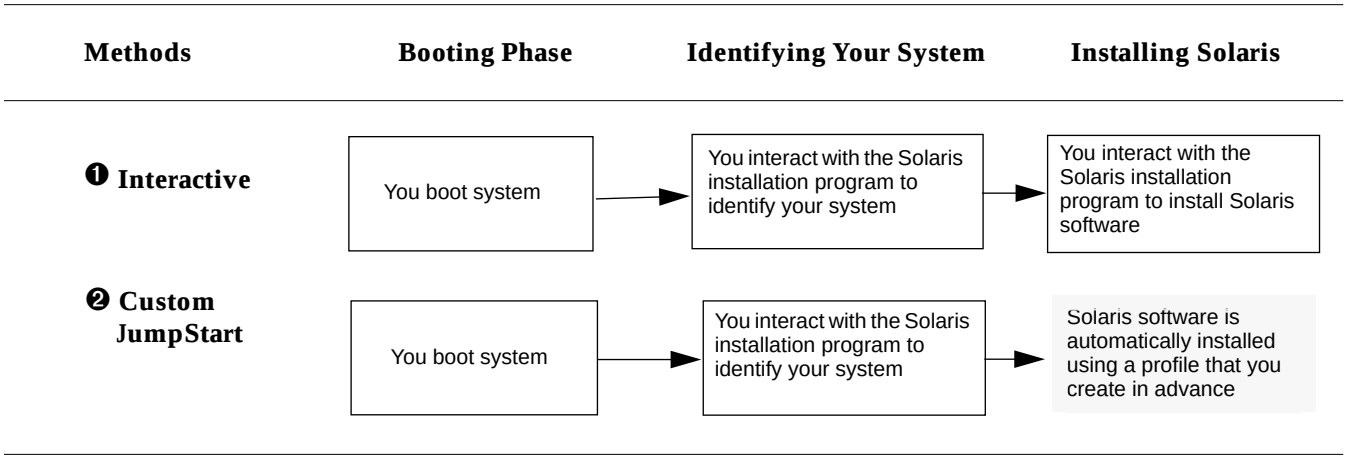
If your system has any files on it that you want to save, make sure you perform a backup. The safest way to back up files is to do a level 0 dump. If you do not have a backup procedure in place, see *System Administration Guide, Volume I* for instructions.

11 Choose a method for installing Solaris software.

There are two methods for installing Solaris software:

❶ *Interactive*—easy, hands-on method of installing Solaris software. The Solaris installation program guides you step-by-step through identifying your system and installing Solaris software. You're in control all the way!

❷ *Custom JumpStart* (formerly called auto-install)—for the advanced user with experience in Bourne shell scripting. By creating profiles and rules files, you can set up systems to automatically install Solaris software in different ways on different systems. This method requires up-front work before systems are turned on, but it's the most cost-effective way to install Solaris software for large, enterprise sites. You can even set up a hands-off installation where the user just boots the system and nothing more needs to be done!



12 Choose a task map and complete required tasks.

Choose the task map from the following pages that matches the method you've chosen for installing Solaris software: interactive or custom JumpStart.The task maps guide you through all the tasks you need to complete before turning on, booting, and installing Solaris software on a system.

2

Task Map: Interactive Installations

Activity	Description	For Instructions, Go To	
Gather information	Optional. Use worksheets to gather information that may need to be supplied during the Solaris installation program.	Appendix A, “Worksheets for the Solaris Installation Program”	page 111
Prepare system for upgrade	**Upgrade option only** Perform tasks such as backing up files and preserving local modifications.	Chapter , “Preparing a System for Upgrade”	page 111
Install Solaris Software	**From another system on the network** Set up systems (standalone, OS server, or dataless client) for network installations.	Chapter 3, “Preparing to Install Solaris Over a Network” Chapter 7, “Booting and Installing Solaris: Interactive”	page 17 page 117
	From local CD-ROM Boot and install Solaris software.	Chapter 7, “Booting and Installing Solaris: Interactive”	page 117

Figure 2-2 Task Map for Interactive Installations

Task Map: Custom JumpStart Installation

Activity	Description	For Instructions, Go To	
Prepare system for upgrade	**Upgrade option only** Perform tasks such as backing up files and preserving local modifications.	Chapter , “Preparing a System for Upgrade”	page 111
Set up system for custom JumpStart	Perform the following tasks: • Create a JumpStart directory • Enable clients to access the JumpStart directory • Create profiles • Create a `rules` file • Use check to validate the `rules` file	Chapter 4, “Preparing Custom JumpStart Installations”	page 47
Install Solaris Software	**From another system on the network** Set up systems (standalone, OS server, or dataless client) for network installations.	Chapter 3, “Preparing to Install Solaris Over a Network” Chapter 8, “Booting and Installing Solaris: Custom JumpStart”	page 17 page 125
	From local CD-ROM Boot and install Solaris software.	Chapter 8, “Booting and Installing Solaris: Custom JumpStart”	page 125

Figure 2-3 Task Map for Custom JumpStart Installations

2

Preparing to Install Solaris Over a Network

3

About Installing Solaris Over a Network

A typical way to install Solaris software is to use the installation program to copy the Solaris CD shipped with your system. However, it is uncommon at most sites for every system to have its own local CD-ROM drive. When a system does not have a local CD-ROM drive, you can perform a *network installation*. Network installation means that you install software over the network—from a system with a CD-ROM drive to a system without a CD-ROM drive.

3

Note – Instructions in this chapter are valid for either an x86 or SPARC server being used for network installations. An x86 server can provide the Solaris CD image for SPARC systems, and a SPARC server can provide the Solaris CD image for x86 systems.

Servers Required for Network Installation

As shown in Figure 3-1, systems that install Solaris software over the network require:

- *Install server* – A networked system with the CD-ROM drive that provides installation services for other 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 that boots 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.

Dataless clients also require:

- *OS server* – A system that provides Solaris operating environment software including services and/or file systems. For dataless clients, OS servers provide the `/usr` file system.

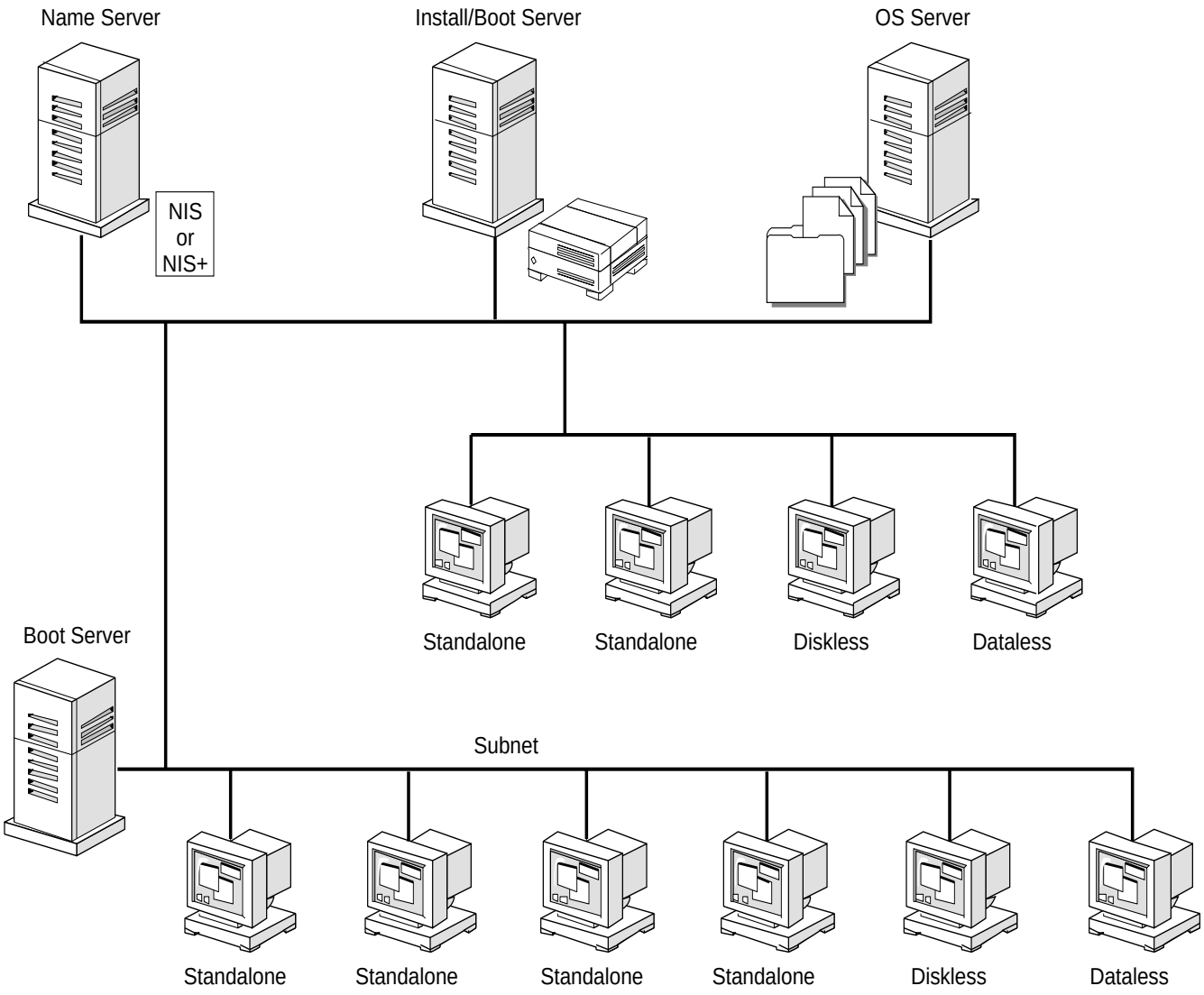


Figure 3-1 Network Installation Servers

3

Requirements for Hands-Off Network Installation

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

- Use the custom JumpStart installation method. (See Chapter 4, “Preparing Custom JumpStart Installations.”)
- Make sure all systems are properly configured in the name service. (Procedures in this chapter include information on how to add systems to the name service.)
- Preconfigure network information, such as the date, time, geographic region, site subnet mask, and language. By using the Solstice Host Manager to set up a server for network installations, you automatically preconfigure network information. This eliminates many prompts that are otherwise necessary to identify the system during an installation.

Commands You Should Know About

Table 3-1 shows programs available for use when setting up for network installation.

Table 3-1 Network Installation Commands

Program	Description
`setup_install_server`	A script that copies all or part of the Solaris CD onto a server's local disk. This enables you to perform network installations from the install server's disk instead of its CD-ROM drive. (Installing from the install server's disk is faster than installing from the server's CD-ROM drive.) See the `setup_install_server(1m)` man page for more information.
Host Manager	A graphical user interface that is available from within the Solstice AdminSuite (`solstice`). You can use Host Manager to specify naming services, system's IP and Ethernet addresses, and other information to be used for installations across a network.
`mount`	A command that shows mounted file systems, including the Solaris CD file system. See the `mount(1M)` man page for more information.
`uname -i`	A command for determining a system's platform name (for example, i86pc). This information is sometimes required during installation. See the `uname(1)`man page for more information.
`kdmconfig`	A command to run on the install server to preconfigure the keyboard, display, and other peripherals for a system being installed over a network. See the `kdmconfig(1M)` man page for more information.

3

Creating an Install Server

If you are installing systems over the network, you must have an install server—a system with a CD-ROM drive or with Solaris software copied to its local disk. This system will provide the installation services for systems on the network that do not have a local CD-ROM drive.

You can create an install server to install the following system types:

- OS server
- Standalone system
- Dataless client

This section describes how to:

- Set up a new system to be an install server.
- Create an install server by mounting the Solaris CD or by copying the Solaris CD to the install server's local disk.

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

- Create a separate boot server (required *only* if systems are not on same subnet as the install server).

▼ How to Set Up a New System to be an Install Server

Overview – Setting up a new system to be an install server involves:

- Installing the system hardware, including a CD-ROM drive
- Connecting the system to a network
- Installing Solaris software

Follow this procedure to create an install server

1. **Install the system hardware, including a CD-ROM drive.**
   Follow the hardware documentation for detailed information.

2. **Connect the system to the network.**
   Follow the hardware documentation for detailed information.

3. **Install Solaris software.**
   During an interactive installation, you are prompted to select a name service. Follow the guidelines in the next table:

Is a Name Service Running at Your Site ...	Then ...
Yes	Select NIS or NIS+, whichever your site uses, when prompted for a name service. After the Solaris software is installed, go to How to Create an Install Server on page 24.
No	Select None when prompted for a name service. After the Solaris software is installed, go to Step 4.

4. **If you want this system to be a name server, set up a name service after you have installed the Solaris software. For detailed information, see the** *NIS+ and DNS Setup and Configuration Guide***.**

The system is now set up so that you can make it an install server. Next, you must make a Solaris CD image available to other systems. To continue, go to the How to Create an Install Server on page 24.

▼ How to Create an Install Server

Overview – Creating an install server involves:

- Choosing a system with a CD-ROM drive to be the install server
- Mounting the Solaris CD
- Using the `setup_install_server` command to copy the Solaris CD to the install server's local disk (optional, but recommended)

Follow this procedure 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.
2. **Insert the Solaris CD into the CD-ROM drive.**
3. **Mount the Solaris CD (if needed) and change the directory to the mounted CD:**

If the Install Server Is ...	Then ...

Running Volume Management	`cd /cdrom/cdrom0/s0`
	or
	`cd /cdrom/cdrom0/s2`
	
	Note: Volume Management is running if the `/vol` directory on the system contains files. Systems running Solaris 2.0 or 2.1 do not have Volume Management.

Not running Volume Management	**1)** `mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom` or `mount -F hsfs -o ro /dev/dsk/c0t6d0p0 /cdrom` **2)** `cd /cdrom`

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

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

5. **Use the** `setup_install_server` **command to copy the contents of the Solaris CD to the install server's local disk.**

```
# ./setup_install_server install_dir_path
```

In this command,

install_dir_path	Specifies the directory where the Solaris CD image will be copied. You can substitute any directory path.

For example, the following command copies the Solaris CD image from the Solaris CD to the `/export/install` directory on the local disk:

```
./setup_install_server /export/install
```

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

6. **Type** `cd` *install_dir_path* **and press Return.**

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	Go to Task Complete on page 26.
Not on the same subnet as the system to be installed	Follow the procedure How to Create a Boot Server on a Subnet on page 27. You must complete this procedure when the install server is *not* on the same subnet as the system to be installed. After completing that procedure, go to Task Complete on page 26.

Task Complete

The install server is now created. Next, you must add information to the install server's configuration files so it recognizes the systems to be installed. To continue, go to How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client on page 31.

▼ 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 will use an install server on another subnet *requires* a separate boot server on its own subnet.

Overview – Creating a boot server involves:

- Choosing a system with a CD-ROM drive to be the boot server for the subnet
- Mounting the Solaris CD
- Using the `setup_install_server -b` command to copy required platform information from the Solaris CD to the local disk

Follow this procedure to set up a boot server on a subnet.

1. **On the system that will be the boot server for the subnet, log in and become root.**
   This system must have a CD-ROM drive or an NFS mount of a 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.)

2. **Determine your next step based on whether the boot server uses a local CD-ROM drive or an NFS mount of a Solaris CD image.**

If the Boot Server Uses ...	Then ...
Local CD-ROM drive	**1)** Insert the Solaris CD into the CD-ROM drive. **2)** Go to Step 3.
NFS mount of a Solaris CD image	**1)** `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)** `cd /mnt` **3)** Go to Step 4.

3. Mount the Solaris CD (if needed) and change the directory to the mounted CD:

If the Boot Server Is ...	Then ...

Running Volume Management	`cd /cdrom/cdrom0/s0`
	or
	`cd /cdrom/cdrom0/s2`
	
	Note: Volume Management is running if the `/vol` directory on the system contains files. Systems running Solaris 2.0 or 2.1 do not have Volume Management.

Not running Volume Management	**1)** `mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom` or `mount -F hsfs -o ro /dev/dsk/c0t6d0p0 /cdrom` **2)** `cd /cdrom`

4. **Use the `setup_install_server` command to set up the boot server for the subnet.**
   The `setup_install_server` command copies all supported platform information to the local disk.

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

   In this command,

`-b`	Specifies that the system will be set up as a boot server.
boot_dir_path	Specifies the directory where the platform information will be copied. You can substitute any directory path.

   For example, the following command copies platform information necessary for booting supported platforms over the network:

```
./setup_install_server -b /export/install
```

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.

Task Complete

The boot server is now set up to boot supported systems on a subnet. To continue, go to How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client on page 31.

3

Setting Up Servers for Network Installation

This section describes how to set up appropriate servers necessary to install a system over a network. The next table shows servers required for each system type you want to install.

If System Type You Are Installing Is [1] ...	Then You Need This Server Support ...
Standalone system	Install server
OS server	Install server
Dataless client	Install server and OS server

1. Systems also require a boot server if they are on a different subnet than the install server.

Note – When you install a server, you must allocate disk space required for the clients that server will support. Then, *after* the server is installed, you must use the Solstice Host Manager and add the platform support required by those clients.

▼ How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client

Once you have an install server set up, you are ready to use it to install Solaris software on other systems on the network. Before you can actually do that, however, you need to provide some basic system information about the systems that you are going to install. You do this by using the Solstice Host Manager.

Note – SunSoft plans to remove support for the dataless client system type after Solaris 2.5. You can set up this system type now, but in future releases you will need to change it to a different type (standalone, OS server, diskless client, or AutoClient system.)

The procedure to add system information about standalone systems, servers, and dataless clients to the install server can all be accomplished using the Solstice Host Manager. You simply fill out a form and apply the information you provide, and the Solstice Host Manager updates the appropriate files and name server maps or tables.

