SPARC: Installing Solaris Software

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

SunSoft
A Sun Microsystems, Inc. Business

Portions of this product may be derived from the UNIX® and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's Font Suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS
Sun, the Sun logo, Sun Microsystems, Sun Microsystems Computer Corporation, SunSoft, the SunSoft logo, Solaris, SunOS, OpenWindows, DeskSet, ONC, ONC+, Sun-4, IPC, IPX, SLC, ELC, JumpStart, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and certain other countries. UNIX is a registered trademark of Novell, Inc., in the United States and other countries; X/Open Company, Ltd., is the exclusive licensor of such trademark. OPEN LOOK® is a registered trademark of Novell, Inc. PostScript and Display PostScript are trademarks of Adobe Systems, Inc. DCE® is a registered trademark of DCE Group Limited. Internet ™ is a trademark of Internet, Inc. MS-DOS® is a registered trademark of Microsoft Corporation. Motif® is a registered trademark of Open Software Foundation, Inc. Sony® is a registered trademark of the Sony Corporation. Conner® is a registered trademark of Conner Peripherals, Inc. NEC® is a registered trademark of Nippon Electric Corporation, Inc. Powermate® is a registered trademark of NEC Corporation, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCstorage, SPARCware, SPARCcenter, SPARCclassic, SPARCcluster, SPARCdesign, SPARC811, SPARCprinter, UltraSPARC, microSPARC, SPARCworks, and SPARCompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a trademark and product of the Massachusetts Institute of Technology.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Please
Recycle

Adobe PostScript

Contents

About This Book ix

1. About Installing Solaris. 1

Definition: Installing Solaris Software. 1

2. Preparing to Install Solaris 5

Why You Should Not Ignore This Chapter. 5

Steps to Prepare to Install Solaris. 6

3. Preparing to Install Solaris Over a Network 17

About Installing Solaris Over a Network 17

Servers Required for Network Installation. 18

Requirements for Hands-Off Network Installation. 20

Files You Should Know About. 21

Commands You Should Know About. 22

Creating an Install Server 24

▼ How to Create an Install Server. 25

▼ How to Create a Boot Server on a Subnet 28

Setting Up Servers for Network Installation 30

▼ How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client 31

▼ How to Set Up Servers for Network Installation of a Diskless Client. 41

Preconfiguring Network Information 45

▼ How to Preconfigure Network Information. 45

Preconfiguring the Default Locale 48

▼ How to Preconfigure Default Locale Using NIS 48

▼ How to Preconfigure Default Locale Using NIS+ 51

4. Preparing Custom JumpStart Installations 55

Definition: Custom JumpStart Installation 55

Reasons to Choose a Custom JumpStart Installation 56

Tasks to Prepare for Custom JumpStart Installations 57

What Happens During a Custom JumpStart Installation 58

Creating a JumpStart Directory on a Diskette 61

▼ How to Create a JumpStart Directory on a Diskette for x86 Systems 61

▼ How to Create a JumpStart Directory on a Diskette for SPARC Systems 67

Creating a JumpStart Directory on a Server 71

▼ How to Create a JumpStart Directory on a Server 71

Enabling All Systems to Access the JumpStart Directory 74

▼ How to Enable All Systems to Access the JumpStart Directory 74

Creating a Profile 76

What Is a Profile 76

Requirements for Profiles 76

▼ How to Create a Profile 77

Profile Examples 78

Profile Keyword and Profile Value Descriptions 81

How the Size of Swap Is Determined 91

Creating the `rules` File 92

What Is the `rules` File 92

When Does a System Match a Rule 92

▼ How to Create the `rules` File 93

Rule Examples 96

Important Information About the `rules` File 97

Rule Keyword and Rule Value Descriptions 98

How the Installation Program Sets the Value of `rootdisk` 102

Using `check` to Validate the `rules` File 103

▼ How to Use `check` to Validate the `rules` File 104

5. Using Optional Custom JumpStart Features 107

Overview 107

Creating Begin Scripts 108

What Is a Begin Script 108

Important Information About Begin Scripts 108

Ideas for Begin Scripts 108

Creating Derived Profiles With Begin Scripts 108

Creating Finish Scripts 110

What Is a Finish Script 110

Important Information About Finish Scripts 110

Ideas for Finish Scripts 110

Adding Files With Finish Scripts 111

Customizing the Root Environment 112

Setting the System’s Root Password With Finish Scripts . . 112

Using `pfinstall` to Test Profiles 114

Why Use `pfinstall` 114

Ways to Use `pfinstall` 114

▼ How to Use `pfinstall` to Test a Profile 115

`pfinstall` Examples 116

▼ How to Create a Disk Configuration File for a SPARC System 117

▼ How to Create a Multiple Disk Configuration File for a SPARC System 119

Using a Site-Specific Installation Program 121

6. Preparing a System for Upgrade 123

Overview. 123

Can You Use Upgrade? 123

Backing Up Your System. 124

Preserving Local Modifications 125

7. Booting and Installing Solaris: JumpStart 127

8. Booting and Installing Solaris: Interactive. 133

9. Booting and Installing Solaris: Custom JumpStart. 139

10. Completing an Upgrade. 145

Overview. 145

What Happened During the Upgrade. 145

Cleaning Up the System After an Upgrade 145

11. Where to Go After Installing Solaris. 147

A. Work Sheets for the Solaris Installation Program. 149

B. System Identification Label 159

C. Kernel Architectures. 163

D. Sample Custom JumpStart Installation 165

E. Troubleshooting 173

F. Time Zones. 181

G. Glossary 183

Index 191

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 your hardware guides.

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: JumpStart

8 Installing Solaris: Interactive

9 Installing Solaris: Custom JumpStart

After Installing Solaris

10 Completing an Upgrade

11 Where To Go After Installing Solaris

Related Books

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

- *SPARC Hardware Platform Guide*

  Contains vendor-specific information for installing Solaris software.

- *Solaris 2.4 Open Issues and Late-Breaking News*

  Describes any late-breaking news about installing Solaris software including known problems.