Overview – This procedure involves:

- Using the Solstice Host Manager to update the name service, to add information about the standalone, server, or dataless client to the install server's configuration files, and, if necessary, to set up a file server

Follow this procedure to set up the install server to install a standalone system, a server, or a dataless client on the network.

1. **On the install server, log in as a user in the administration group (group 14).**

Caution – If your system is part of the NIS+ name service, you must run the Solstice AdminSuite™ while logged in as a user in the NIS+ admin group. Otherwise, you will not have permission to update configuration information on the name server.

---

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

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

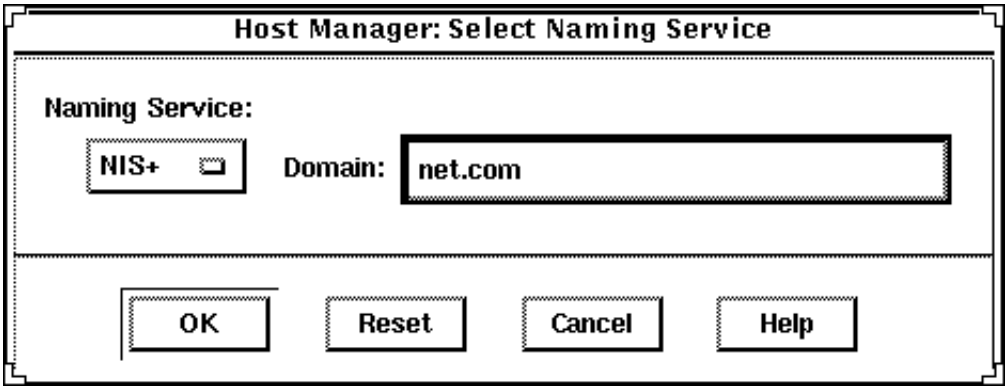
3. **After the Solstice AdminSuite main window appears, click on the Host Manager icon.**

Note – If your site uses the Domain Name Service (DNS), you will have to modify the `/etc/nsswitch.conf` file and create the `/etc/resolv.conf` file manually. For detailed information, see the *NIS+ and FNS Administration Guide*.

---

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. This will provide enough information for systems to boot, and to install Solaris software over the network.

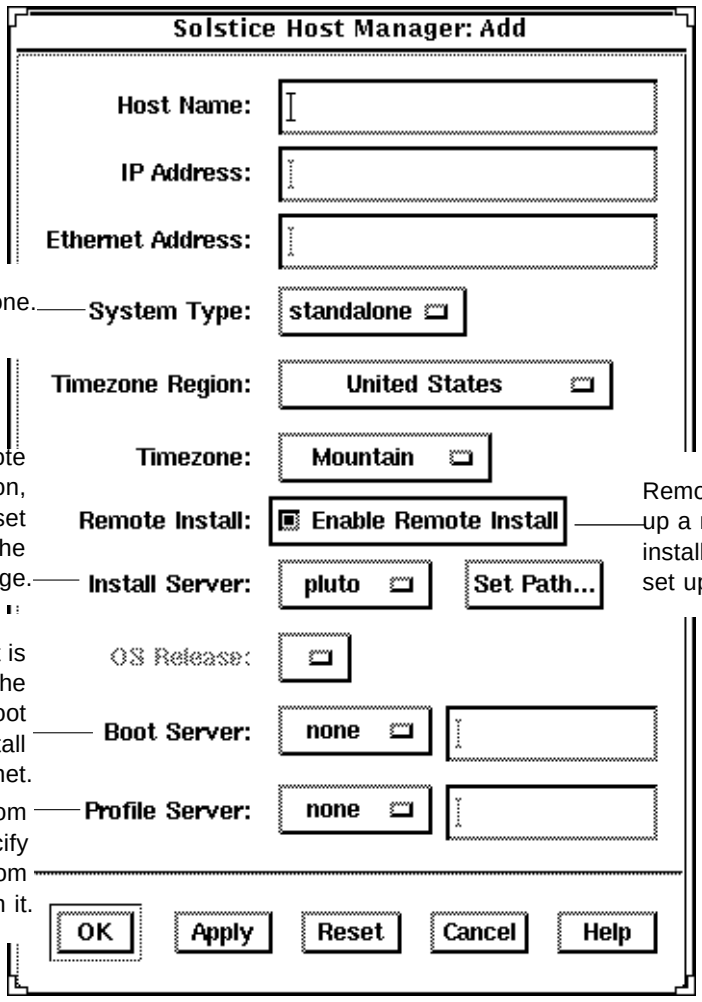


5. **On the Host Manager main window, choose Add... from the Edit menu.**

6. **Determine your next step based on what kind of system you want to install.**

System to Be Installed Is ...	Then ...
Standalone system	Go to Step 7.
OS server	Go to Step 10.
Dataless client	Go to Step 13.

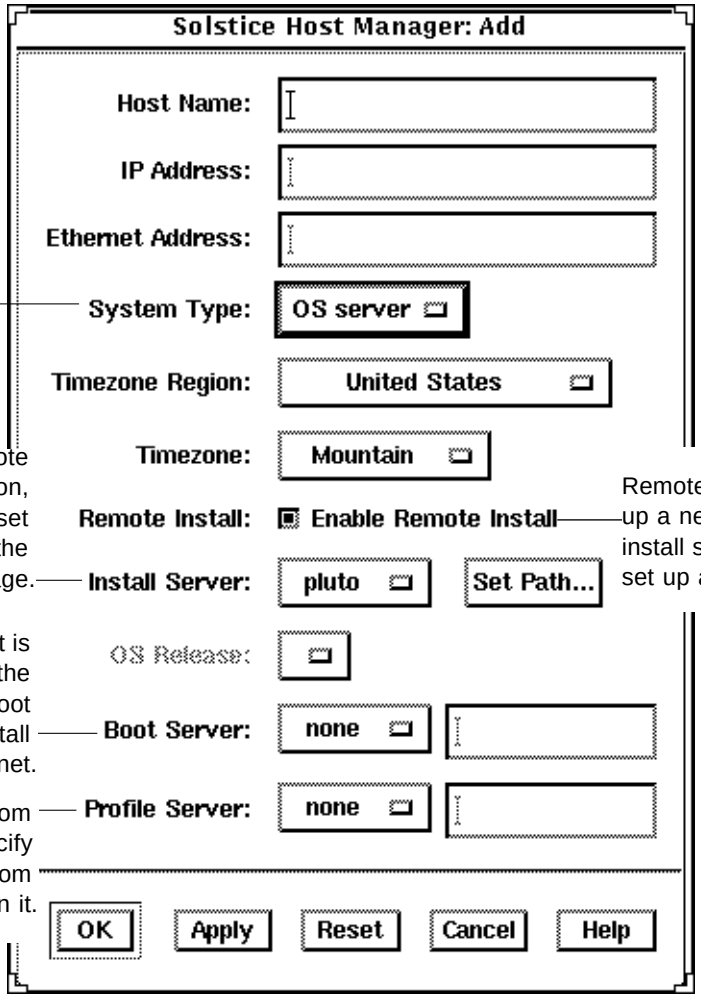
7. **On the Host Manager: Add Host screen, complete all fields and click on the OK or Apply button.**



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

9. **Go to Task Complete on page 37.**
   Step 10 through Step 13 are only required if you are adding an OS server or dataless client.

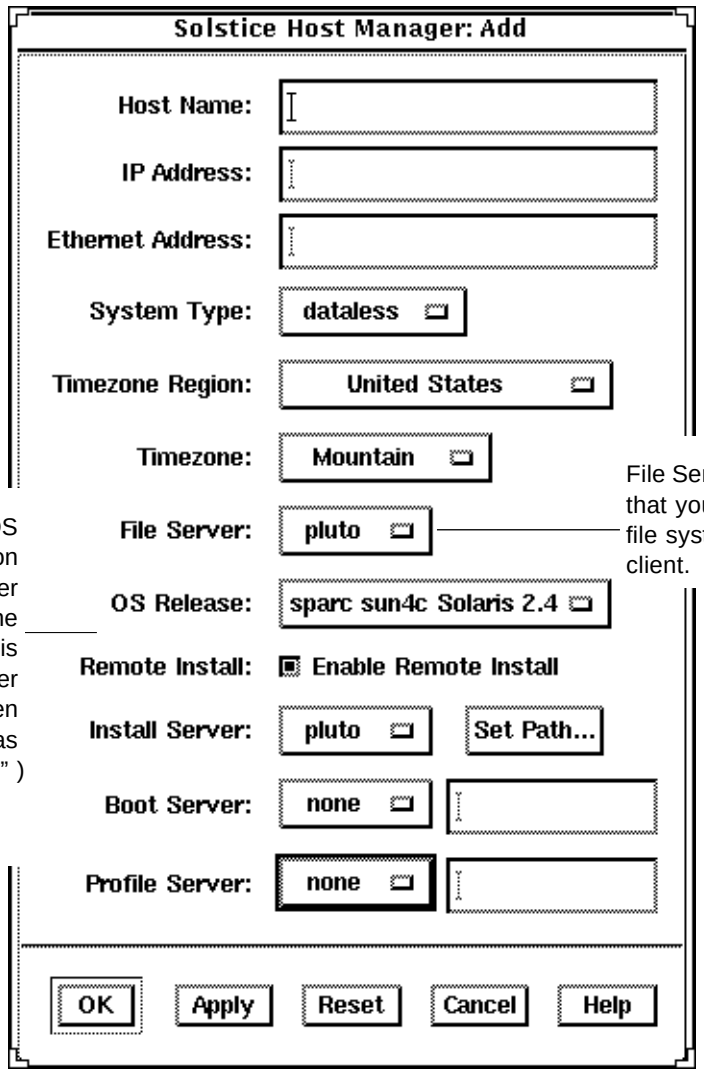
10. **On the Host Manager: Add Host screen, complete all fields and click on the OK or Apply button.**



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

12. **Go to Task Complete on page 37.**
    Step 13 is required only for adding dataless clients for network installation.

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



File Server - Specify the system that you want to provide `/usr` file systems for the dataless client.

OS Release - Select the OS Release you want installed on the dataless client. The proper OS support must reside on the install server. (You set up this support by using Host Manager after the server has been installed. This is referred to as "adding services.")

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

Task Complete

The standalone system, OS server, or dataless client is now added for network installation. You are now ready to boot and install over the network. To find the correct booting and installing procedure in this book, see the appropriate chapter for the installation you want to perform:

- Chapter 7, “Booting and Installing Solaris: Interactive”
- Chapter 8, “Booting and Installing Solaris: Custom JumpStart”

Using `add_install_client` *to Set Up Servers for Network Installation*

Install servers running versions of Solaris released prior to Solaris 2.4 software may not have access to the Solstice AdminSuite. You can install the Solstice products on Solaris 2.3 and 2.4 systems and follow the instructions in How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client on page 31. Alternatively, you can use the `add_install_client` command to set up the install server for network installations, as described in the next section.

3

▼ How to Use `add_install_client` to Set Up Servers for Network Installation

Use the `add_install_client` command to add information about the system to be installed to the boot server configuration files.

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

In this command,

`-c`	Specifies a JumpStart directory for custom JumpStart installations. This option and its arguments are required only for custom JumpStart installations.
server:*jumpstart_dir_path*	*server* is the host name of the server on which the JumpStart directory is located. *jumpstart_dir_path* is the absolute path of the JumpStart directory.
`-s`	Specifies the install server.
install_server:*install_dir_path*	*install_server* is the host name of the install server. *install_dir_path* is the absolute path name of the mounted Solaris CD or the directory that has the copy of the Solaris CD image.
host_name	Is the host name of the standalone system or the server where Solaris software will 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, “Platform Names and Groups.”)

For example, the following command copies boot information from Solaris CD image on an install server’s local disk in `/export/install`. `add_install_client` sets up the `/rplboot` directory on the local system,

which will be the boot server. The system that will be installed is named `watson`, and it is an x86 system. The platform group for an x86 system is i86pc.

```
./add_install_client -s install_server1:/export/install watson i86pc
```

Task Complete

The standalone system, server, or dataless client is now added for network installation. You are now ready to boot and install over the network. To find the booting and installing procedure in this book, see the appropriate chapter for the installation you want to perform:

- Chapter 7, “Booting and Installing Solaris: Interactive”
- Chapter 8, “Booting and Installing Solaris: Custom JumpStart”

3

Preconfiguring the Default Locale

When installing a localized version of Solaris software, you are prompted for the locale (the language) you want to use for the duration of the installation process. The locale you choose for installing Solaris software is also the default locale the installed version of Solaris will provide to the system's user.

You can set up a default locale of your choice. You do this by modifying the naming service (NIS or NIS+). After you modify the name service, as described in this section, the operating system will use this default locale for users. Also, the installation software will use the default locale for all future installations.

Choose from the following two procedures, depending on whether your site uses the NIS or NIS+ name service.

▼ How to Preconfigure Default Locale Using NIS

Overview – The procedure to preconfigure the default locale at sites using the NIS name service involves:

- Using the `vi` or text editor to edit files in the `/var/yp/Makefile` file
- Creating a locale file in the `/etc` directory
- Updating the NIS maps

Follow these instructions to set up a default locale for the system being installed over the network.

1. **On the name server, log in and become root and 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. **Edit the** `/var/yp/Makefile` **file.**
   a. **Add** `locale` **to the line starting with the word** `all`**.**
   b. **Add** `locale: locale.time` **on a new line.**

```
all: passwd group hosts 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.**

```
domain_name   locale
```

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

```
worknet.com fr
```

You can also use a host name instead of the domain name to specify the default locale for a particular host. For example, the following line specifies Korean to be the default locale for system named sherlock:

```
sherlock ko
```

See the table on page 44 for a list of valid locale values.

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

4. **Make the maps.**

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

Task Complete

Systems in the name service are now set up to use the default locale. The default language you have specified will be used during the installation and will also be the language provided to system users.

▼ How to Preconfigure Default Locale Using NIS+

Overview – The procedure to preconfigure the default locale at sites using the NIS+ name service involves:

- Using the `nistbladm` command to create a locale table and add information to it
- Updating the NIS+ tables

Note – The installation software is not translated to the Asian locales, so this procedure will not work for any of the Asian locales.

Follow these instructions to set up a default locale for a system being installed over the network. (This procedure assumes the NIS+ domain is set up. Setting up the NIS+ domain is documented in the *NIS+ and FNS 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.org_dir` **table by typing the following** `nistbladm` **command.**

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

In this command,

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

locale — Is the locale you want to use to install the system and the locale you want to come up on users' systems. The following table shows valid values for *locale*.

Language	Valid *locale* Values
Chinese	zh
English (Solaris default)	C
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Latin American	es
Swedish	sv
Taiwanese	zh_TW

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

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

Task Complete

Systems in the name service are now set up to use the default locale. The default language you have specified will be used during the installation and will also be the language provided to system users.

3

Preparing Custom JumpStart Installations

4

Definition: Custom JumpStart Installation

A custom JumpStart installation is 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.

Note – Appendix D, “Sample Custom JumpStart Installation” provides an example of how a fictitious site is prepared for custom JumpStart installations.

Note – Instructions in this chapter are valid for either an x86 or SPARC server that is being used for custom JumpStart installations. An x86 server can provide custom JumpStart files for SPARC systems, and a SPARC system can provide custom JumpStart files for x86 systems.

4

Reasons to Choose a Custom JumpStart Installation

You should choose custom JumpStart installations when you have to install Solaris software on:

- Many systems.
- Particular groups of systems.

For example, the following scenario would be ideal for performing custom JumpStart installations:

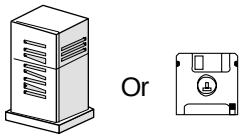
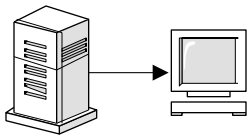
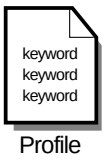
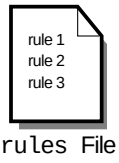
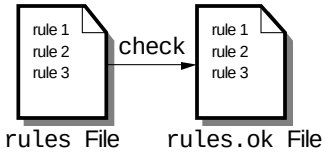
- You need to install the Solaris software on 100 new systems.
- The engineering group owns 70 out of the 100 new systems, and its systems must be installed as standalone systems with the developer software group.
- The marketing group owns 30 out of the 100 new systems, and its systems must be installed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

These installations would be time-consuming and tedious if you chose to perform an interactive installation on each system.

Tasks to Prepare for Custom JumpStart Installations

Table 4-1 shows the tasks that are required to prepare for custom JumpStart installations.

Table 4-1 Tasks to Prepare for Custom JumpStart Installations

Task		Description
Creating a JumpStart directory on a diskette or on a server		You must create a JumpStart directory to hold the custom JumpStart files. If you are going to use a diskette for custom JumpStart installations, see “Creating a JumpStart Directory on a Diskette” on page 53. If you are going to use a server for custom JumpStart installations, see “Creating a JumpStart Directory on a Server” on page 59.
Enabling all clients to access the JumpStart directory		When you use a server to provide the JumpStart directory, you can enable all clients to access the JumpStart directory. See “Enabling All Systems to Access the JumpStart Directory” on page 61 for detailed information.
Creating profiles		A profile is a text file used as a template by the custom JumpStart installation software. It defines how to install the Solaris software on a system (for example, initial or upgrade installation option, system type, disk partitioning, software group), and it is named in the `rules` file. See “Creating a Profile” on page 63 for detailed information.
Creating a `rules` file		The `rules` file is a text file used to create the `rules.ok` file. The `rules` file is a look-up table consisting of one or more rules that define matches between system attributes and profiles. See “Creating the rules File” on page 78 for detailed information.
Using `check` to validate the `rules` file		The `rules.ok` file is a generated version of the `rules` file, and 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. See “Using check to Validate the rules File” on page 89 for detailed information.

4

What Happens During a Custom JumpStart Installation

Figure 4-1 describes what happens after you boot a system to perform a custom JumpStart installation.

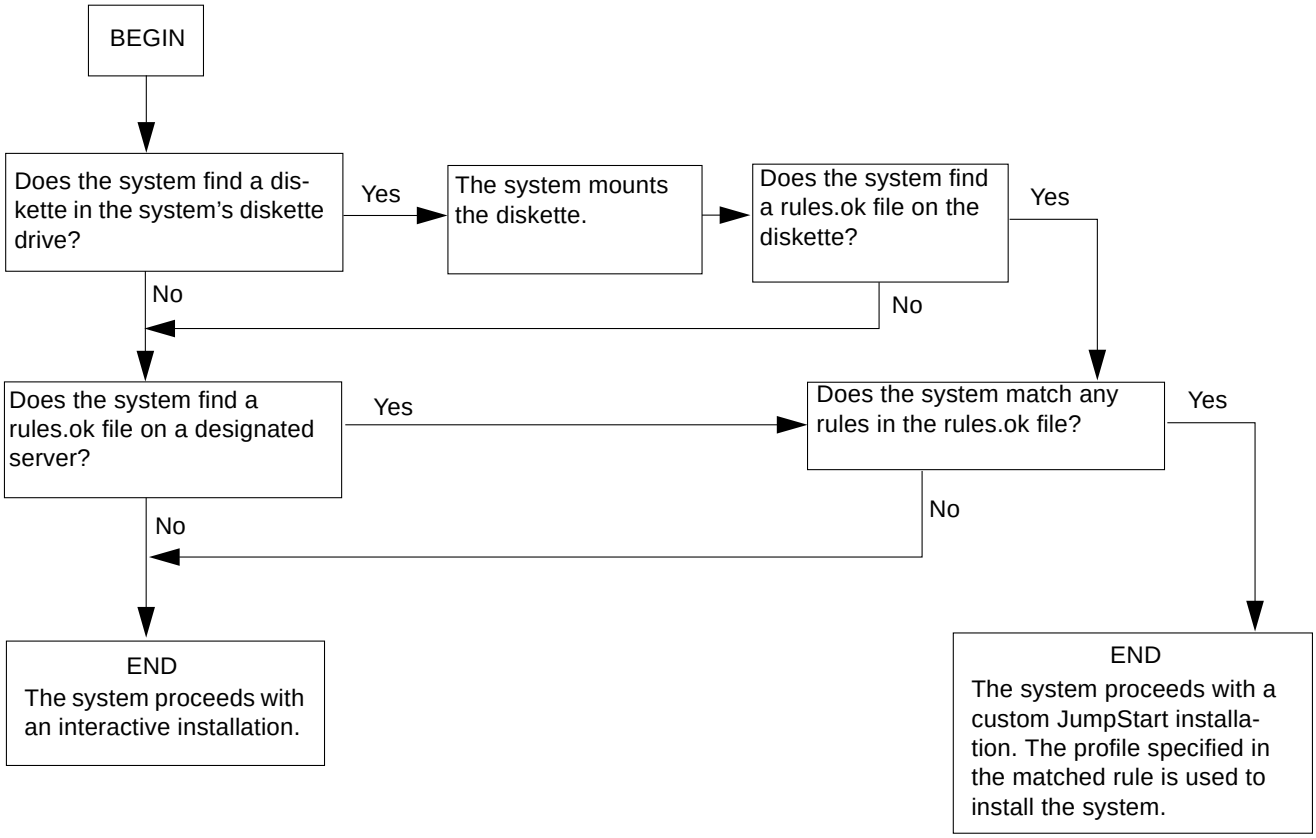


Figure 4-1 What Happens During a Custom JumpStart Installation

Figure 4-2 is an example of how a custom JumpStart installation works on a standalone, non-networked system using the system's diskette drive.

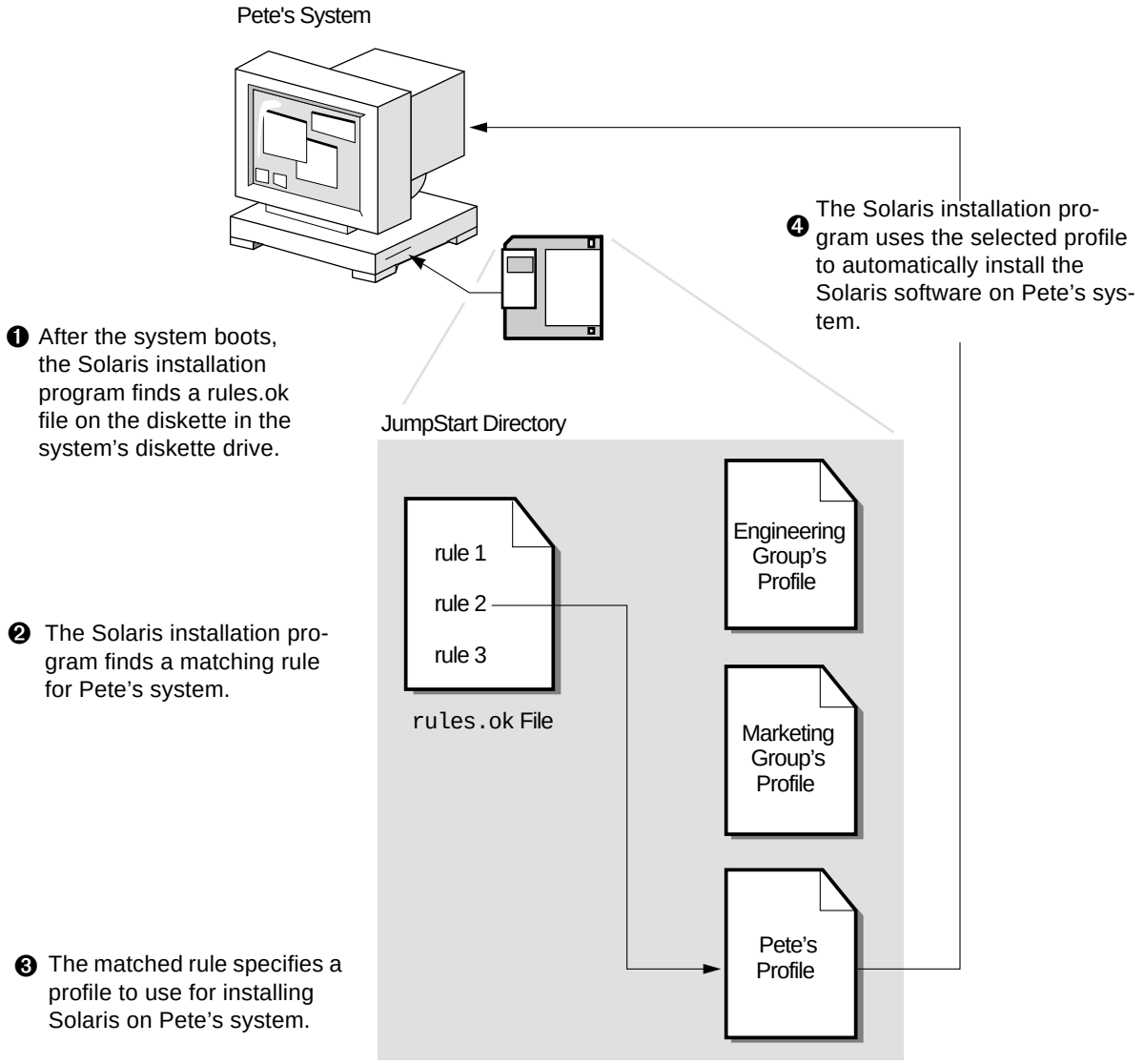


Figure 4-2 How a Custom JumpStart Installation Works: Non-Networked Example

Figure 4-3 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.

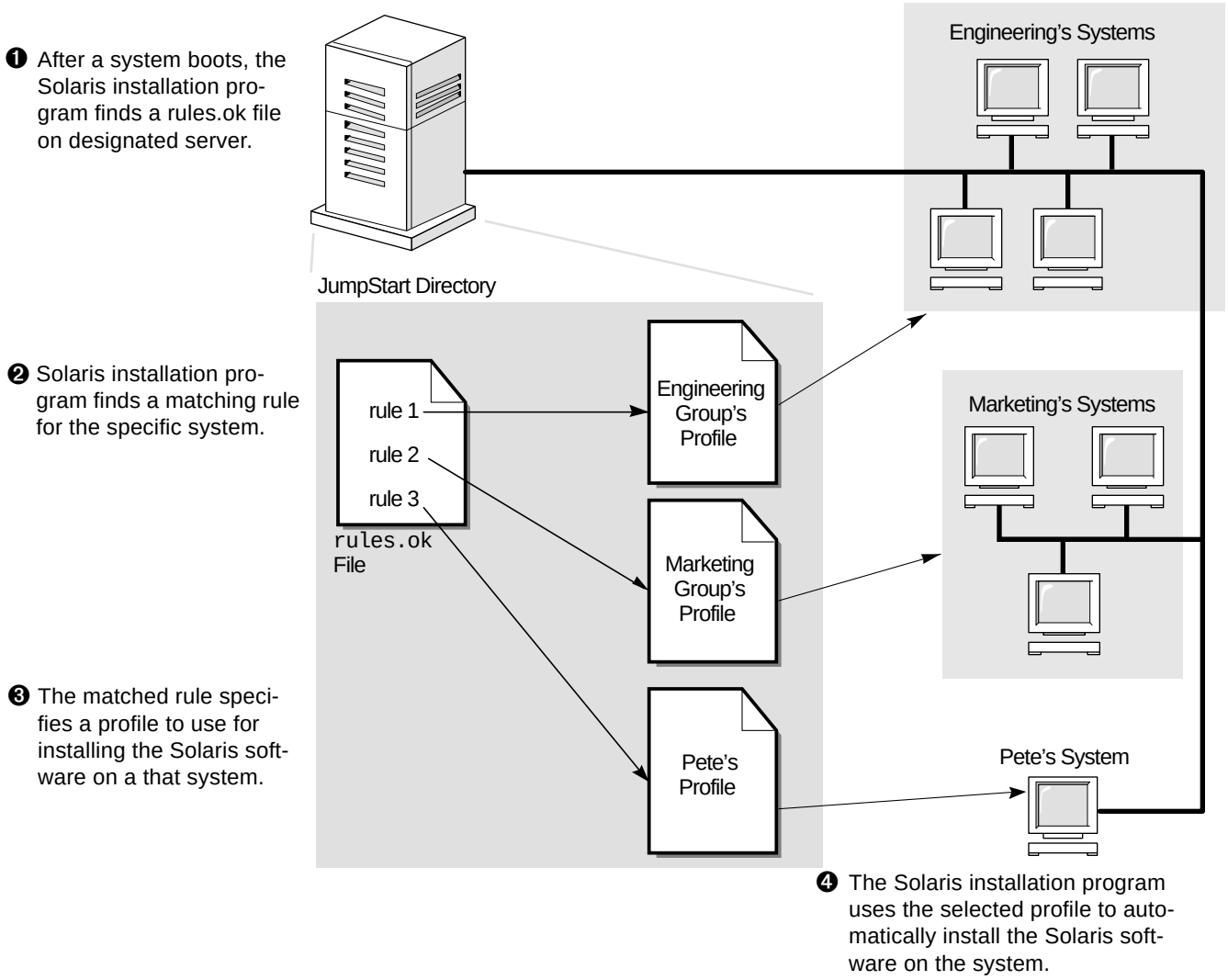


Figure 4-3 How a Custom JumpStart Installation Works: Networked Example

Creating a JumpStart Directory on a Diskette

You should use a diskette for a custom JumpStart installation if the system:

- Has a diskette drive
- Has a local CD-ROM drive
- Is *not* connected to a network

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

The diskette requirements for the JumpStart directory are different for x86 and SPARC systems. This section has a procedure to only create a JumpStart directory on a diskette for x86 systems. A SPARC system is required to create a JumpStart directory on a diskette for SPARC systems.

▼ How to Create a JumpStart Directory on a Diskette for x86 Systems

Overview – The procedure to create a JumpStart directory on a diskette for x86 systems involves:

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

Follow this procedure to create a JumpStart directory on a diskette for x86 systems.

1. **Log in as root on an x86 or SPARC system that has a diskette drive and a CD-ROM drive.**
2. **Insert the Solaris boot diskette into the diskette drive.**

3. **Choose the appropriate steps, depending on whether or not the system is running Volume Management:**

If the System Is ...	Then ...
Running Volume Management	Go to Step 4. **Note:** Volume Management is running if the `/vol` directory on the system contains files. Systems running Solaris 2.0 or 2.1 do not have Volume Management.
Not running Volume Management	Go to Step 12.

4. **Make sure Volume Management knows about the diskette:**

```
# volcheck
```

5. **Copy the Solaris boot diskette image to the system's hard disk:**

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

In this command,

boot_image — Is the file name where the Solaris boot diskette image is copied. You can specify an absolute path name.

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

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

6. **Eject the Solaris boot diskette:**

Note – The following command is not required for x86 systems; you must manually eject the diskette on an x86 system.

```
# eject floppy
```

7. **Insert a blank diskette into the diskette drive.**

8. **Make sure Volume Management knows about the diskette:**

```
# volcheck
```

9. **Format the diskette:**

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

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

10. **Copy the Solaris boot 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 5.

11. **Go to Step 17.**
Step 12 through Step 16 are used if the system is not running Volume Management.

12. **Copy the Solaris boot diskette image to the system's hard disk:**

```
# dd if=/dev/rdiskette of=boot_image
```

In this command,

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

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

```
dd if=/dev/rdiskette of=boot_save
```

13. **Eject the Solaris boot diskette:**

Note – The following command is not required for x86 systems; you must manually eject the diskette on an x86 system.

```
# eject fd
```

14. **Insert a blank diskette into the diskette drive.**

15. **Format the diskette:**

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

```
# fdformat -d
```

16. **Copy the Solaris boot diskette image from the system's hard disk to the formatted diskette:**

```
# dd if=boot_image of=/dev/rdiskette
```

The *boot_image* variable should be the same as in Step 12.

17. **Mount the diskette:**

If the System Is ...	Then ...
Running Volume Management	**1)** Eject the copied Solaris boot diskette. **2)** Insert the copied Solaris boot diskette back into the diskette drive. **3)** Make sure Volume Management knows about the diskette: `volcheck`
Not running Volume Management	`mount -F pcfs /dev/diskette` *jump_dir_path* **Note:** *jump_dir_path* is the absolute directory path where the diskette is mounted.

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

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)** Go to Step 19.
Solaris CD image on local disk	**1)** Change the directory to the Solaris CD image on the local disk. For example: `cd /export/install` **2)** Go to Step 20.

19. **Mount the Solaris CD (if needed) and change the directory to the mounted CD:**

If the System Is ...	Then ...

Running Volume Management	`cd /cdrom/cdrom0/s0`
	or
	`cd /cdrom/cdrom0/s2`

Not running Volume Management	**1)**`mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom` or `mount -F hsfs -o ro /dev/dsk/c0t6d0p0 /cdrom` **2)**`cd /cdrom`

20. **Copy the JumpStart installation files from the `auto_install_sample` directory on the Solaris CD into the JumpStart directory (root directory) of the diskette:**

If the System Is ...	Then ...
Running Volume Management	`cp -r auto_install_sample/* /floppy/floppy0/.`
Not running Volume Management	`cp -r auto_install_sample/*` *jumpstart_dir_path* **Note:** *jump_dir_path* is the absolute directory path where the diskette is mounted.

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 – The custom JumpStart installation files must be in the root directory of the diskette.

Task Complete

You have completed creating a JumpStart directory on the diskette. To continue, see “How to Create a Profile” on page 64.

Creating a JumpStart Directory on a Server

If you want to perform custom JumpStart installations by using a server on the network, you must create a JumpStart directory on the server. When you use a server for custom JumpStart installations, the JumpStart directory is a directory on the server that contains all the essential custom JumpStart files (for example, the `rules` file, `rules.ok` file, and profiles). The JumpStart directory should be owned by root and have permissions equal to 755.

▼ How to Create a JumpStart Directory on a Server

Overview – The procedure to create a JumpStart directory on a server involves:

- Creating a directory on the server
- Editing the `/etc/dfs/dfstab` file
- Copying sample custom JumpStart installation files into the directory on the server

Follow this procedure to create a JumpStart directory on a server.

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
```

In this command,

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 be correct for the example shown in Step 2:

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

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

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

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

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)** Go to Step 7.
Solaris CD image on the local disk	**1)** Change the directory to the Solaris image on the local disk. For example: `cd /export/install` **2)** Go to Step 8.

7. **Mount the Solaris CD (if needed) and change the directory to the mounted CD:**

If the System Is ...	Then ...

Running Volume Management	`cd /cdrom/cdrom0/s0`
	or
	`cd /cdrom/cdrom0/s2`
	
	Note: Volume Management is running if the `/vol` directory on the system contains files. Systems running Solaris 2.0 or 2.1 do not have Volume Management.

Not running Volume Management	**1)**`mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom` or `mount -F hsfs -o ro /dev/dsk/c0t6d0p0 /cdrom` **2)**`cd /cdrom`

8. **Copy the contents of the** `auto_install_sample` **directory from the Solaris CD-ROM into the JumpStart directory:**

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

For example, the following command would copy the `auto_install_sample` directory into the JumpStart directory created in Step 2:

```
cp -r auto_install_sample/* /jumpstart
```

Task Complete

You have completed creating a JumpStart directory on the server. To continue, see "How to Create a Profile" on page 64.

Enabling All Systems to Access the JumpStart Directory

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

- Using the `-c` option of the `add_install_client` command every time you add a system for network installation.

  or

- Enabling all systems to access the JumpStart directory.

To save you time when adding systems for network installations, use the following procedure to enable all systems to access the JumpStart directory from a server.