- *Administration Application Reference Manual*

  Describes applications such as Administration Tool that you may use if you’re setting up network installations.

- *File System Administration*

  Describes how to back up system files.

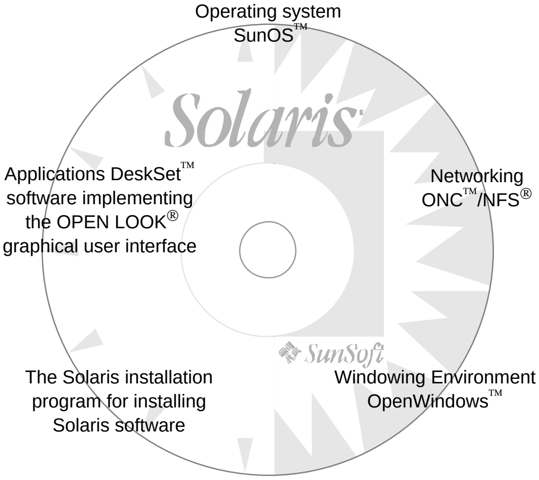
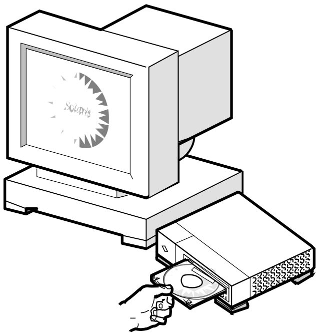
- *Solaris 1.x to Solaris 2.x Transition Guide*

  Describes transition issues including backing up 4.1.x files before installing Solaris software, and restoring files after Solaris software is installed.

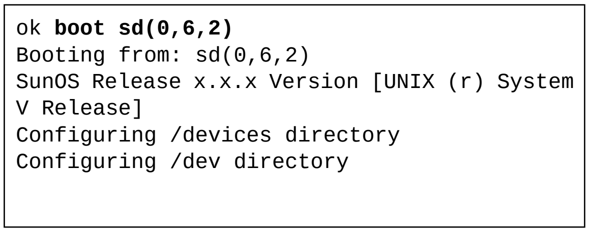
About Installing Solaris 1

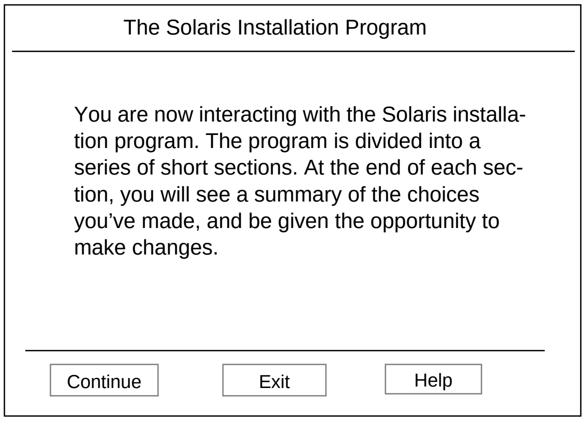
Definition: Installing Solaris Software

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.

Stage	Diagram	What Happens
1	 Operating system SunOS™ Solaris Applications DeskSet™ software implementing the OPEN LOOK® graphical user interface Networking ONC™/NFS® SunSoft The Solaris installation program for installing Solaris software Windowing Environment OpenWindows™	You choose a Solaris CD for your hardware platform. The Solaris CD contains the SunOS operating system and other software.
2		You load the Solaris CD into the CD-ROM drive.

Stage	Diagram	What Happens

3	
	`ok `**`boot sd(0,6,2)`**
	`Booting from: sd(0,6,2)`
	`SunOS Release x.x.x Version [UNIX (r) System V Release]`
	`Configuring /devices directory`
	`Configuring /dev directory`
		After you power on the system, you boot the system using a boot command specific to your system hardware.
		During the booting phase, checks are performed on the hardware.

4	 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. 	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.

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:

Determine if your system is networked.	*page 6*
Determine your system type.	*page 6*
Determine if you have required hardware.	*page 8*
If you have a standalone system, determine if you have enough disk space to install Solaris software.	*page 9*
Back up your system.	*page 10*
Choose a method for installing Solaris software.	*page 11*
Choose a task map and complete required tasks.	*page 12*

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.

Steps to Prepare to Install Solaris

Follow these steps before turning on your system.

1 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.

2 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 7, choose a system type from the following lists:

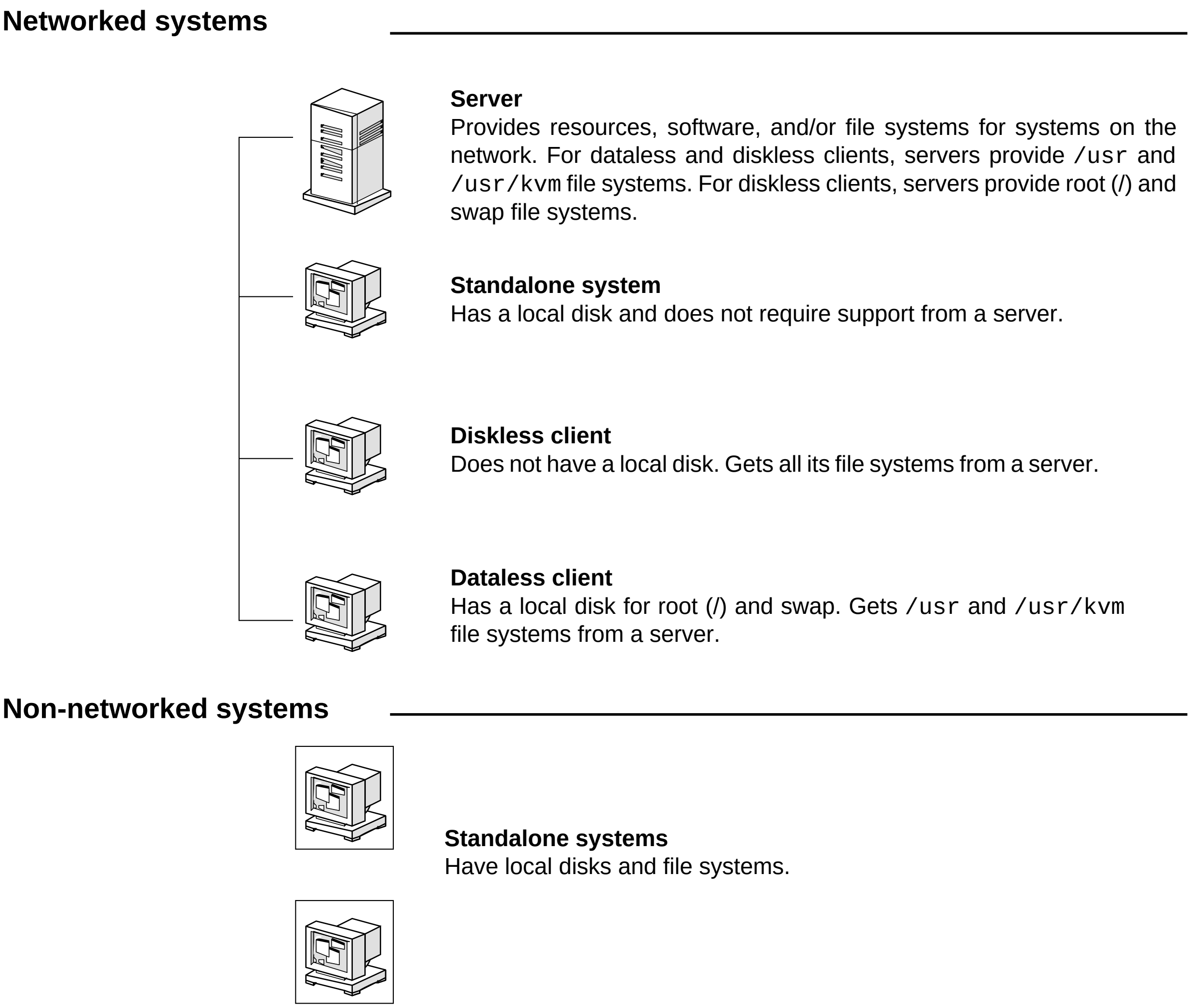
Networked systems:

- *Server*
- *Standalone system*
- *Dataless client*
- *Diskless client*

Non-networked systems:

- *Standalone system*

Note – If you choose a diskless client as the system type, you do not need go any further in this chapter. Go to “How to Set Up Servers for Network Installation of a Diskless Client” on page 41.



Networked systems

Server
Provides resources, software, and/or file systems for systems on the network. For dataless and diskless clients, servers provide `/usr` and `/usr/kvm` file systems. For diskless clients, servers provide root (/) and swap file systems.

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

Diskless client
Does not have a local disk. Gets all its file systems from a server.

Dataless client
Has a local disk for root (/) and swap. Gets `/usr` and `/usr/kvm` file systems from a server.

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.

3 Determine if you have required hardware.

For a detailed description of hardware requirements, see your hardware guide.

Table 2-1 Hardware Requirements

Hardware Platform	**Minimum Memory**	**Disk Interfaces**	**Buses**	**Device for Installing Solaris Software**
SPARC system [1]	16 Mbytes	• SMD • IPI • SCSI	• VMEbus • Sbus	You must have one of the following devices for installing Solaris software: • Local CD-ROM drive • Remote CD-ROM drive available over the network • Remote hard disk available over the network

[1] See the *SPARC Hardware Platform Guide* for a list of supported systems.

4 If you have a standalone system, determine if you have enough disk space to install Solaris software.

The software on the Solaris CD is bundled into four *software groups* (see Table 2-2). You must choose one of these software groups when you install Solaris software. Use Table 2-2 to determine approximately how much disk space you'll need. Keep in mind:

- The disk space shown does not account for file system overhead or local file systems required for electronic mail, printer spooling, users' personal file systems, or swap space.
- You can remove software packages from these groups to reduce the disk space required.

Table 2-2 Approximate Disk Space Required for Software Groups

If You Want to Install This Software Group[1]	**Which Installs**	**Then You'll Need Approximately This Much Disk Space**
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.	180 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.	270 Mbytes
Entire Distribution	The entire Solaris release (everything on the CD). Compilers and debuggers are not included.	360 Mbytes

[1] Another software group, entire distribution plus OEM support may also be available; disk space required will vary from release to release. To find out if this software group is available for your hardware platform, see the *SPARC Hardware Platform Guide.*

5 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 *File System Administration* for instructions.

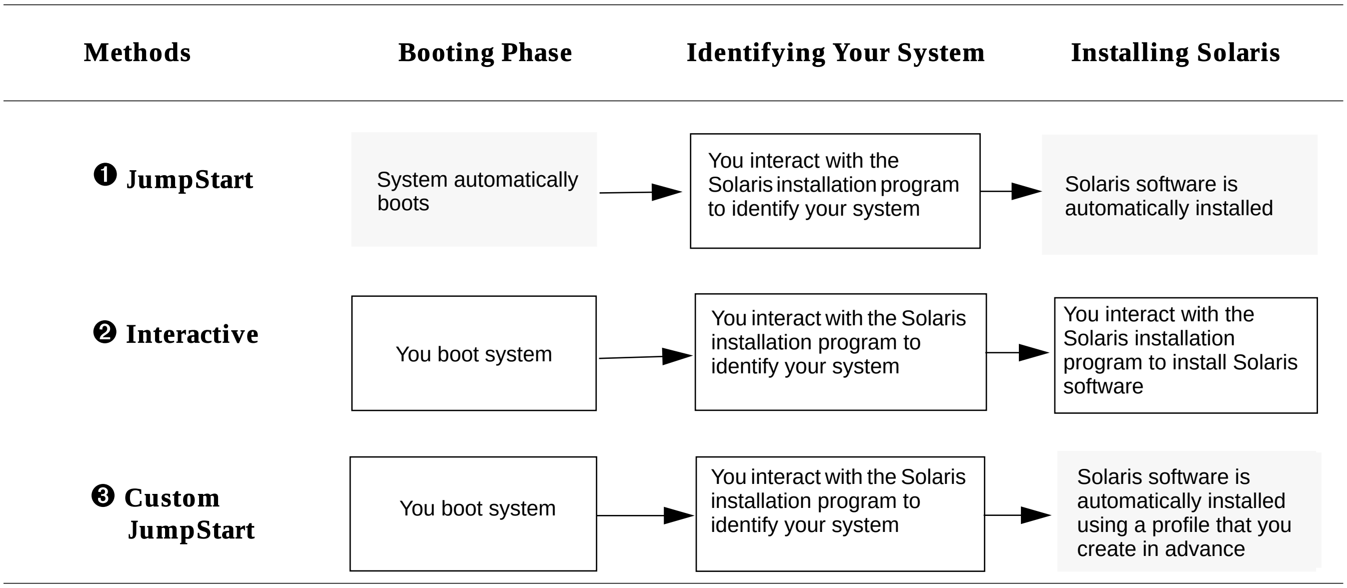
6 Choose a method for installing Solaris software.