Note – The following procedure is not necessary if you are using a diskette for the JumpStart directory.

▼ How to Enable All Systems to Access the JumpStart Directory

Overview – The procedure to enable all systems to access the JumpStart directory from a server involves:

- Editing the `/etc/bootparams` file
- Updating the name service (NIS or NIS+) with the information you've added to the `/etc/bootparams` file

Follow this procedure to enable all systems to access the JumpStart directory from a server.

1. **On the NIS or NIS+ master server, log in as root.**
2. **Edit the `/etc/bootparams` file or create the `/etc/bootparams` file if it does not exist. Add the following entry:**

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

In this entry,

*	Is a wildcard character specifying all systems.
server	Is the host name of the 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 server named `sherlock`:

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

Caution – Using this procedure may produce the following error message when booting an install client:
`WARNING: getfile: RPC failed: error 5: (RPC Timed out).`
See page 170 for more details on this error message.

3. **Update the NIS or NIS+ tables (if necessary) with the information you added to the `/etc/bootparams` files.**

If Your Site Uses ...	Then ...
NIS	`/var/yp/make`
NIS+	`/usr/lib/nis/nispopulate -F -p /etc bootparams`
No name service	Go to Task Complete on page 63.

Task Complete

All systems can now access the JumpStart directory. You no longer need to use the `-c` option of the `add_install_client` command when adding systems for network installations.

Creating a Profile

What Is a Profile

A profile is a text file used as a template by the custom JumpStart installation software. It defines how to install the Solaris software on a system (for example, initial or upgrade installation option, system type, disk partitioning, software group), and it is named in the `rules` file.

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 page 53 or page 59, example profiles should already be in the JumpStart directory.

Requirements for Profiles

The following are requirements when creating a profile:

- The `install_type` profile keyword is required.
- Only one profile keyword can be on a line.

▼ How to Create a Profile

Overview – The procedure to create a profile involves:

- Editing a file
- Selecting profile keywords and profile values to define how to install the Solaris software on a system

Follow this procedure to create as many profiles as you need for your site.

1. **Open a file (the profile) and give it a descriptive name.**
   You can create a new file 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.**
   Be aware of these things as you edit the profile:

   - Profile Examples on page 65 provides some examples of profiles.
   - Table 4-2 on page 68 provides the list of valid profile keywords and values.
   - You can have as many lines in the profile as necessary to define how to install the Solaris software on a system.
   - You can add 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 are also allowed in a profile.
   - The profile keywords and their values *are* case sensitive.
   - Profiles should be owned by root and have permissions equal to 644.

Note – See “Using pfinstall to Test Profiles” on page 100 for detailed information about testing profiles.

Task Complete

This completes the procedure to create a profile. To continue setting up for a custom JumpStart installation, see How to Create the rules File on page 79.

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 Table 4-2 on page 68 for the list of profile keywords and profile values.

```
    # profile keywords      profile values
    # ----------------      --------------------
❶   install_type           initial_install

❷   system_type            standalone

❸   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

❹   cluster                SUNWCprog
❺   package                SUNWman delete
    package                SUNWolman delete
    package                SUNWxwman delete
    package                SUNWoldem add
    package                SUNWxwdem add
    package                SUNWoldim add
    package                SUNWxwdim add
```

❶ This profile keyword is required in every profile.

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

❸ 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.

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

❺ 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.

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

❶  partitioning          default
   filesys               c0t0d0s0 auto /
   filesys               c0t3d0s1 32 swap
❷  cluster               SUNWCall
```

❶ The file system slices are determined by the software to be installed (`default` value). However, the size of root is based on the selected software (`auto` value) and it is installed on c0t0d0s0, and the size of `swap` is set to 32 Mbytes and it is installed on c0t3d0s1.

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

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

❶  fdisk                 c0t0d0 0x04 delete
❷  fdisk                 c0t0d0 solaris maxfree
❸  cluster               SUNWCall
❹  cluster               SUNWCacc delete
```

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

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

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

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

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

❷  package               SUNWbcp delete
❸  package               SUNWolman add
   package               SUNWxwman add
   cluster               SUNWCumux add

❹  locale                de
```

❶ This profile upgrades a system (SPARC only).

❷ The binary compatibility package (SUNWbcp) is selected to be deleted from the system or prevented from being installed.

❸ 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.)

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

4

Profile Keyword and Profile Value Descriptions

Table 4-2 shows the profile keywords and profile values that you can use in a profile.

Table 4-2 Profile Keyword and Profile Value Descriptions *(1 of 10)*

Profile Keyword	Profile Values and Description
`client_arch`	*karch_value*
	`client_arch` defines that the server will support a different platform group than it uses. If you do not specify `client_arch`, any diskless client must have the same platform group as the server. You must specify `client_arch` once for each platform group. Valid values for *karch_value* are sun4d, sun4c, sun4m, or i86pc. (See Appendix C, "Platform Names and Groups" for a detailed list of the platform names of various systems.) **Restriction:** `client_arch` can be used only when `system_type` is specified as `server`.
`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. **Restriction:** `client_root` can be used only when `system_type` is specified as `server`.
`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`, 24 Mbytes of swap space is allocated. Example: `client_swap 32` The example defines that each diskless client will have a swap space of 32 Mbytes. **Restriction:** `client_swap` can be used only when `system_type` is specified as `server`.

† **Profile keywords that can be used for upgrading**

Table 4-2 Profile Keyword and Profile Value Descriptions *(2 of 10)*

Profile Keyword	Profile Values and Description
`cluster` *(use for software groups)*	*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 — `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`† *(use for clusters)*	*cluster_name* `[add | delete]`
	`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 select Software from the Browse menu. 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.

† **Profile keywords that can be used for upgrading**

Table 4-2 Profile Keyword and Profile Value Descriptions *(3 of 10)*

Profile Keyword	Profile Values and Description
`dontuse`	*disk_name* `dontuse` designates a disk that the Solaris installation program should *not* use when `partitioning default` is specified. You can specify `dontuse` once for each disk, and *disk_name* must be specified in the form `c`*x*`t`*y*`d`*z* or `c`*y*`d`*z*, for example, `c0t0d0`. By default, the Solaris installation program uses all the operational disks on the system. **Restriction:** You cannot specify the `dontuse` keyword and the `usedisk` keyword in the same profile.
`fdisk`	*disk_name type size* `fdisk` defines how the fdisk partitions are set up on an x86 system (only required for x86 systems), and you can specify `fdisk` more than once. This is what happens by default with fdisk partitions on x86 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 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 appear 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 disk where the root file system is placed during an installation, which is determined by the Solaris installation program (described on page 87). • `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).

† **Profile keywords that can be used for upgrading**

Table 4-2 Profile Keyword and Profile Value Descriptions *(4 of 10)*

Profile Keyword	Profile Values and Description
`fdisk` continued	*disk_name type size* *type* - Choose what type of fdisk partition will be created or deleted on the specified disk: • *DDD* - A decimal fdisk partition. *DDD* is a decimal number (valid values are 1 through 255). **Restriction:** 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). **Restriction:** This value can be specified only if *size* is `delete`. The following table shows the decimal and hexadecimal numbers for some of the fdisk types: fdisk type*DDDHH* DOSOS12101 PCIXOS 202 DOSOS16404 EXTDOS505 DOSHUGE606 DOSDATA8656 OTHEROS9862 UNIXOS9963 *size* - Choose one of the following: • *DDD* - An fdisk partition of size *DDD* (in Mbytes) is created on the specified disk. *DDD* must be a decimal 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). **Restriction:** 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 an error will occur. **Restriction:** 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.

† **Profile keywords that can be used for upgrading**

Table 4-2 Profile Keyword and Profile Value Descriptions *(5 of 10)*

Profile Keyword	Profile Values and Description
`filesys` *(use for mounting remote file systems)*	*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. **For Dataless Clients:** Profiles for dataless clients (when `system_type dataless` is specified) must include a remote `filesys` entry for both the `/usr` file system. The following line is an example of the `filesys` entries that must be used in a profile to install the Solaris software on a dataless client: `filesys sherlock:/export/exec/Solaris_2.4_sparc.all/usr - /usr` *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 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`

† **Profile keywords that can be used for upgrading**

Table 4-2 Profile Keyword and Profile Value Descriptions *(6 of 10)*

Profile Keyword	Profile Values and Description
`filesys` *(use for creating local file systems)*	*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. **Restriction:** `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 logical name of the disk where the installation program places the root file system. 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. **Restriction:** If `free` is used as the value to `filesys`, it must by 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.

† **Profile keywords that can be used for upgrading**

Table 4-2 Profile Keyword and Profile Value Descriptions *(7 of 10)*

Profile Keyword	Profile Values and Description
`filesys` *(use for creating local file systems)* continued	*slice size* [*file_system*] [*optional_parameters*]
	file_system - You can use this optional value when *slice* is specified as `any` or `cwtxdysz`. 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. **Restriction**: `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 could 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. *optional_parameters* - Choose one of the following: • `preserve` - The file system on the specified *slice* is preserved. **Restriction**: `preserve` can be specified only when *size* is `existing` and *slice* is `cwtxdysz`. • *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`†	`initial_install | upgrade`
	`install_type` defines whether to perform the initial installation option or upgrade option on the system. **Restriction**: `install_type` must be the first profile keyword in every profile.

† **Profile keywords that can be used for upgrading**

Table 4-2 Profile Keyword and Profile Value Descriptions *(8 of 10)*

Profile Keyword	Profile Values and Description
`locale`†	*locale_name*
	`locale` designates that the localization packages associated with the selected software should be installed (or added for upgrade) for the specified *locale_name*. The *locale_name* values are the same as the values used for the `$LANG` environment variable. Solaris 2.4 supports the following localizations: Language — *locale_name* Chinese — `zh` French — `fr` German — `de` Italian — `it` Japanese — `ja` Korean — `ko` Latin American — `es` Swedish — `sv` Taiwanese — `zh_TW` The English localization packages are installed by default. You can specify `locale` once for each localization you need to support.
`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. **Restriction:** `num_clients` can be used only when `system_type` is specified as `server`.

† **Profile keywords that can be used for upgrading**

Table 4-2 Profile Keyword and Profile Value Descriptions *(9 of 10)*

Profile Keyword	Profile Values and Description
`package`†	*package_name* `[add	delete]`
	`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 (select Software from the Browse menu) on an installed system to view detailed information about packages and their names.  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.

† **Profile keywords that can be used for upgrading**

Table 4-2 Profile Keyword and Profile Value Descriptions *(10 of 10)*

Profile Keyword	Profile Values and Description
`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. **Restriction:** 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. **Restriction:** 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.
`system_type`	`standalone	dataless	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`	*disk_name*
	`usedisk` designates a disk that the Solaris installation program will use when `partitioning default` is specified. You can specify `usedisk` once for each disk, and *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. **Restriction:** You cannot specify the `usedisk` keyword and the `dontuse` keyword in the same profile.

† **Profile keywords that can be used for upgrading**

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 4-3 shows how the maximum size of swap is determined during a custom JumpStart installation.

Table 4-3 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
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 4-3.

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

Creating the `rules` *File*

What Is the `rules` *File*

The `rules` file is a text file used to create the `rules.ok` file. The `rules` file is a look-up table consisting of one or more rules that define matches between system attributes and profiles. For example, the rule

```
karch sun4c - basic_prof -
```

matches a system with a sun4c platform name to the `basic_prof` profile, which the Solaris installation program would use to install the system.

Note – If you set up the JumpStart directory by using the procedures on page 53 or page 59, 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 rules in the `rules.ok` file in order: first rule through the last rule. A rule match occurs when the system being installed matches any of the rule values in the rule (as defined in Table 4-5 on page 84). As soon as a system matches a rule, the Solaris installation program stops reading the `rules.ok` file and begins to install the system as defined by the matched rule's profile.

▼ How to Create the `rules` File

Overview – The procedure to create a `rules` file involves:

- Editing a file
- Selecting rule keywords and rule values for each group of systems you want to install using custom JumpStart. Any systems that match the rule keyword and rule value will be installed as specified by the corresponding profile.

Follow this procedure to create a `rules` file.

1. **Open a file (the `rules` file) and name it `rules`.**
   You can create a new 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.**
   Be aware of these things as you add rules to the `rules` file:

   - Rule Examples on page 82 provides some examples of rules.
   - Table 4-5 on page 84 provides the list of valid rule keywords and values.
   - The `rules` file must have at least one rule

- A rule must have at least a rule keyword, a rule value, and a corresponding profile.

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

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

Table 4-4 describes the fields of a rule.

Table 4-4 Field Descriptions of a Rule

Field	Description
!	A symbol used before a rule keyword to indicate negation.
[ ]	A symbol used to indicate an optional expression or field.
...	A symbol used to indicate the preceding expression may be repeated.
rule_keyword	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 4-5 on page 84 for the list of `rule` keywords.
rule_value	A value that provides the specific system attribute for the corresponding `rule` keyword. See Table 4-5 on page 84 for the list of `rule` values.
&&	A symbol that must be used to join (logically AND) rule keyword and rule value pairs together in the same rule. During a custom JumpStart installation, a system must match every pair in the rule before the rule matches.

Table 4-4 Field Descriptions of a Rule *(Continued)*

Field	Description
begin	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 94 for detailed information on how to create begin scripts.
profile	A name of a text file used as a template that defines how to install Solaris on a system. 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 110 and "Creating Derived Profiles With Begin Scripts" on page 94.
finish	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 96 for detailed information on how to create finish scripts.

Task Complete

This completes the procedure to create a `rules` file. To validate the `rules` file, see How to Use check to Validate the rules File on page 90.

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 valuesbegin script  profile    finish script
    # ---------------------------------------- -------    -------------
❶   hostname eng-1                    -          basic_prof     -
❷   network 192.43.34.0 && !model \
    'SUNW,Sun 4_50'                   -          net_prof       -
❸   model SUNW,SPARCstation-LX        -          lx_prof    complete
❹   network 193.144.2.0 && karch i86pcsetup      x86_prof   done
❺   memsize 16-32 && arch sparc       -          prog_prof      -
❻   any  -                            -          generic_prof   -
```

❶ 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.

❷ 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.

❸ 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 page 83.

❹ This rule matches if the system is on subnet 193.144.2.0 and the system is an x86. 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.

❺ This rule matches if the system has 16-32 Mbytes of memory and its processor type is SPARC. The `prog_prof` profile is used to install the Solaris software on systems that match this rule.

❻ 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.

Important Information About the `rules` *File*

The following information is important to know about the `rules` file:

- **Name** - The `rules` file *must* have the file name, `rules`.
- `rules.ok` **file** - The `rules.ok` file is a generated version of the `rules` file, and it is required by the custom JumpStart installation software to match a system to a profile. You must run the `check` script to create the `rules.ok` file, and the `rules.ok` file should be owned by root and have permissions equal to 644.
- **Comments** - You can add 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 are also allowed in the `rules` file.

Note – When creating the `rules.ok` file, the `check` script removes all the comment lines, comments at the end of a rule, and blank lines.

- **Rule wrap** - When a rule spans multiple lines, you can let a rule to wrap to a new line, or you can continue a rule on a new line by using a backslash (\) before the carriage return.
- **Rule fields** - The *rule_value*, *begin*, and *finish* fields must have a valid entry or a minus sign (-) to specify that there is no entry.

4

Rule Keyword and Rule Value Descriptions

Table 4-5 describes the rule keywords and rule values that you can use in the `rules` file.

Table 4-5 Rule Keyword and *Rule Value* Descriptions *(1 of 4)*

Rule Keyword	Rule Values	Description
`any`	minus sign (-)	Match always succeeds.
`arch`	*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`. `rootdisk` should be used only when trying to match systems that contain the factory-installed JumpStart software. `rootdisk` is described on page 87. *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 "207 Mbyte" disk, but it may have only 207 million bytes of disk space. The Solaris installation program will actually view the "207 Mbyte" disk as a 197 Mbyte disk because 207,000,000 / 1,048,576 = 197. So, a "207 Mbyte" disk would not match a *size_range* equal to 200-210.
`hostaddress`	*IP_address*	Matches a system's IP address.

Table 4-5 Rule Keyword and *Rule Value* Descriptions *(2 of 4)*

Rule Keyword	Rule Values	Description
`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.
`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, any disk attached to the system attempts to match. `rootdisk` should be used only when trying to match systems that contain the factory-installed JumpStart software. `rootdisk` is described on page 87. *version* - A version name, such as `Solaris_2.3`, 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.
`karch`	*platform_group* Valid values are sun4d, sun4c, sun4m, or i86pc. (See Appendix C, "Platform Names and Groups" for a detailed list of platform groups and names.)	Matches a system's platform name. 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 `prtconf` command (line 2) reports the system's physical memory size.

Table 4-5 Rule Keyword and *Rule Value* Descriptions *(3 of 4)*

Rule Keyword	Rule Values		Description
`model`	*model_name*		Matches a system's model number, which is system-dependent and varies by the manufacturer. The list shown may not be complete. If you have a system already installed, the `prtconf` command (line 5) reports the system's model number. **Note:** If the *model_name* contains spaces, the *model_name* must be inside a pair of single quotes ('). For example: `'SUNW,Sun 4_50'`
	System	*model_name*	
	Sun-4/110	Sun 4_100 Series	
	Sun-4/2*xx*	Sun 4_200 Series	
	SPARCstation 1 (4/60)	Sun 4_60	
	SPARCstation 1+ (4/65)	Sun 4_65	
	SPARCstation SLC™ (4/20)	Sun 4_20	
	SPARCstation IPC (4/40)	SUNW,Sun 4_40	
	SPARCstation ELC™ (4/25)	SUNW,SUN 4_25	
	SPARCstation IPX (4/50)	SUNW,Sun 4_50	
	SPARCstation 2 (4/75)	SUNW,SUN 4_75	
	Sun-4/3*xx*	Sun SPARCsystem 300	
	Sun-4/4*xx*	Sun SPARCsystem 400	
	SPARCserver™ 6*xx*	SUNW,SPARCsystem-600	
	SPARCstation 10	SUNW,SPARCstation-10	
	SPARCclassic™ (4/15)	SUNW,SPARCclassic	
	SPARCstation LX (4/30)	SUNW,SPARCstation-LX	
	SPARCcenter™ 1000	SUNW,SPARCserver-1000	
	SPARCcenter 2000	SUNW,SPARCcenter-2000	
	SPARCstation 10 SX	SUNW,SPARCstation-10,SX	
	SPARCstation 20	SUNW,SPARCstation-20	
	SPARCstation 5	SUNW,SPARCstation-5	
	SPARCstation Voyager	SUNW,S240	
	Sun Ultra™ 1 Model 140	SUNW,Ultra1-140	
	x86	i86pc	

Table 4-5 Rule Keyword and *Rule Value* Descriptions *(4 of 4)*

Rule Keyword	Rule Values	Description
`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 would match a system with a 193.144.2.8 IP address (if the subnet mask were 255.255.255.0).
`osname`	*Solaris_version*	Matches a version of Solaris already installed on a system. *Solaris_version* is the version of Solaris environment installed on the system: for example, Solaris_2.5.
`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 1048576 bytes. A disk may be advertised as a "207 Mbyte" disk, but it may have only 207 million bytes of disk space. The Solaris installation program will actually view the "207 Mbyte" disk as a 197 Mbyte disk because 207000000 / 1048576 = 197. So, a "207 Mbyte" disk would not match a *size_range* equal to 200-210.

How the Installation Program Sets the Value of `rootdisk`

`rootdisk` is the logical name of the disk where the root file system is placed during an installation. During a custom JumpStart installation, the Solaris installation program sets the value of `rootdisk` (that is, the actual disk it represents) depending on various situations; this is described in Table 4-6.

Table 4-6 How the Solaris Installation Program Sets the Value of `rootdisk`

Situation	What Happens
A system contains the factory-installed JumpStart software. (this applies to some SPARC systems only).	`rootdisk` is set to the disk that contains the factory-installed JumpStart software before the system tries to match any rules.
`rootdisk` has *not* been set and a system tries to match the following rule: `disksize rootdisk` *size_range* or `installed rootdisk` *version*	`rootdisk` is set to c0t3d0 *or* the first available disk attached to the system. After `rootdisk` is set, the system tries to match the rule.
If `rootdisk` has been set and the system tries to match the following rule. `disksize rootdisk` *size_range* or `installed rootdisk` *version*	The system tries to match the rule.
A system tries to match the following rule: `installed` *disk version*	If *disk* is found on the system with a root file system that matches the specified *version*, the rule matches and `rootdisk` is set to *disk*.
A system tries to match the following rule: `installed any` *version*	If any disk is found on the system with a root file system that matches the specified *version*, the rule matches and `rootdisk` is set to the found disk. (If there is more than one disk on the system that can match, the system will match the first disk that is found.)

Table 4-6 How the Solaris Installation Program Sets the Value of `rootdisk` *(Continued)*

Situation	What Happens
`rootdisk` has not been set after a system matches a rule and the system is going to be upgraded (which is defined in the profile).	`rootdisk` is set to the first disk found with a root file system that matches an upgradable version of Solaris software. If no disk is found, the system proceeds with an interactive installation.
`rootdisk` has not been set after a system matches a rule.	`rootdisk` is set to c0t3d0 *or* the first available disk attached to the system.

For the Solaris installation program to use the value of `rootdisk`, the following conditions must be true in the profile specified for the system:

- Default partitioning is used.
- No slice has been explicitly set for the root file system.

Using `check` *to Validate 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. The following table shows what the check script does.

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`

▼ How to Use `check` to Validate the `rules` File

Overview – The procedure to use `check` to validate the `rules` file involves:

- Making sure the check script resides in the JumpStart directory
- Running the check script

Follow this procedure to use `check` 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 `auto_install_sample` directory on the Solaris CD.

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

```
$ cd jumpstart_dir_path
```

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

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

In this command,

`-p` *path*	Is the path to the Solaris 2.4 CD. You can use a Solaris CD image on a local disk or a mounted Solaris CD.This option ensures that you are using the most recent version of the `check` script. You should use this option if you are using `check` on a system that 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 auto-install configuration is ok`.

Task Complete

The rules files is now validated. To read about the optional features available for custom JumpStart installations, see Chapter 5, “Using Optional Custom JumpStart Features.” To perform a custom JumpStart installation on a system, see Chapter 8, “Booting and Installing Solaris: Custom JumpStart.”

4

Using Optional Custom JumpStart Features 5

Overview

This chapter describes the optional features available for custom JumpStart installations, and it is a supplement to Chapter 4, “Preparing Custom JumpStart Installations.” You can use the following optional features to enhance and test custom JumpStart installations:

- Begin scripts
- Finish scripts
- `pfinstall`
- Site-specific installation program

Note – This chapter is valid for either an x86 or SPARC server that is being used for custom JumpStart installations. An x86 server can provide custom JumpStart files for SPARC systems and a SPARC system can provide custom JumpStart files for x86 systems.

5

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 upgrade

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}
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.

5

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:

- Installing patches
- Restoring backed up files
- Setting up print servers
- Adding entries to the automount map

The following finish scripts are provided as examples:

- Adding files
- Customizing the root environment
- Setting the system's root password

Adding Files With Finish Scripts

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:

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 finish script would copy the `site_prog` from the JumpStart directory into a system's `/usr/bin` directory during a custom JumpStart installation:

   ```
   #!/bin/sh
   cp ${SI_CONFIG_DIR}/site_prog  /a/usr/bin
   ```

Customizing the Root Environment

Through a finish script, you can customize files already installed on the system. For example, the following finish script customizes the root environment by appending information to the `.cshrc` file in the root directory.

```
#!/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 Finish Scripts

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. This allows the initial reboot of the system to be completed without prompting for a root password.

The `set_root_pw` file is shown below.

```
#!/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
❶ PASSWD=dKO5IBkSF42lw
mv /a/etc/shadow /a/etc/shadow.orig
nawk -F: '{
    if ( $1 == "root" )
❷       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

❸ # 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}
```

There are three main things you must do to set the root password in a finish script.

❶ Set the variable `PASSWD` to an encrypted root password obtained from an existing entry in a system's `/etc/shadow` file.

❷ Change the root entry in the `/etc/shadow` file for the newly installed system using `$PASSWD` as the password field.

❸ Change 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.

Using `pfinstall` *to Test Profiles*

Why Use `pfinstall`

When `install_type initial_install` is defined in a profile, you can use the `pfinstall` command to test the profile without actually installing the Solaris software on a system. `pfinstall` shows the results of how a system would be installed according to the specified profile, before you actually perform a custom JumpStart installation.

Note – You cannot use `pfinstall` to test a profile using the upgrade option (when the `initial_type upgrade` is defined in a profile).

Ways to Use `pfinstall`

`pfinstall` enables you to test a profile against:

- The system's disk configuration where `pfinstall` is being run.
- A disk configuration file that you can create with the `prtvtoc` command. A *disk configuration file* is 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.

You must run `pfinstall` on an x86 system to test profiles for x86 systems (using the x86 system's disk configuration or an x86 disk configuration file). Conversely, you must run `pfinstall` on a SPARC system to test profiles for SPARC systems (using the SPARC system's disk configuration or a SPARC disk configuration file).

▼ How to Use `pfinstall` to Test a Profile

Overview – The procedure to use `pfinstall` to test a profile involves:

- Changing the directory to the JumpStart directory
- Using the `pfinstall` command to test the profile

Follow this procedure to use `pfinstall` to test a profile.

1. **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
```

2. **Change the directory to the JumpStart directory where the profile resides:**

```
$ cd jumpstart_dir_path
```

For example, the following command would change the directory to the `jumpstart` directory on the root file system.

```
cd /jumpstart
```

3. **Run the `pfinstall -d` or `pfinstall -D` command to test the profile:**

Caution – Without the `-d` or `-D` option, `pfinstall` will install 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
```

In this command,

`-D`	Tells `pfinstall` to use the current system's disk configuration to test the profile against. You must be root to execute `pfinstall` with the `-D` option.

-d *disk_config*	Tells `pfinstall` to use a disk configuration file, *disk_config*, to test the profile against.
`-c` *path*	Is the path to the Solaris CD. This is required if the Solaris CD is not mounted on `/cdrom`. (For example, use this option if you copied the Solaris CD image to disk or mounted the Solaris CD on a directory other than `/cdrom`).
profile	The name of the profile to test.

Note – You should run `pfinstall` on a system running the same version of Solaris software that will be installed by the profile. Otherwise, use `pfinstall` on the Solaris CD that will be installed by the profile, which is located in the `/export/exec/`*arch*`.Solaris_2.4/sbin/install.d` directory.

Run `pfinstall` from the directory where the *profile* and *disk_config* files reside (which should be the JumpStart directory). If the *profile* or *disk_config* file is not in the directory where `pfinstall` is run, you must specify the path.

4. **Check to see if the results of `pfinstall` are as you expected. If not, change the profile and go to Step 3.**

Task Complete

You have completed testing the profile. To perform a custom JumpStart installation on a system, see Chapter 8, “Booting and Installing Solaris: Custom JumpStart.”

`pfinstall` *Examples*

Below are some examples of using `pfinstall` to test the `basic_prof` profile against the `104_test` disk configuration file:

```
/usr/sbin/install.d/pfinstall -D basic_prof

/usr/sbin/install.d/pfinstall -d 104_test basic_prof

/usr/sbin/install.d/pfinstall -D -c /export/install basic_prof
```

Creating a Disk Configuration File for an x86 System

The following procedures enable you to use the `-d` option of the `pfinstall` command to test custom JumpStart installations for x86 systems.

▼ How to Create a Disk Configuration File for an x86 System

A disk configuration file represents a disk structure (for example, bytes/sectors, flags, slices), and it enables you to use `pfinstall` from a single system to test profiles against different sized disks. Disk configuration files for x86 systems must also contain information about a disk's fdisk partitions.

Overview – Creating a disk configuration file for an x86 system involves:

- Locating an x86 system with a disk that you want to test a profile against
- Saving the output of the `prtvtoc` command to a file
- Saving the output of the `fdisk` command to a file
- Concatenating the two files to create a disk configuration file

Follow this procedure to create a disk configuration file for an x86 system

1. **Locate an x86 system with a disk that you want to test a profile against.**
2. **Determine the device name for the system's disk.**
3. **Redirect the output of the following `prtvtoc` command to a file:**

```
# prtvtoc /dev/rdsk/device_name > file1
```

   where `/dev/rdsk/`*device_name* is the device name of the system's disk, and *file1* is the file that contains the output of the `prtvtoc` command. *device_name* must be in the form `c`*w*`t`*x*`d`*y*`p0` or `c`*x*`d`*y*`p0`. Partition 0 must be specified in *device_name*.

4. **Save the output of the following `fdisk` command to a file:**

```
# fdisk -R -d -n /dev/rdsk/device_name 2>file2
```

Note – This version of the `fdisk` command may not be supported in the next release.

where `/dev/rdsk/`*device_name* is the device name of the system's disk. *file2* is the file that contains the output of the `fdisk` command. *device_name* must be in the form `c`*w*`t`*x*`d`*y*`p0` or `c`*x*`d`*y*`p0`. Partition 0 must be specified in *device_name*.

5. **Concatenate the two files to create a disk configuration file:**

```
# cat file1 file2 > disk_config
```

Note – The output of the `prtvtoc` command must be first in a disk configuration file for an x86 system.

6. **Copy the disk configuration file to the JumpStart directory:**

```
# cp disk_config jumpstart_dir_path
```

Task Complete

You have created a disk configuration file for an x86 system. The following page provides an example of creating a disk configuration file. This example creates a disk configuration file, `500_test`, on an x86 system with a 500-Mbyte disk.

First, you would save the output of the `prtvtoc` command to a file:

```
# prtvtoc /dev/rdsk/c0t0d0p0 > output1
```

The `output1` file is shown as follows:

```
* /dev/rdsk/c0t0d0p0 partition map
*
* Dimensions:
*     512 bytes/sector
*      79 sectors/track
*       7 tracks/cylinder
*     553 sectors/cylinder
*    1481 cylinders
*    1479 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                           First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00         553    194103    194655
       1      3    01      194656     65807    260462
       2      6    00           0    819546    819545
       3      6    00      260463     50876    311338
       4      6    00      311339     72996    384334
       6      4    00      384335    434105    818439
       8      1    01           0       553       552
```

Second, you would save the output of the `fdisk` command to a different file:

```
# fdisk -R -d -n /dev/rdsk/c0t0d0p0 2>output2
```

The `output2` file is shown as follows:

```
fdisk physical geometry:
cylinders[1855] heads[7] sectors[79] sector size[512] blocks[1025815] mbytes[500]
fdisk virtual (HBA) geometry:
cylinders[500] heads[64] sectors[32] sector size[512] blocks[1024000] mbytes[500]
fdisk table on entry:
SYSID ACT BHEAD BSECT BEGCYL   EHEAD ESECT ENDCYL   RELSECT   NUMSECT
6     0   1     1     0        63    32    99       32        204768
130   128 0     1     100      63    96    243      204800    819200
100   0   0     0     0        0     0     0        100       100
100   0   0     0     0        0     0     0        100       100
```

Finally, you would concatenate the two files (`output1` and `output2`) together to create the disk configuration file named `500_test`.

```
# cat output1 output2 > 500_test
```

The `500_test` file is shown as follows:

```
* /dev/rdsk/c0t0d0p0 partition map
*
* Dimensions:
*     512 bytes/sector
*      79 sectors/track
*       7 tracks/cylinder
*     553 sectors/cylinder
*    1481 cylinders
*    1479 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00         553    194103    194655
       1      3    01      194656     65807    260462
       2      6    00           0    819546    819545
       3      6    00      260463     50876    311338
       4      6    00      311339     72996    384334
       6      4    00      384335    434105    818439
       8      1    01           0       553       552
fdisk physical geometry:
cylinders[1855] heads[7] sectors[79] sector size[512] blocks[1025815] mbytes[500]
fdisk virtual (HBA) geometry:
cylinders[500] heads[64] sectors[32] sector size[512] blocks[1024000] mbytes[500]
fdisk table on entry:
SYSID ACT BHEAD BSECT BEGCYL   EHEAD ESECT ENDCYL   RELSECT   NUMSECT
6     0   1     1     0        63    32    99       32        204768
130   128 0     1     100      63    96    243      204800    819200
100   0   0     0     0        0     0     0        100       100
100   0   0     0     0        0     0     0        100       100
```

▼ How to Create a Multiple Disk Configuration File for an x86 System

If you need to test a profile on multiple disks, you can concatenate disk configuration files together to create multiple disk configuration scenarios.

Creating a multiple disk configuration file for an x86 system involves:

- Concatenating two or more disk configuration files to create a multiple disk configuration file
- Changing the target numbers of disks in the multiple disk configuration file

Note – You can't have disks with the same target number on a system.

The following procedure creates a multiple disk configuration file. (The procedure uses the `500_test` file from the previous procedure.)

1. **Concatenate a disk configuration file with itself and save it to a file.**
   The new file becomes the multiple disk configuration file. For example, the following command creates a multiple disk configuration file named `dual_500_test`:

```
$ cat 500_test 500_test > dual_500_test
```

2. **Make sure that each disk device name is specified with a different target.**

The `dual_500_test` file is shown as follows:

```
❶ * /dev/rdsk/c0t0d0p0 partition map
  *
  * Dimensions:
  *     512 bytes/sector
  *      79 sectors/track
  *       7 tracks/cylinder
  *     553 sectors/cylinder
  *    1481 cylinders
  *    1479 accessible cylinders
  *
  * Flags:
  *   1: unmountable
  *  10: read-only
  *
  *                           First     Sector    Last
  * Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
        0      2    00         553    194103    194655
        1      3    01      194656     65807    260462
        2      6    00           0    819546    819545
        3      6    00      260463     50876    311338
        4      6    00      311339     72996    384334
        6      4    00      384335    434105    818439
        8      1    01           0       553       552
  fdisk physical geometry:
  cylinders[1855] heads[7] sectors[79] sector size[512] blocks[1025815] mbytes[500]
  fdisk virtual (HBA) geometry:
  cylinders[500] heads[64] sectors[32] sector size[512] blocks[1024000] mbytes[500]
  fdisk table on entry:
  SYSID ACT BHEAD BSECT BEGCYL   EHEAD ESECT ENDCYL   RELSECT   NUMSECT
  6     0   1     1     0        63    32    99       32        204768
  130   128 0     1     100      63    96    243      204800    819200
  100   0   0     0     0        0     0     0        100       100
  100   0   0     0     0        0     0     0        100       100
```

continued

```
❷  * /dev/rdsk/c0t1d0p0 partition map
   *
   * Dimensions:
   *     512 bytes/sector
   *      79 sectors/track
   *       7 tracks/cylinder
   *     553 sectors/cylinder
   *    1481 cylinders
   *    1479 accessible cylinders
   *
   * Flags:
   *   1: unmountable
   *  10: read-only
   *
   *                           First     Sector    Last
   * Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
         0      2    00         553    194103    194655
         1      3    01      194656     65807    260462
         2      6    00           0    819546    819545
         3      6    00      260463     50876    311338
         4      6    00      311339     72996    384334
         6      4    00      384335    434105    818439
         8      1    01           0       553       552
   fdisk physical geometry:
   cylinders[1855] heads[7] sectors[79] sector size[512] blocks[1025815] mbytes[500]
   fdisk virtual (HBA) geometry:
   cylinders[500] heads[64] sectors[32] sector size[512] blocks[1024000] mbytes[500]
   fdisk table on entry:
   SYSID ACT BHEAD BSECT BEGCYL   EHEAD ESECT ENDCYL   RELSECT   NUMSECT
   6     0   1     1     0        63    32    99       32        204768
   130   128 0     1     100      63    96    243      204800    819200
   100   0   0     0     0        0     0     0        100       100
   100   0   0     0     0        0     0     0        100       100
```

This is what was done to the `dual_500_test` file:

❶ The first disk device name was not changed.

❷ The second disk device name was changed from `/dev/rdsk/c0t3d0p0` to `/dev/rdsk/c0t1d0p0`. This gives each disk a different target.

Task Complete

You have created a multiple disk configuration file for an x86 system.

5

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
```

Preparing a System for Upgrade 6

Overview

This chapter provides some guidelines for performing an upgrade installation and for preserving local modifications before upgrading from a previous version of Solaris software.