There are three methods for installing Solaris software. One, two, or all three methods may be available to you.

❶ *JumpStart* - easiest method of installing Solaris software. The JumpStart software automatically installs a new system as a standalone (networked or non-networked) with Solaris software. However, not all new systems have the JumpStart software; see Step 1 on page 128 to find out if your system has JumpStart software.

❷ *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!

❸ *Custom JumpStart* (formerly called auto-install) - for the advanced user with experience in Bourne shell scripting and the `vi` editor. 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!



Methods	Booting Phase	Identifying Your System	Installing Solaris
❶ **JumpStart**	System automatically boots →	You interact with the Solaris installation program to identify your system →	Solaris software is automatically installed
❷ **Interactive**	You boot system →	You interact with the Solaris installation program to identify your system →	You interact with the Solaris installation program to install Solaris software
❸ **Custom JumpStart**	You boot system →	You interact with the Solaris installation program to identify your system →	Solaris software is automatically installed using a profile that you create in advance

7 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: JumpStart, 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. The box marked END points you to the chapter for step-by-step instructions for booting your system and installing Solaris software.

JumpStart Installations

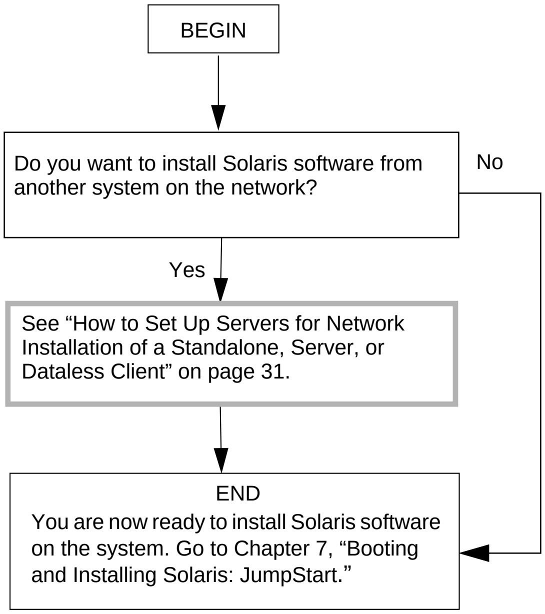

BEGIN
Do you want to install Solaris software from another system on the network?
No
Yes
See “How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client” on page 31.
END
You are now ready to install Solaris software on the system. Go to Chapter 7, “Booting and Installing Solaris: JumpStart.”