Can You Use Upgrade?

Upgrade and *initial* are options in the Solaris installation program that determine how Solaris software is copied to disk:

- **Upgrade** – This option merges the new version of Solaris software with existing files on the system's disk. It saves as many local modifications as possible.
- **Initial** – This option overwrites the system's disk with the new version of Solaris software.

To determine if your system supports the upgrade, see the following table.

Upgrade Option Is Available

If at least one disk attached to the system has a Solaris 2.4 or later root file system.

1) To determine if the system has a root file system, type the following command:

```
df -a
```

Look for a line of output similar to the following:

```
Filesystem      kbytes  used   avail  capacity   Mounted on
/dev/dsk/c0t1d0s048295  15665  27810     36%     /
```

2) To determine if the system is running Solaris 2.4 or later, type the following commands:

```
cd /var/sadm/softinfo
more INST_RELEASE
```

Note – The upgrade option may require deleting software packages if the disk is full; this requires knowledge of packages and dependencies.

Upgrading and Install Patches

You do not need to back out install patches before performing an upgrade.

Check Other Software Documentation

Check the documentation of other software you are running (e.g. Solstice™ DiskSuite™, FrameMaker®) before using the upgrade option. There may be additional instructions you need to follow to preserve existing configurations.

If You're Using Custom JumpStart

Make sure you change the profile keyword from `initial` to `upgrade` in your profiles.

If You Override the Boot File Location

For Solaris 2.5 and later, the kernel resides in `/platform/<arch>/kernel/unix;` not in `/kernel/unix.` If you override the boot file location by explicitly setting it to `kernel/unix`, you can successfully install Solaris software, but you won't be able to boot the system.

Upgrading Dataless Clients

Dataless clients do not get automatically upgraded when the OS server is upgraded. After upgrading the OS server, you must perform an upgrade on dataless clients.

Important Files For Upgrading

The following files must be present on standalone systems, servers, and each diskless client being upgraded:

`/var/sadm/softinfor/INST_RELEASE`

`/var/sadm/install/contents`

`/var/sadm/install_data.clustertoc`

`/var/sadm/install_data/CLUSTER`

Backing Up Your System

Always back up an existing system before using the upgrade option and installing a new version of Solaris software. 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 *System Administration Guide, Volume I.*

6

Preserving Local Modifications

During an upgrade, the Solaris installation program attempts to preserve local modifications to the system whenever possible; however, sometimes local modifications can make an upgrade fail or perform differently than you would expect. Table 6-1 indicates tasks you should perform before upgrading to make your upgrade a success.

Table 6-1 Preserving Local Modifications

What to Do Before Upgrading	How to	Why

Preserve symbolic links	Replace symbolic links that use absolute paths with symbolic links that use relative paths.
	
	For example, if `/usr/openwin` is a symbolic link to
	
	`/export/openwin`
	
	change the symbolic link to
	
	`../export/openwin`
		During an upgrade, the installation program cannot reference symbolic links that use absolute paths, because the Solaris installation program mounts the root (`/`) file system at `/a`. For example, a symbolic link to `/export/openwin` would fail, because during an upgrade, `/export/openwin` is really `/a/export/openwin`.
		
		When the Solaris installation program cannot reference a symbolic link, it will overwrite the symbolic link and install the software (the installation program doesn't think the software exists). As a result, duplicate software will be installed on the system and the upgrade may fail because of insufficient disk space.

Preserve symbolic links to automounted file systems	Remove packages (by using Software Manager or the `pkgrm` command) that will create files or directories currently automounted.	The automounter is not active during an upgrade, so 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. (If you cannot remove a package, you can replace the symbolic link after the upgrade is completed.) **Note:** The `/var/mail` and `/var/news` directories, which usually reside on an automounted file system, are not affected by an upgrade.

Table 6-1 Preserving Local Modifications *(Continued)*

What to Do Before Upgrading	How to	Why
Prevent unneeded file systems from being mounted	Comment out file systems in the `/etc/vfstab` file that you do not want mounted during an upgrade.	During an upgrade, the Solaris installation program attempts to mount all the file systems listed in the `/etc/vfstab` file on the root file system being upgraded. If the Solaris installation program cannot mount a file system, it reports the failure and exits.

6

Booting and Installing Solaris: Interactive

This chapter provides procedures to boot a system and perform an interactive installation using the Solaris installation program. If you're using the:

- **Initial installation option** – you can choose the defaults shown and have software automatically laid out for you, or you can customize the software and file system layout.
- **Upgrade option** – you choose the disk for upgrading, add or delete software if desired, then start the upgrade.

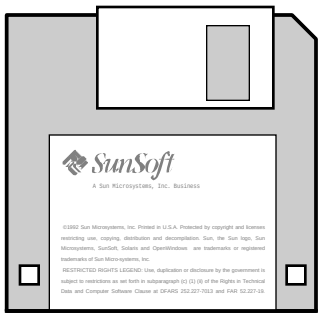
The procedure in this chapter should be done on the system that is being installed.

1 Make sure that the system's peripheral devices are properly configured.

Before installing Solaris software, you must configure peripheral devices so they do not conflict with each other, and so that Solaris software can access them. Configuring peripheral devices involves setting jumpers or running a software program under MS-DOS. See the *x86 Device Configuration Guide* for detailed instructions.

Caution – If peripheral devices are not correctly configured before you install the Solaris software, or if you are using unsupported devices, the Solaris installation program will fail.

2 Insert the Solaris boot diskette (shown below) into the system's a: diskette drive.

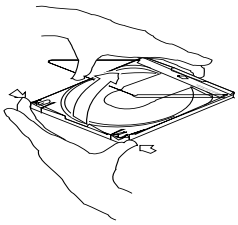


3 If you are using the system's local CD-ROM drive to install the Solaris software on the system, prepare the CD-ROM drive.

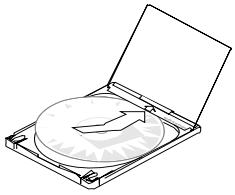
The instructions that follow cover the most common types of CD-ROM drives. If your CD-ROM drive is not one of the following types, see your hardware manual for instructions.

CD-ROM Drive - Caddy Version

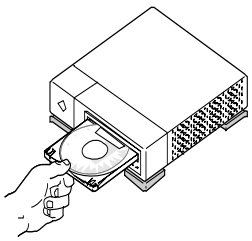
❶ To open the caddy, pinch the corners while lifting the lid at the center.



❷ Insert the Solaris CD (logo up) and close the caddy.

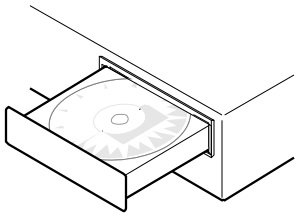


❸ With the arrow on the caddy pointing toward the CD-ROM drive, insert the caddy.



CD-ROM Drive - Tray Version

❶ Press the eject button on the front panel of the CD-ROM drive (a tray will emerge if you have this type of caddy).



❷ Insert the Solaris CD (logo up) into the tray.

❸ Push the tray back into CD-ROM drive.

4 Make sure you've reviewed the task map for an interactive installation (shown in Figure 2-2 on page 14).

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

Otherwise, the character installation interface will display improperly. You can use the `stty` command to find out the current dimensions of your tip window.

6 Follow the instructions before you boot the system:

If The System Is ...	Then ...
Off	**1)** Turn on the system components in the order recommended in your hardware guide. **2)** Go to Step 7.

On	**1)** If the system is running Solaris software, enter the following commands:
	`$ su root`
	`# halt`
	2) Go to Step 7.

7 Choose to boot from the local CD-ROM drive (CD) or from another system on the network (NET).

The following is a sample screen.

```
SunSoft Solaris x.x              Multiple Device Boot, vsn 2.0

               Solaris/x86 Multiple Device Boot Menu

Code   Device   Vendor   Model/Desc        Rev
=================================================
10     DISK     CONNER   CP3300-360MB-3.5  3236
11     CD       SONY     CD-ROM CDU-8012   3.1e
12     NET      SMC/WD   I/O=280 IRQ=10

               Enter the boot device code:


30
```

8 Choose the interactive option to install the Solaris software.

The following is a sample screen.

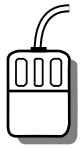
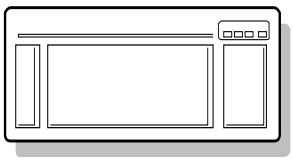
```
Select the type of installation you want to perform


      1  interactive
      2  custom JumpStart

Enter the number of your choice followed by the Enter key.

If you enter anything else, or if you wait for 15 seconds,
an interactive installation will be started
```

Caution – 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. For detailed instructions, go to Appendix E, “Troubleshooting,” page 176.

9 Install the Solaris software on the system by using the Solaris installation program displayed on the screen.

You are now in the hands of the Solaris installation program. This menu-driven, interactive program guides you step-by-step through installing Solaris software; it also has online help to answer your questions. As shown below, the program has two interfaces to accommodate different hardware.

If you want to record your responses, use the work sheet in Appendix A, “Worksheets for the Solaris Installation Program.” There is also a time zone map on page 180 to help you set your system’s time clock.

If Your System Has A ...	Then the Interface Is ...	And the Navigation Device Is ...
Graphics monitor	**Graphics-based** The Solaris Installation Program You are now interacting with the Solaris installation program. The program is divided into a series of short sections. At the end of each section, you will see a summary of the choices you’ve made, and be given the opportunity to make changes. Continue Exit Help	**Mouse** 
Non-graphics monitor, or an unsupported graphics card	**Character-based** --- The Solaris Installation Program You are now interacting with the Solaris installation program. The program is divided into a series of short sections. At the end of each section, you will see a summary of the choices you’ve made, and be given the opportunity to make changes. --- F2_Continue F5_Help	**Keyboard** 

10 If you used the upgrade option, see Chapter 9, "Completing an Upgrade," for information needed to complete an upgrade.

You must reboot the system after an upgrade.

11 Start the OpenWindows software to display the desktop:

```
$ /usr/openwin/bin/openwin
```

12 If you've allocated space for diskless clients, dataless clients, or AutoClient systems during the Solaris installation program, you must use the Solstice Host Manager to complete set up of these clients.

The Solaris installation program only allocates space for clients during an initial installation. The Solstice Host Manager completes client set up by providing their required directories. See the *System Administration Guide, Volume I.*

Task Complete

For post-installation information, see Chapter 10, "Where to Go After Installing Solaris."

Booting and Installing Solaris: Custom JumpStart 8

This chapter provides a procedure to boot a system and perform a custom JumpStart installation using profiles you've created to install Solaris software. If you're using the:

- **Initial installation option** – Solaris software is automatically installed on the system after you boot the system (or turn on the system in some cases).
- **Upgrade option** – the system is automatically upgraded to new version of Solaris software after you boot the system.

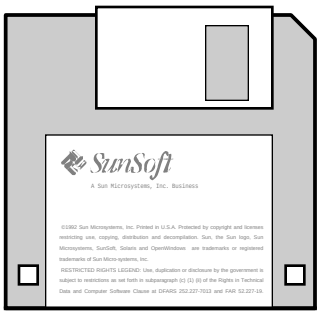
The procedure in this chapter should be done on the system that is being installed.

1 Make sure that the system's peripheral devices are properly configured.

Before installing Solaris software, you must configure peripheral devices so they do not conflict with each other, and so that Solaris software can access them. Configuring peripheral devices involves setting jumpers or running a software program under MS-DOS. See the *x86 Device Configuration Guide* for detailed instructions.

Caution – If peripheral devices are not correctly configured before you install the Solaris software, or if you are using unsupported devices, the Solaris installation program will fail.

2 Insert the Solaris boot diskette (shown below) into the system's a: diskette drive.



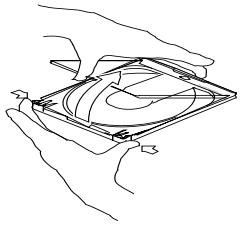
Note – If you are using a diskette to perform a custom JumpStart installation, you must insert a copy of the Solaris boot diskette that contains a JumpStart directory into the system's a: diskette drive.

3 If you are using the system's local CD-ROM drive to install the Solaris software on the system, prepare the CD-ROM drive.

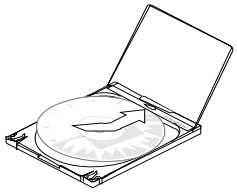
The instructions that follow cover the most common types of CD-ROM drives. If your CD-ROM drive is not one of the following types, see your hardware manual for instructions.

CD-ROM Drive - Caddy Version

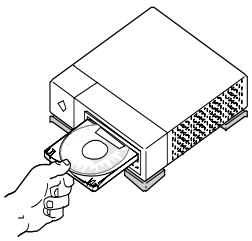
❶ To open the caddy, pinch the corners while lifting the lid at the center.



❷ Insert the Solaris CD (logo up) and close the caddy.

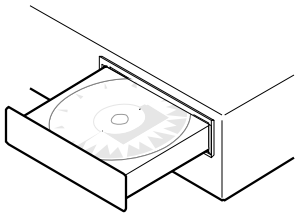


❸ With the arrow on the caddy pointing toward the CD-ROM drive, insert the caddy.



CD-ROM Drive - Tray Version

❶ Press the eject button on the front panel of the CD-ROM drive (a tray will emerge if you have this type of caddy).



❷ Insert the Solaris CD (logo up) into the tray.

❸ Push the tray back into CD-ROM drive.

4 Make sure you have reviewed the task map for a custom JumpStart installation (shown in Figure 2-3 on page 15).

5 Follow the instructions before you boot the system:

If The System Is ...	Then ...
Off	**1)** Turn on the system components in the order recommended in the hardware guide. **2)** Go to Step 6.

On	**1)** If the system is running Solaris, enter the following commands:
	`$ su root`
	`# halt`
	2) Go to Step 6.

6 If prompted, provide information about the system.

After the booting phase, the Solaris installation program may be displayed on the screen, prompting you to provide information about the system.

7 Choose to boot from the local CD-ROM drive (CD) or from another system on the network (NET).

The following is a sample screen[1].

```
SunSoft Solaris x.x                Multiple Device Boot, vsn 2.0

                Solaris/x86 Multiple Device Boot Menu

Code   Device    Vendor    Model/Desc         Rev
=======================================================
10     DISK      CONNER    CP3300-360MB-3.5   3236
11     CD        SONY      CD-ROM CDU-8012    3.1e
12     NET       SMC/WD    I/O=280 IRQ=10

                Enter the boot device code:


30
```

Caution – 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. For detailed instructions, go to Appendix E, “Troubleshooting,” page 176.

1. Sony® is a registered trademark of the Sony Corporation; Conner® is a registered trademark of Conner Peripherals, Inc.

8 Choose the custom JumpStart option to install the Solaris software.

The following is a sample screen.

```
Select the type of installation you want to perform


      1  interactive
      2  custom JumpStart

Enter the number of your choice followed by the Enter key.

If you enter anything else, or if you wait for 15 seconds,
an interactive installation will be started
```

9 Wait as the Solaris installation program automatically installs the Solaris software on the system.

You're done for awhile; installing Solaris software can take between 15 minutes and 2 hours.

10 Start the OpenWindows software to display the desktop:

```
$ /usr/openwin/bin/openwin
```

11 If you used the upgrade option, see Chapter 9, "Completing an Upgrade," for information needed to complete an upgrade.

You must reboot the system after an upgrade.

12 If you installed Solaris on a server and allocated space for diskless clients, dataless clients, or AutoClient systems, and you selected the initial installation option, you must use the Solstice Host Manager to complete set up of these clients.

The Solaris installation program only allocates space for clients during an initial installation. The Solstice Host Manager completes client set up by providing their required directories. See the *System Administration Guide, Volume I*.

Task Complete

For post-installation information, see Chapter 10, "Where to Go After Installing Solaris."

8

Completing an Upgrade 9

Overview

This chapter describes files to check and steps for completing an upgrade.

What Happened During the Upgrade

A record of what the installation program did during an upgrade can be important to determine if the upgrade was a success. The output of the upgrade is saved in the following files:

- If upgrade fails – `/a/var/sadm/system/logs/upgrade_log`
- If upgrade completes and system reboots – `/var/sadm/system/logs/upgrade_log`

Cleaning Up the System After an Upgrade

During an upgrade, the Solaris installation program merges local software modifications of the existing system with the new software; however, in some cases, it is not possible. The following file provides a list of the unpreserved local modifications during the upgrade that may need to be fixed:

- If upgrade fails –-`/a/var/sadm/system/data/upgrade_cleanup`
- If upgrade completes and system reboots – `/var/sadm/system/data/upgrade_cleanup`

9

Table 9-1 provides the entry descriptions of the unpreserved local modifications that may be found in the `upgrade cleanup` file.

Table 9-1 Entry Descriptions of the `upgrade_cleanup` File

Entry	Explanation	Recommended Action
file1: existing file renamed to *file2*	*file1* was on the system at the time of the upgrade. It had been modified since its original installation, so upgrade renamed *file1* to *file2* and saved the new file as *file1*.	Determine whether the modifications made to *file2* should be made to the new version of the file, *file1*.
file1: existing file preserved, the new version was installed as *file2*	*file1* has been preserved, and upgrade installed the new version of the file as *file2*.	Determine whether the differences in the new version of *file2* should be incorporated into the preserved version of *file1*.
file: had been deleted and has now been restored	*file* had been deleted from the system since its original installation, and the upgrade installed the new version of *file*.	Determine whether *file* should be deleted.
file: file type was changed from *type1* to *type2*	*file* has changed types since its original installation, from *type1* to *type2*. For example, you may have changed an actual file to be a symbolic link. The upgrade changed *file* back to its original type, *type1*.	Determine whether *file* should be changed back to *type2*.
file: target of symbolic link was changed from *target1* to *target2*	Since the original installation, *file* was changed to be a symbolic link to *target2* instead of *target1*. The upgrade changed *file* to point to its original *target1*.	Determine whether *file* should be changed to point to *target2*.
file1: target of hard link was changed from *file2*	When originally installed, *file1* was a hard link to *file2*. At the time of the upgrade, *file1* was no longer a hard link to *file2*. Upgrade restores the original hard link.	Determine whether *file1* should be changed to what it was before the upgrade.

How to 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 (SPARC, x86) and platform group (for example, sun4d, sun4L, 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` man page or *x86: Solaris 2.5 Installation Notes* or the *SPARC: Installing Solaris Software* for more instructions.

9

Where to Go After Installing Solaris 10

Table 10-1 shows the Solaris documentation that you may need after installing the Solaris software on a system. For a complete description of all the Solaris 2.4 documentation, refer to the *Solaris 2.5 Introduction.*

Table 10-1 Where to Go After Installing Solaris

Information Needed	**Manual Title**
Configuring additional devices on your system	*x86 Device Configuration Guide*
Adding and removing Solaris, third-party, or unbundled software	*Solstice AdminSuite 2.1 User's Guide*
Setting up mail accounts	*Mail Administration Guide*
Installing AnswerBooks	*System Administration Guide, Volume I* **Note:** The End User AnswerBook is the only AnswerBook installed with the Solaris software; and is only installed if the entire distribution software group is installed.
Setting up user accounts	*System Administration Guide, Volume I*

Table 10-1 Where to Go After Installing Solaris *(Continued)*

Information Needed	Manual Title
Halting a system	*System Administration Guide, Volume I*
Boot files	*System Administration Guide, Volume I*
Adding systems to a network	*System Administration Guide, Volume I*
Accessing remote files and systems	*System Administration Guide, Volume I*
Administering file systems	*System Administration Guide, Volume I*
Setting up system security	*System Administration Guide, Volume I*
CD-ROM and diskette drives	*System Administration Guide, Volume I*
Setting up printers	*System Administration Guide, Volume II*
Increasing your system's performance	*System Administration Guide, Volume II*
Managing disk use	*System Administration Guide, Volume II*
Examining and changing system information	*System Administration Guide, Volume II*
Using crontabs	*System Administration Guide, Volume II*
Adding and Maintaining Peripherals	*System Administration Guide, Volume II*
Accessing devices	*System Administration Guide, Volume II*
Setting up disks	*System Administration Guide, Volume II*

Table 10-1 Where to Go After Installing Solaris *(Continued)*

Information Needed	Manual Title
Terminals and modems, disk drives, tape drives, service access facility, connecting devices to serial port, format utility	*System Administration Guide, Volume II*
Using system administration tools	*Solstice AdminSuite 2.1 User's Guide*

10

Worksheets for the Solaris Installation Program

A

What's in the Worksheets

The worksheets list information that may need to be supplied during the Solaris installation program. There are separate worksheets for the initial and upgrade options.

How to Use the Worksheets

The worksheets can be used before installing Solaris software, or to record your responses when you install Solaris software. For example, if you're a system administrator of a large site and need to delegate the task of installing Solaris software to less-experienced staff, you can fill out the worksheet in advance; installers will know exactly how you want them to install Solaris software on their system, thus reducing the errors that come from guessing.

What shaded areas mean in the following table

Note – If you're using the following worksheet to gather information before installing Solaris software, you can ignore shaded areas if you are installing a system as a standalone, non-networked system.

A

Worksheet for the Initial Installation Option

The following worksheet lists important information you may need to supply when installing Solaris software on a system for the first time.

Information You May Need When Installing Solaris	**Description/Example**	**Work Space**
Boot device	Boot from local CD-ROM or network?	
Method for installing Solaris software	Interactive or custom JumpStart?	
Graphics card	Type of graphics card in your system. Example: Graphics Ultra Pro	
Screen size	Size of the screen on your monitor from the following list: - 14 inch - 16 inch - 19 inch	
Keyboard language	Language your keyboard supports. Example: UK english	
Pointing device	Type of mouse. Example: Logitech MouseMan, serial, 3-button	
Monitor type	Type of monitor. NEC-2a, MultiFrequency-38kHz	

Information You May Need When Installing Solaris	Description/Example	Work Space
Color depth number	Number of bits for the color plane that the graphics card supports. Example: 8, 24	
Graphics adaptor / monitor combination	Screen resolution. Example: 1024 X 768, 1280 X 1024	
Host name	Host name of the system. Example: sherlock	
Network connectivity	Is the system connected to a network?	
Primary network interface	Primary network interface for the system. Example: smc0	
IP address	Internet™ protocol address of the system you're installing. Example: 129.221.2.1	
Name service	Is the system a client of NIS+ or NIS? Part of another name service? Or is a name service not yet established? Example: Other, DCE	
Domain name	Domain in which the system resides. Example: chandy.West.Arp.COM	

A

Information You May Need When Installing Solaris	Description/Example	Work Space
Name server	Specify a name server for the system, or have software try to find one?	
Name server information	Name server's host name and IP address. Example: toucan, 125.221.2.1	
Subnet	Is the system part of a subnet?	
Netmask	Netmask of the subnet. Example: 255.255.0.0	
Time zone	Default time zone. Can be set by selecting geographic region, hours offset from GMT, or by pointing to a file in `/usr/share/zoneinfo` directory. Example: US/Mountain	
System type	System type: standalone system, OS server, or dataless client?	
Select platforms	All the platforms for clients that are *different* from the OS server's platform (see Appendix C, "Platform Names and Groups").	

Information You May Need When Installing Solaris	Description/Example	Work Space
Allocate client services	Number of clients the OS server will support. Example: 5 (default) Number of megabytes of swap, root, or both to assign clients. Example: 24 Mbytes (default)	
Dataless clients	**Note**: SunSoft plans to remove support for the dataless client system type after Solaris 2.5. You can select this system type now, but in future releases you will need to select a different type. Host names and IP addresses of the server(s) where dataless clients will mount their `/usr` file systems. Example: morton, 112.221.2.1, `/export/home`	
Languages	Language(s) to select for displaying the user interface after installing Solaris software. Example: French, Spanish	

A

Information You May Need When Installing Solaris	Description/Example	Work Space
Software	Which software group to install? • Core • End user system support • Developer system support • Entire distribution Should the software group be customized? **Note**: Removing or adding software often creates software dependencies; system administration knowledge is often required to fix dependencies.	
Disk(s) to install Solaris software on	Disk(s) for installing Solaris. Example: c0t0d0, c0t3d0, c0t4d1	
Create Solaris partition	If there is no Solaris partition on any of the disks selected, a Solaris partition must be created. Which partition, 1-4? What size should it be? Example: Solaris on partition 3, 198 Mbytes.	
Auto-layout file systems?	Should file systems be laid out on disks automatically or manually? **Note**: Manual layout requires advanced system administration skills.	

Information You May Need When Installing Solaris	Description/Example	Work Space
File systems to auto-layout	Which file systems should be used for auto-layout? Example: /, /opt, /var	
Preserve existing data?	Should existing data on disk(s) be preserved?	
Disk slices to preserve data	Which slices should be preserved? Example: /opt on c0t2d2	
File system and disk layout	Accept the layout of file systems as created by auto-layout or manual layout, or customize? **Note**: Customizing requires advanced system administration skills.	
Mount remote file systems?	Mount remote file systems from a server?	

Information You May Need When Installing Solaris	Description/Example	Work Space
Mount remote file systems	Provide remote file system information: • Local mount point • Server's host name • Server's IP address • File system path Example: Server: mitra Server IP address: 129.222.2.2 File system path: `/export/home` Local mount point: `/export`	
Reboot	Reboot after installing Solaris?	
Root password	Create root password. Example: cat6file	

Worksheet for the Upgrade Option

The following worksheet lists the important information you may need to supply when upgrading a system from Solaris 2.4 or later.

Information You May Need When Upgrading Solaris	**Description/Example**	**Work Space**
Boot device	Boot from local CD-ROM or network?	
Method for installing Solaris software	Interactive or custom JumpStart?	
Graphics card	Type of graphics card in your system. Example: Graphics Ultra Pro	
Screen size	Size of the screen on your monitor: - 14 inch - 16 inch - 19 inch	
Keyboard language	Language your keyboard supports. Example: UK english	
Pointing device	Type of mouse. Example: Logitech MouseMan, serial, 3-button	
Monitor type	Type of monitor. NEC-2a, MultiFrequency-38kHz	

A

Information You May Need When Upgrading Solaris	Description/Example	Work Space
Color depth number	Number of bits for the color plane that the graphics card supports. Example: 8, 24	
Graphics adaptor / monitor combination	Screen resolution. Example: 1024 X 768, 1280 X 1024	
Host name	Host name of the system Example: sherlock	
Network connectivity	Is the system connected to a network?	
Primary network interface	Primary network interface for the system. Example: le0	
IP address	Internet protocol address of the system you're installing. Example: 129.221.2.1	
Name service	Is the system a client of NIS+ or NIS? Part of another name service? Or is a name service not yet established? Example: Other, DCE	
Domain name	Domain in which the system resides. Example: chandon	

Information You May Need When Upgrading Solaris	Description/Example	Work Space
Name server	Specify a name server for the system, or have software try to find one?	
Name server information	Name server's host name and IP address. Example: toucan, 125.221.2.1	
Subnet	Is the system part of a subnet?	
Netmask	Netmask of the subnet. Example: 255.255.0.0	
Time zone	Default time zone. Can be set by selecting geographic region, hours offset from GMT, or by pointing to a file in `/usr/share/zoneinfo` directory. Example: US/Mountain	
Installing Solaris - upgrade or initial?	Choose the upgrade or initial option for installing Solaris.	
Disks to upgrade	Select a disk to use for upgrading to new version of Solaris. Example: c0d0t0	

A

Information You May Need When Upgrading Solaris	Description/Example	Work Space
Languages	Language(s) to select for displaying the user interface after installing Solaris software. Example: French, Spanish	
Customizing software?	Do you want to customize software before starting the upgrade? **Note**: Removing or adding software often creates software dependencies; system administration knowledge is often required to fix dependencies.	

System Identification Label B

For your convenience, the following pages contain system identification labels (a sample, and one for photocopying). Attach them to systems at your site so users have access to information needed to install Solaris software and operate their system.

B

Sample - System Identification Label

As shown in the following sample, system identification labels can provide a quick reference to important system information.

Host name: roylat

System type

- ☑ Standalone system
- ☐ OS server
- ☐ Diskless client
- ☐ Dataless client
- ☐ AutoClient system

Boot device	network
Graphics card	Graphics Ultra Pro
Screen size	14”
Pointing device	Logitech MouseMan, serial, 3-button
Screen resolution	1024 X 768
Networked?	Yes
Primary network interface	smc0
IP address	129.221.2.1
Name service	NIS+
Domain name	hawthorne.West.Arp.COM
System part of a subnet?	Yes
Netmask	225.225.225.225
Name server and IP address	hudson, 125.222.2.2
Ethernet address	8:0:20:b:40:e7

System Identification Label

Photocopying the following label and identifying system information for each system provides a quick reference for users.

Host name:

System type

- ☐ Standalone system
- ☐ OS server
- ☐ Diskless client
- ☐ Dataless client
- ☐ AutoClient system

Boot device _____

Graphics card _____

Screen size _____

Pointing device _____

Screen resolution _____

Networked? _____

Primary network interface _____

IP address _____

Name service _____

Domain name _____

System part of a subnet? _____

Netmask _____

Name server and IP address _____

Ethernet address _____

B

Platform Names and Groups C

Table C-1 shows the platform names of various hardware platforms. You may need this information when preparing a system to install Solaris software.

Use `uname -i` to determine a system's platform name; use `uname -m` to determine a system's platform group.

Table C-1 Platform Names and Groups

System	Platform Name	Platform Group
x86	i86pc	i86pc
SPARCstation 1	SUNW,Sun_4_60	sun4c
SPARCstation1+	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
SPARCcenter 1000	SUNW,SPARCserver-1000	sun4d
SPARCcenter 2000	SUNW,SPARCcenter-2000	sun4d

Table C-1 Platform Names and Groups

System	Platform Name	Platform Group
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
SPARCserver6xx	SUNW,SPARCsystem-600	sun4m
SPARCstation LX	SUNW,SPARCstation-LX	sun4m
SPARCstation LX+	SUNW,SPARCstation-LX+	sun4m
SPARCclassic	SUNW,SPARCclassic	sun4m
SPARCclassic X	SUNW,SPARCclassic-X	sun4m
SPARCengine EC3	SUNW,SPARCengine-EC-3	sun4m
SPARCstation Voyager	SUNW,S240	sun4m
Sun Ultra 1 Model 140	SUNW,Ultra1-140	sun4u
Other SPARC systems	See your hardware vendor documentation for platform name information.	

Sample Custom JumpStart Installation

D

This example shows a set of steps a system administrator would take to do a custom JumpStart installation for a fictitious site.

Sample Site Setup

Figure D-1 shows the sample site setup for this example.

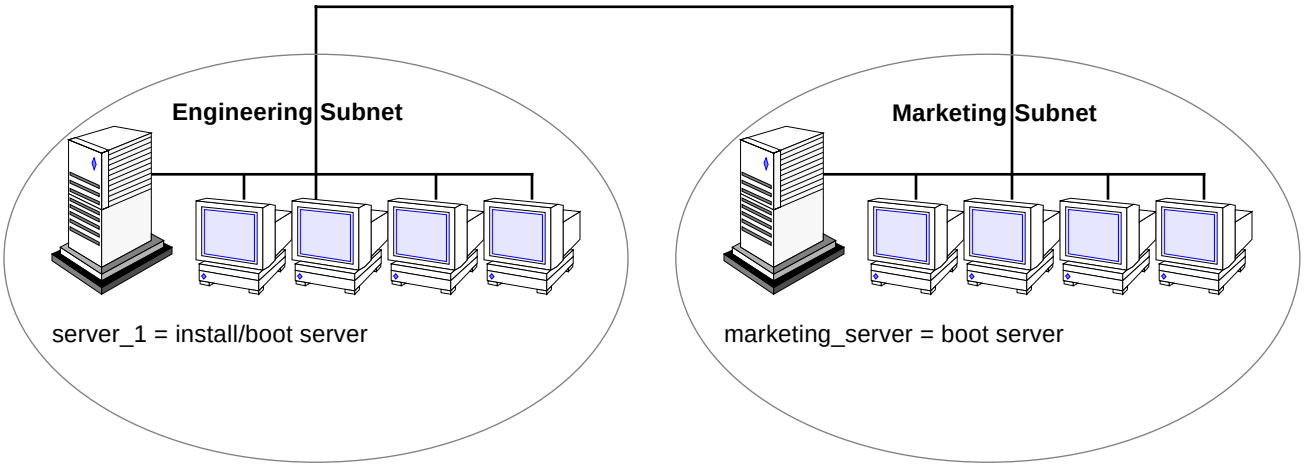


Figure D-1 Sample Site Setup

At this fictitious site:

- The engineering group is on its own subnet. This group uses NEC® Powermate® 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 are in NIS+ tables.
- The engineering server named `server_1` has a copy of Solaris 2.4 software on its local disk in a directory named `/export/install`. Both the engineering and marketing groups will install Solaris software over the network from `server_1`.

1 Create a JumpStart directory.

The administrator sets up a JumpStart directory on the install server, `server_1`. This directory will hold files necessary for a custom JumpStart installation of Solaris software. The easiest way to set up this directory is to copy the sample directory from the copy of the Solaris CD that has been put in `/export/install`.

```
# cp -r /export/install/auto_install_sample /jumpstart
```

2 Share the JumpStart directory.

The system administrator shares the `/jumpstart` directory so that the `rules` file and profiles are accessible to systems on the network. To accomplish this, the administrator adds the following line to the `/etc/dfs/dfstab` file:

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

Then, at the command line, the administrator uses the `unshareall` and `shareall` commands:

```
# unshareall
# shareall
```

3 Create the `eng_profile` profile.

The administrator creates a file named `eng_profile` in the `/jumpstart` directory. The `eng_profile` file has the following entries, which define the Solaris software to be installed on systems in the engineering group.

```
❶ install_type  initial_install
❷ system_type   standalone
❸ partitioning  default
❹ cluster       SUNWCprog
❺ filesys       any 50 swap
```

❶ Specifies that the installation will be treated as an initial installation, as opposed to an upgrade.

❷ Specifies that the engineering systems are standalone systems.

❸ Specifies that the JumpStart software uses default disk partitioning for installing Solaris on the engineering systems.

❹ Specifies that the developer's software cluster will be installed.

❺ Specifies that each system in the engineering group will have 50 Mbytes of swap space.