Figure 2-2 Task Map for JumpStart Installations

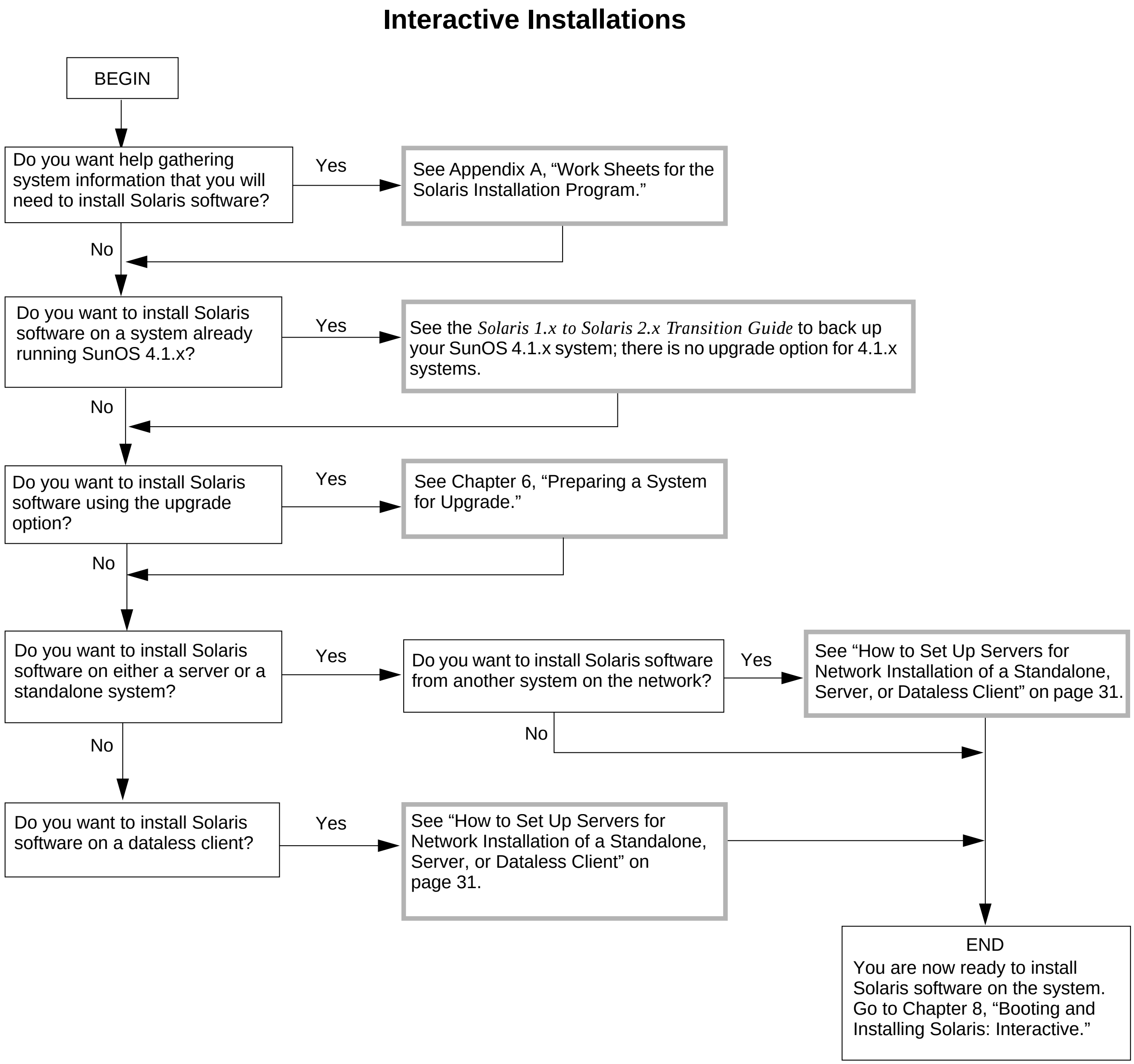

Interactive Installations
BEGIN
Do you want help gathering system information that you will need to install Solaris software?
Yes
See Appendix A, “Work Sheets for the Solaris Installation Program.”
No
Do you want to install Solaris software on a system already running SunOS 4.1.x?
Yes
See the *Solaris 1.x to Solaris 2.x Transition Guide* to back up your SunOS 4.1.x system; there is no upgrade option for 4.1.x systems.
No
Do you want to install Solaris software using the upgrade option?
Yes
See Chapter 6, “Preparing a System for Upgrade.”
No
Do you want to install Solaris software on either a server or a standalone system?
Yes
Do you want to install Solaris software from another system on the network?
Yes
See “How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client” on page 31.
No
No
Do you want to install Solaris software on a dataless client?
Yes
See “How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client” on page 31.
END
You are now ready to install Solaris software on the system. Go to Chapter 8, “Booting and Installing Solaris: Interactive.”

Figure 2-3 Task Map for Interactive Installations

Custom JumpStart Installations

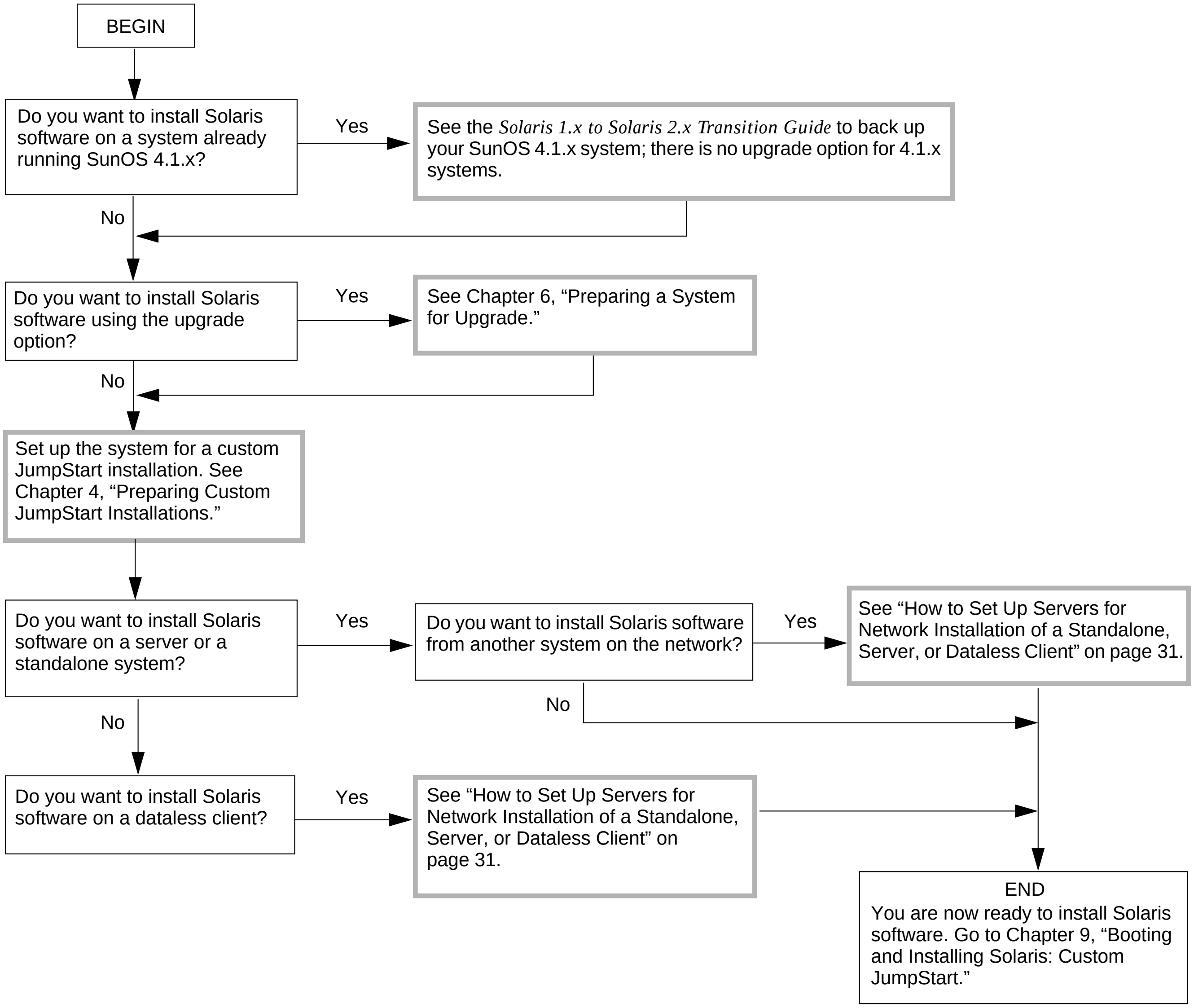

BEGIN
Do you want to install Solaris software on a system already running SunOS 4.1.x?
Yes
See the *Solaris 1.x to Solaris 2.x Transition Guide* to back up your SunOS 4.1.x system; there is no upgrade option for 4.1.x systems.
No
Do you want to install Solaris software using the upgrade option?
Yes
See Chapter 6, “Preparing a System for Upgrade.”
No
Set up the system for a custom JumpStart installation. See Chapter 4, “Preparing Custom JumpStart Installations.”
Do you want to install Solaris software on a server or a standalone system?
Yes
Do you want to install Solaris software from another system on the network?
Yes
See “How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client” on page 31.
No
No
Do you want to install Solaris software on a dataless client?
Yes
See “How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client” on page 31.
END
You are now ready to install Solaris software. Go to Chapter 9, “Booting and Installing Solaris: Custom JumpStart.”

Figure 2-4 Task Map for Custom JumpStart Installations

Preparing to Install Solaris Over a Network 3

How to Create an Install Server	*page 25*
How to Create a Boot Server on a Subnet	*page 28*
How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client	*page 31*
How to Set Up Servers for Network Installation of a Diskless Client	*page 41*
How to Preconfigure Network Information	*page 45*
How to Preconfigure Default Locale Using NIS	*page 48*
How to Preconfigure Default Locale Using NIS+	*page 51*

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.

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.

Diskless and dataless clients also require:

- *File server* – a system that provides files for other systems.

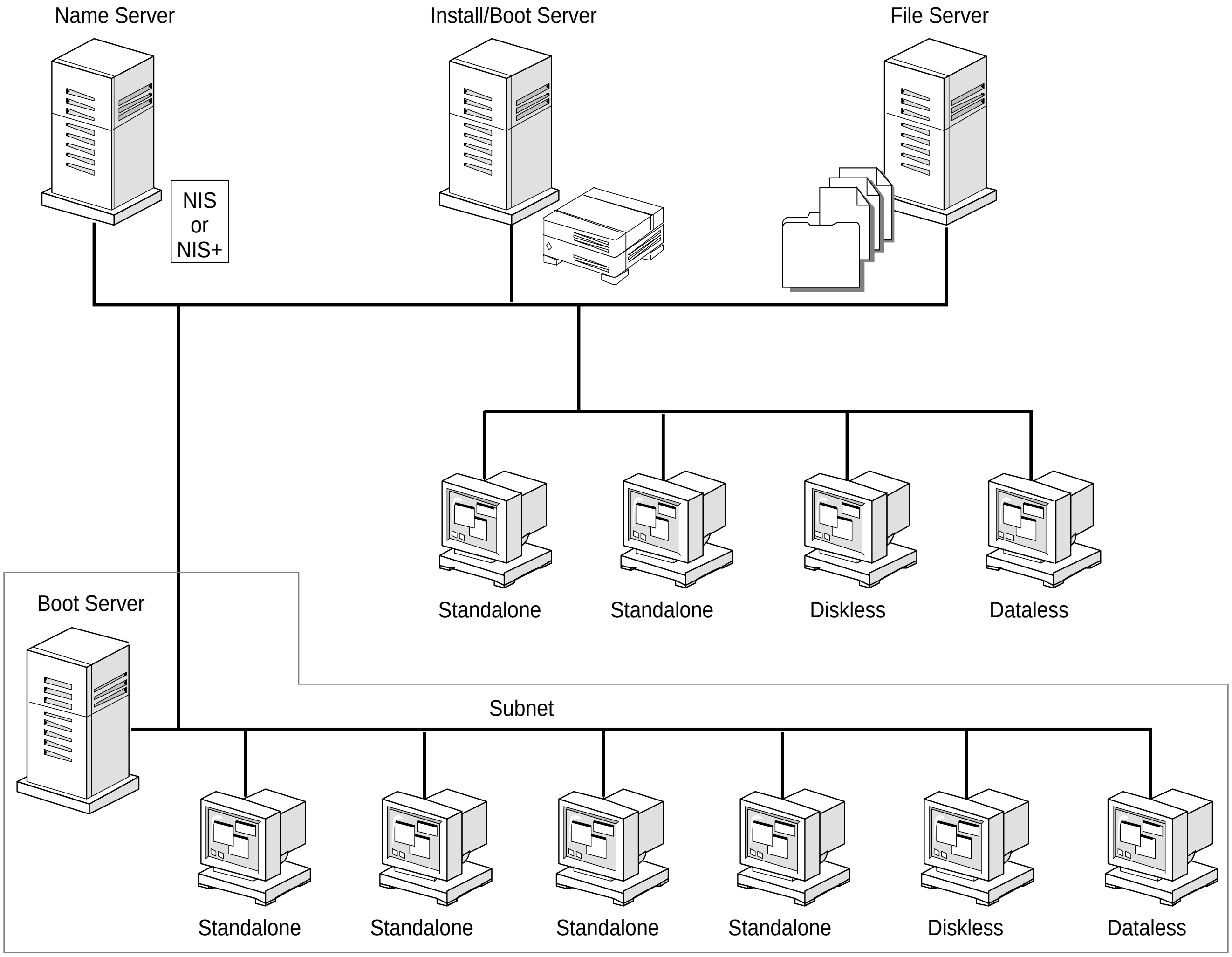


Figure 3-1 Network Installation Servers

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. (See “Preconfiguring Network Information” on page 45 for detailed information.) Doing so provides default values that the Solaris installation program can access, which eliminates many prompts to identify the system during an installation.

Files You Should Know About

Table 3-1 shows files that you may need to modify or monitor to set up a network installation.