4 Create the `marketing_profile` profile.

The administrator creates a file named `marketing_profile` in the `/jumpstart` directory. The `marketing_profile` file has the following entries, which define the Solaris software to be installed on systems in the marketing group.

```
❶ install_type  initial_install
❷ system_type   standalone
❸ partitioning  default
❹ cluster       SUNWCuser
❺ package       SUNWaudmo
```

❶ Specifies that the installation will be treated as an initial installation, as opposed to an upgrade.

❷ Specifies that the marketing systems are standalone systems.

❸ Specifies that the JumpStart software will use default disk partitioning for installing Solaris on the marketing systems.

❹ Specifies that the end user software cluster is to be installed.

❺ Specifies that the audio demo software package is to be added to each system.

5 Edit the `rules` file.

The administrator must define the `rules` file. The Solaris installation program will use the contents of this file to select the proper installation for each department.

At this site, each department is on its own subnet and has its own network address. The administrator uses this information to control how systems are installed. The engineering department is on subnet 255.222.43.0, and marketing is on 255.222.44.0.

In the `/jumpstart` directory, the administrator edits the `rules` file, deletes all of the example rules, and enters:

```
network 255.222.43.0 - eng_profile    -
network 255.222.44.0 - marketing_profile -
```

Note – These are sample rules in which an administrator uses a network address to identify which systems will be installed with the `eng_profile` and `marketing_profile`, respectively. The administrator could also have chosen to use host names, memory size, or model type as the rule keyword. See “Rule Keyword and Rule Value Descriptions” on page 84 for a complete list of keywords you can use in a `rules` file.

6 Execute the `check` script.

After the `rules` and profile files are properly set up, the system administrator runs the `check` script to verify the files.

```
# cd /jumpstart
# ./check
```

When `check` finds no errors, it creates the `rules.ok` file.

7 Set up the engineering systems for installation.

After setting up the `/jumpstart` directory and appropriate files, the administrator sets up the install server to install Solaris software on the engineering systems.

The administrator first sets up the engineering systems because they are on the same subnet as the install server. On the install server, the administrator uses the `add_install_client` command:

```
# cd /export/install
# ./add_install_client -c server_1:/jumpstart host_eng1 i386pc
# ./add_install_client -c server_1:/jumpstart host_eng2 i386pc
    .
    .
    .
```

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.
`i386pc`	Specifies the platform group of the systems that will use `server_1` as an install server. (This is the proper platform name for x86 systems.)

8 Set up the marketing systems for installation.

Systems cannot boot from an install server on a different subnet, so the administrator sets up a boot server on the marketing group's subnet. On a server on the marketing subnet, the administrator inserts a Solaris CD. The administrator then uses the `setup_install_server` command to copy the boot software from the Solaris CD to the marketing server's local disk.

```
# cd /cdrom/cdrom0/s0
# ./setup_install_server -b /marketing/boot-dir i86pc
```

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 `/marketing/boot-dir`.
`i86pc`	Specifies the platform group of the systems that will use this boot server. (This is the proper platform name for x86 systems.)

Next, the administrator sets up the marketing systems to boot from the local boot server and install Solaris from the remote install server. The administrator uses the `add_install_client` command on the marketing group's boot server:

```
# cd /marketing/boot-dir
# ./add_install_client -s server_1:/export/install -c server_1:/jumpstart host_mkt1 i86pc
# ./add_install_client -s server_1:/export/install -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`).
`-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_mkt2`	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 proper platform name for x86 systems.)

9 Boot the systems and install Solaris software.

The administrator boots the engineering systems and marketing systems by using the Solaris boot diskette.

Troubleshooting

E

This appendix describes problems you may encounter when installing Solaris software, and suggests possible solutions.

The following table shows common error messages and the page number where you can find causes and possible solutions.

E

Specific Installation Errors

```
Error: Unknown client "host_name"
```

Reason Error Occurred

The *host_name* argument in the `add_install_client` command must be a host in the name service.

How to Fix the Problem

Add the host *host_name* to the NIS or NIS+ name service. (Either update the NIS maps or populate the NIS+ tables.) Try executing the `add_install_client` command again.

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

Reason Error Occurred

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.

How to Fix the Problem

Ignore this error message. If multicast doesn't work, IP uses layer broadcasts instead.

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

Reason Error Occurred

This system is not connected to the network.

How to Fix the Problem

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
```

Reason Error Occurred

This message cannot find the proper media for booting.

How to Fix the Problem

Verify that you are using the correct boot command for your system. If installing from a CD-ROM drive, make sure the Solaris CD is in the CD-ROM drive and that the CD-ROM drive is mounted.

E

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

Reason Error Occurred

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 problems may cause this error:

- There may be `/etc/bootparams` files on different servers with an entry for this install client.
- There may be multiple `/tftpboot` or `/rplpboot` directory entries for this install client.
- 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.

How to Fix the Problem

Examine the network setup:

- Ensure 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.
- Ensure 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.
- 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.

```
Not a UFS filesystem
```

Reason Error Occurred

When Solaris software was installed (either through the interactive or custom JumpStart), the default boot drive was not selected. When an alternate boot disk is selected, you must use the Solaris boot diskette to boot the system from that point on.

How to Fix the Problem

Insert the Solaris boot diskette into the system's a: diskette drive.

```
pkgrm: ERROR: class action script did not complete successfully. Removal of
<SUNWuodte> failed.
```

Reason Error Occurred

Problem in Solaris 2.4 and early 2.5 with building the package.

How to Fix the Problem

Ignore the message.

```
Requesting Internet address for Ethernet_Address
```

Reason Error Occurred

The client is trying to boot over the network, but it cannot find a system that knows about it.

How to Fix the Problem

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...
```

Reason Error Occurred

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.

How to Fix the Problem

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...
```

Reason Error Occurred

The system is trying to boot over the network, but the server is not set up to boot this system.

How to Fix the Problem

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 Installation 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

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.

E

Problem	How to Fix the Problem
OpenWindows is not available to diskless and dataless clients.	The `/usr/openwin` may be on a separate, mounted file system. This directory is not automatically shared by `admintool`. The `/usr/openwin` file system must be manually added to the server's `/etc/dfs/dfstab` file and the file system must be shared. The entry looks like this: `share -F nfs /usr/openwin` Also, an explicit mount of the file system must be made in the client's `/etc/vfstab` file using the following: *server_name*`:/usr/openwin - /usr/openwin nfs - yes -`

Problem	How To Fix the Problem
Several processes that normally run when booting the system are not run when the system boots from the CD. (This is done to enable the system to boot and the Solaris installation program to run with only 16 Mbytes of memory.)	To boot from the CD to do disaster recovery (that is, restoring the root file system), you need to add functionality normally provided when booting. To provide this functionality, a script is provided, `/sbin/setup_cd`, which can be run to perform the additional setup that might be needed to do disaster recovery. `setup_cd` does the following tasks: – Finishes setting up the device file system for tapes and ports – Enables routing – Enables multicast – Invokes the `statd(1M)` daemon – Invokes the `lockd(1M)` daemon – Starts the `automounter` Note that if `setup_cd` is run on a 16 Mbyte system, it may not be possible to run the Solaris installation program or other programs that need more memory. If a disk with a `swap` partition is available, you can avoid this problem by using the `swap(1M)` command to add swap space to the system before invoking `setup_cd.` When booting from the Solaris CD during disaster recovery, it is possible to boot the system in single user mode from the CD-ROM. For example, at the menu option to install using interactive or custom JumpStart, type: **`b -s`**

Problem	How to Fix the Problem
An installation using the upgrade option fails for reasons beyond your control, such as a power failure or a network connection failure. The system may be 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 installation program will determine if the system has been partially upgraded and will continue the upgrade.

E

Problem

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.

How to Fix the Problem

To perform surface analysis on an IDE disk, follow this procedure:

1) Start an interactive installation, as described in Chapter 7, "Booting and Installing Solaris: Interactive." The Solaris 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. Type `format`.
7) 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.

6) 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.)
7) At the `format>` prompt, type `analyze`.
8) 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`.
9) 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.
10) At the `analyze>` prompt, type `quit`.
11) You may want to specify blocks to re-map. If so, at the `format>` prompt, type `repair`.
12) Type `quit` to quit the `format` program.
13) Type `installtool` to resume the GUI installation or `suninstall` to resume the CUI installation.

Problem	How to Fix the Problem
The Solaris root slice must reside within the first 1024 cylinders of the disk. If it is 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 and try booting again. If the first fdisk partition is extended DOS (EXT DOS) or another operating system, use the `fdisk` program to delete it and try booting again.

E

Time Zones

The next page shows time zones of the world by hours offset from Greenwich Meantime. This may be useful when setting a system's clock during the Solaris installation program.

Figure F-1 reflects Standard Time. If daylight saving time is in effect, add one hour.

F

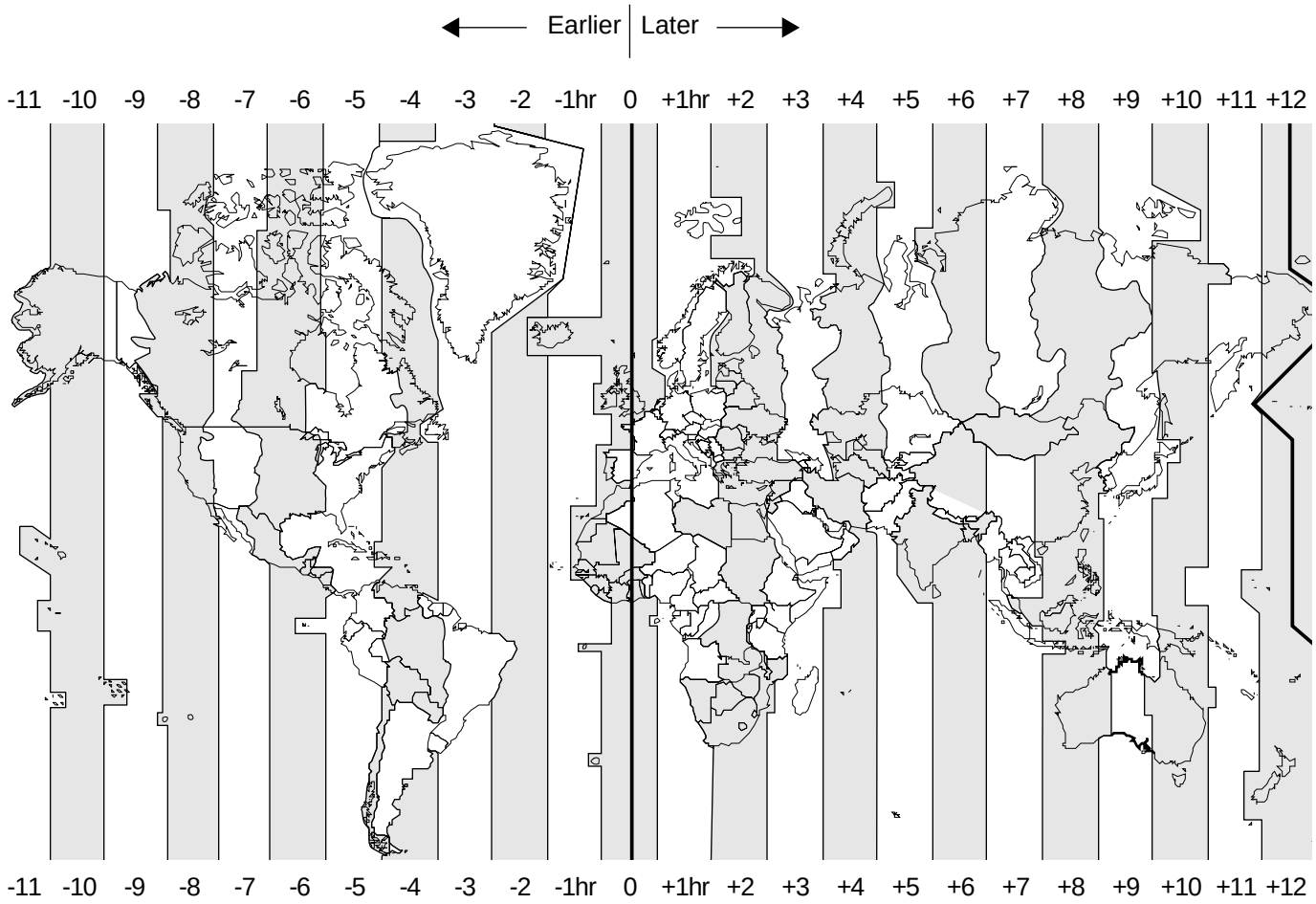


Figure F-1 Greenwich Meantime Map

Preserving MS-DOS and Installing the Solaris Operating Environment G

Many x86 systems are preconfigured with the MS-DOS operating system. It's common on these preconfigured systems to have MS-DOS consume the entire disk. It's also common to have data on the MS-DOS partition that you don't want to lose when you install Solaris software. (When you install Solaris software, it writes over any existing data on the disk.) However, you can back up any data on the MS-DOS partition and then restore it after you have installed the Solaris software. This section describes this procedure.

Note – Although this procedure describes how to save and restore data on an MS-DOS partition, you can use the same general procedure to back up data on any existing operating system before installing Solaris software.

---

1 Back up the existing MS-DOS data.

Use the MS-DOS backup program. See the MS-DOS documentation for detailed information.

2 Install Solaris software and manually create a Solaris and an MS-DOS fdisk partition.

You can use either an interactive installation or a custom JumpStart installation.

If You Are Doing ...	Then ...
An interactive installation	See Chapter 7, "Booting and Installing Solaris: Interactive." During the interactive installation, you are prompted to select the disk(s) you want to install Solaris software on. After selecting disks, you can create both the Solaris fdisk partition and the MS-DOS fdisk partition. Also, if you want to reserve space on the disk(s) for other operating systems, create those fdisk partitions at this time in the Solaris installations. After completing the Solaris software installation, continue with Step 3 on page 183.
A custom JumpStart installation	See Chapter 8, "Booting and Installing Solaris: Custom JumpStart." **Note:** Be sure that the installation profiles specifies a Solaris fdisk partition and an MS-DOS fdisk partition. After completing the Solaris software installation, continue with Step 3 on page 183.

Note – The Solaris software program will not write to the MS-DOS partition, leaving space there for you to install the MS-DOS operating system and to restore data on it after Solaris is installed.

3 After Solaris software is installed, halt the system.

4 Boot the system and install MS-DOS from the MS-DOS setup and installation disks.

Note – 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.

5 Restore any backed-up data to the MS-DOS partition.

Use the MS-DOS backup program to restore backed-up files. See the MS-DOS documentation for detailed information.

6 Set the active fdisk partition to the Solaris partition.

The active partition is the partition that the system automatically boots from. After installing MS-DOS and restoring backed-up files, the MS-DOS partition will be the active fdisk partition. Use the MS-DOS fdisk program to set the active fdisk partition to the Solaris partition.

7 Reboot the system.

The system will boot from the Solaris fdisk partition and start running the Solaris software.

G

Glossary H

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 four main *software groups*, which are each composed of clusters and *packages*.

H

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.

dataless client

A networked system that has its own disk on which it maintains its own root (/) file system and swap space. However, a dataless client relies on an OS server for its `/usr` system.

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.

EISA

Extended Industry Standard Architecture. A type of bus on x86 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 systems. During the Solaris installation program, you must set up at least one Solaris fdisk partition on an x86 system. x86 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.

H

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 installation image for other systems on a network to boot and install from (also known as a *media server*). The Solaris installation image can reside on the install server's CD-ROM drive or hard disk.

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 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 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 server for custom JumpStart installations, the JumpStart directory is a directory on the server that contains all the essential custom JumpStart files.

locale

A specific language associated with a region or territory.

MCA

Micro Channel Architecture. A type of bus on x86 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`).

H

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 dataless clients, an OS server must provide the `/usr` file system. To serve autoclients, 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 four main *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 used as a template by the custom JumpStart installation software. It defines how to install the Solaris software on a system (for example, initial installation or upgrade option, system type, disk partitioning, software group), and it is named in the `rules` file. See *rules file*.

/ (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 used to create the `rules.ok` file. The `rules` file is a look-up table consisting of one or more rules that define matches between system attributes and profiles. See *profile*.

H

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 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, dataless client, and diskless client, AutoClient system, OS server. However, the only system types that are covered in this document are standalone system, dataless client, and OS server.

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. A dataless client must share (mount) `/usr` from a file server; it does not have its own `/usr` file system. 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.

H

Index

Symbols

Numerics

A

B

D

E

F

G

H

I

J

N

O

P

R

S

T

U

X

Z

Revision History

Release	Date	Comments
Solaris 2.5	November 1995	**Change to Service Setup For Clients During Installation** The Solaris installation program (interactive and custom JumpStart) no longer sets up services for clients. During installation, selecting the system type 'OS server' only allocates space for clients; to complete client set up you must use Solstice Host Manager after Solaris software is installed. **New bootparams Keyword/Value** A new bootparams keyword / value forces `sysidtool` to attempt to configure a specified name 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 Manager, and `add_install_client` script (where -n <*ns_string*> is the string to put in the bootparams table). **Change of Location of Diskless Client Booting Information** Information on how to boot diskless clients has been moved to the *System Administration Guide, Volume I*

Release	Date	Comments
		Solstice Host Manager Replaces `add_install_client` The Solstice Host Manager now supports remote installations, and is the recommended tool for setting up network install servers (instead of the `add_install_client` command). Solstice Host Manager can also now be used to set up custom JumpStart installations. **Hardware Support Dropped** The sun4 and sun4e hardware is no longer supported in Solaris 2.4 **Underlying Software Changes** 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).
Solaris 2.4	October 1994	**Book Rewrite** The *System Configuration and Installation Guide* was rewritten. Procedures for installing Solaris software for x86 and SPARC were separated into two books and the titles changed to: *x86: Installing Solaris Software* and *SPARC: Installing Solaris Software*. **New Interfaces For Interactive Installations** New graphical user interface (GUI) and character user interface (CUI) were added for installing Solaris software.