Table 3-1 Network Installation Files

Files	Description
/cdrom/cdrom0/s0	File path of the mounted SPARC Solaris CD on a Solaris 2.x system. If more than one CD-ROM drive is attached to the system, the path may vary. (This mount point is created by Volume Management software.)
/cdrom/cdrom0/s2	File path of the mounted x86 Solaris CD on a Solaris 2.x system. If more than one CD-ROM drive is attached to the system, the path may vary. (This mount point is created by Volume Management software.)
/etc/hosts /etc/ethers /etc/bootparams	Files containing information necessary for exchanging information between systems on a network.
/var/nis/*host*/hosts.org_dir /var/nis/*host*/ethers.org_dir /var/nis/*host*/bootparams.org_dir	The NIS+ tables that correspond to the /etc files. These files are only pertinent if the system is using the NIS+ naming service. *host* is the host name of the NIS+ server.
/etc/nsswitch.conf	A file that you can edit to specify where (in the /etc files or in NIS+ tables) system software should look to find configuration information.
/etc/inet/inetd.conf	A configuration file that has information about daemons that may be required for network installation.
/etc/dfs/dfstab	A configuration file that specifies file systems that will be shared across the network.
/etc/mnttab	A configuration file that specifies file systems that are mounted.

Commands You Should Know About

Table 3-2 shows programs available when you set up for network installation.

Table 3-2 Network Installation Commands

Program	Description
`add_install_client`	A script that adds information to the `/etc` files on the install server to enable network installation. See the `add_install_client(1m)` man page for more information.
`rm_install_client`	A script that removes information from the `/etc` files on the install server to disable network installation. See the `rm_install_client(1m)` man page for more information.
`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 Administration Tool (`admintool`) program. 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.
`ttyhstmgr`	A text-based equivalent of the Administration Tool Host Manager. See the `ttyhstmgr(1m)` man page for more information.
`mount`	A command that shows mounted file systems, including the Solaris CD file system. See the `mount(1m)` man page for more information.

Table 3-2 Network Installation Commands *(Continued)*

Program	Description
`uname -a`	A command for determining system architecture. This information is sometimes required during installation. See the `uname(1)` man page for more information.
`reset`	A command for resetting the terminal settings and display. It is sometimes useful to use `reset` before booting. Or, if you boot and see a series of error messages about I/O interrupts, press the L1 or STOP and A keys at the same time, and then enter `reset` at the `ok` or `>` PROM prompt.
`banner`	A command for displaying system information, such as model name, Ethernet address, or memory installed. Available only from the `ok` or `>` PROM prompt.

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:

- Server
- Standalone
- Dataless

This section describes how to:

- 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 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 as 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. **Follow the instructions depending 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 -a` 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 27.
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 28. 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 27.

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 the appropriate section in this chapter:

- “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 architecture 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 as 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. **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 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 specified kernel architecture information to the local disk. Enter this command once for each type of system to be installed over the network.

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

In this command,

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

boot_dir_path Specifies the directory where the kernel architecture will be copied. You can substitute any directory path.

architecture Specifies the kernel architecture of the system to be installed. Sample valid values include sun4e, sun4d, sun4c, sun4m, and i86pc. (For a detailed list of kernel architectures, see Appendix C, "Kernel Architectures.")

For example, the following command copies kernel architecture information necessary for booting SPARC sun4c systems to be installed over the network:

```
./setup_install_server -b /export/install sun4c
```

Note – The `setup_install_server` command will indicate if you do not have enough disk space to copy the kernel architecture information. Use the `df -a` command to determine available disk space.

Task Complete

The boot server is now set up to boot systems on a subnet. To continue, go to the appropriate section in this chapter:

- "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
Server	Install Server
Dataless Client	Install Server and File Server
Diskless Client	File Server

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

Note – When you install a SPARC server, you don't have the option to support x86 clients. If you want your SPARC server to serve x86 dataless or diskless clients, you must use Software Manager (`swmtool`) and add services for the x86 architecture *after* the server has been installed. To do this, select Add Client Support from the `swmtool` Edit menu.

Choose from the following procedures, based on the type of systems you are going to install Solaris software on:

- "How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client" on page 31
- "How to Set Up Servers for Network Installation of a Diskless Client" on page 41

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

The procedure to add standalone systems and servers for installation over the network is the same. The procedure to add dataless clients involves the added step of specifying a file server. This section describes both procedures.

Overview – This procedure involves:

- Starting the Administration Tool Host Manager to update the name service and (if necessary) set up a file server
- Using the `add_install_client` command to add information about the standalone, server, or dataless clients to the install server configuration files

---

Note – If you have a character-based terminal, you can use `ttyhstmgr` to add systems for network installation. To start `ttyhstmgr`, type `ttyhstmgr` and press Return.

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

1. Follow the instructions in the following table.

If the System to Be Installed Is	And the System Is	Then
Not in the NIS or NIS+ name service	Standalone, server, or dataless client	Go to Step 2.
In the NIS or NIS+ name service	Dataless client	Go to Step 2.
In the NIS or NIS+ name service	Standalone or server	Go to Step 11.

2. **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 Administration Tool 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.

3. **Start Administration Tool with the following command.**

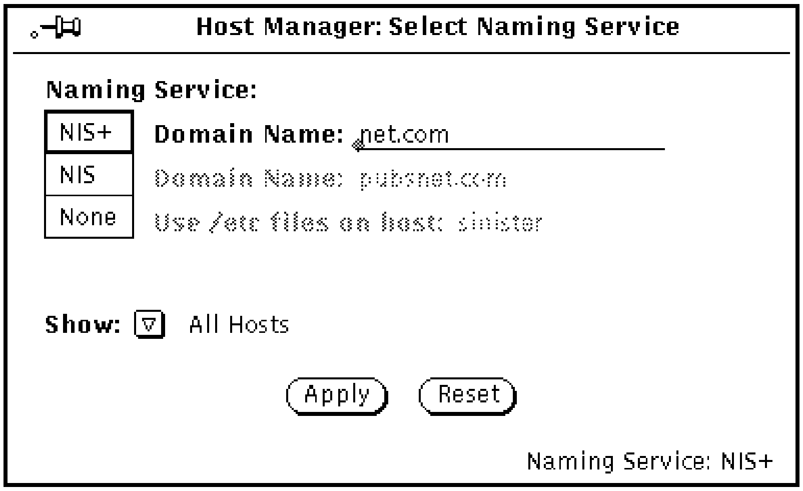
```
$ /usr/bin/admintool &
```

4. **After the Administration Tool 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 *Name Services Administration Guide.*

5. **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	None. Host Manager will store the information you provide in `/etc` files. You will need to update NIS maps manually.
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.



Host Manager: Select Naming Service

Naming Service:

NIS+ | Domain Name: net.com

NIS | Domain Name: pubsnet.com

None | Use /etc files on host: sinister

Show: ▽ All Hosts

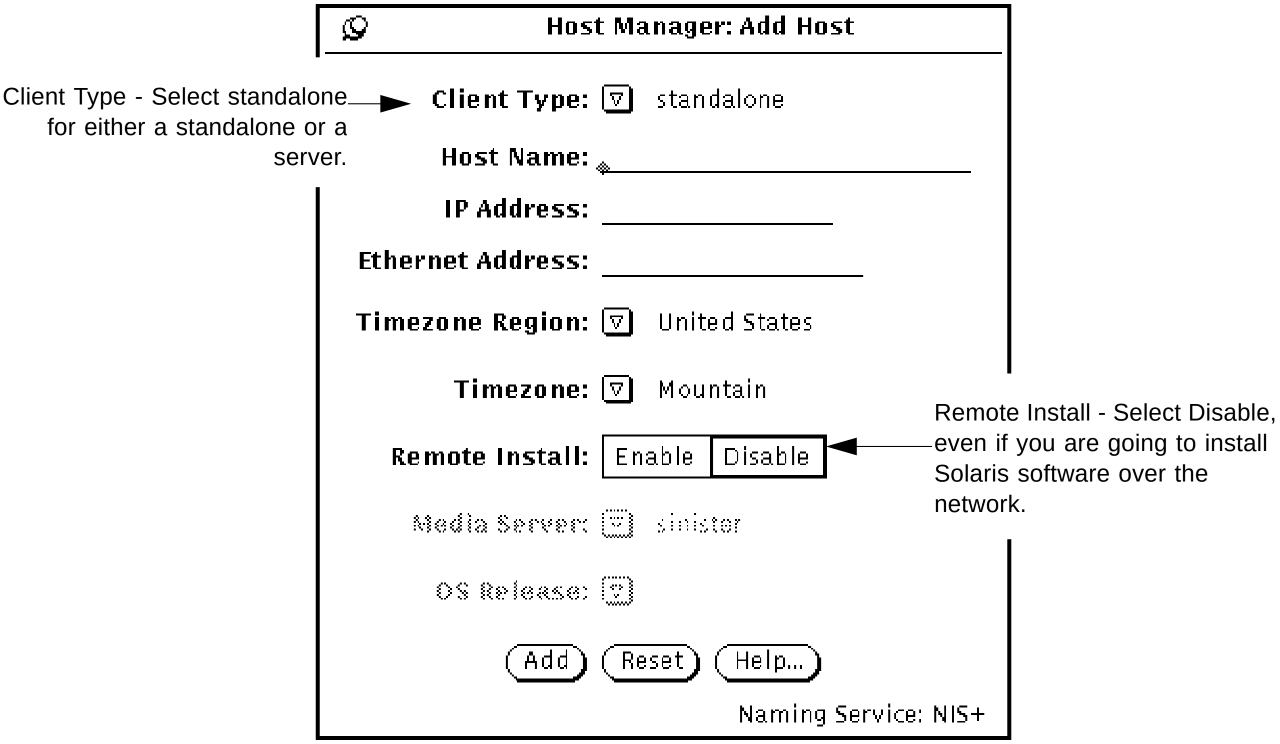
Naming Service: NIS+

6. **On the Host Manager screen, choose Add Host from the Edit menu command.**

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

System To Be Installed Is	Then
Standalone or Server	Go to Step 8.
Dataless Client	Go to Step 10.

8. **On the Host Manager: Add Host screen, complete all fields and click on the Add button.**



Host Manager: Add Host

Client Type: standalone

Host Name:

IP Address:

Ethernet Address:

Timezone Region: United States

Timezone: Mountain

Remote Install: Enable [✓ Disable]

Media Server: sinister

OS Release:

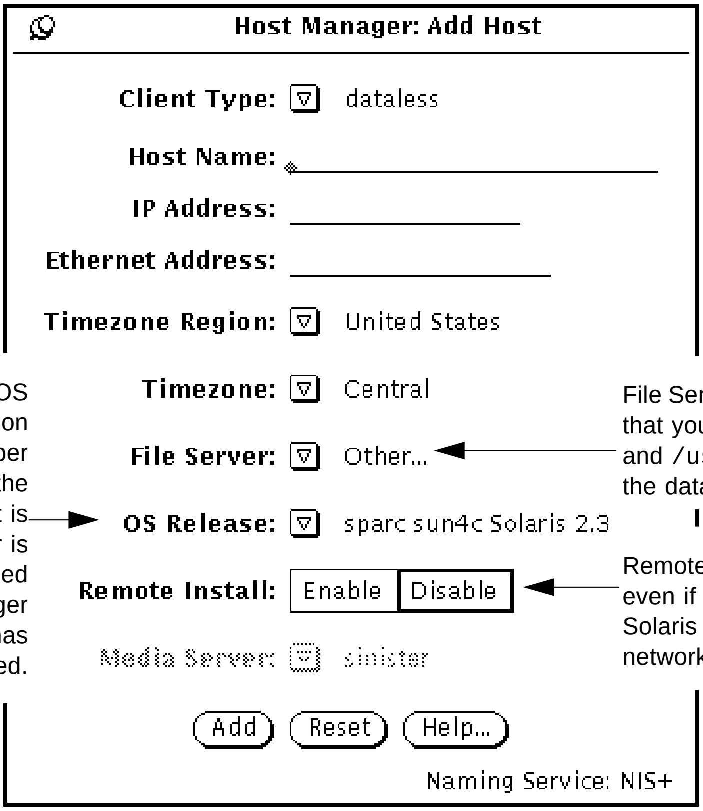
Naming Service: NIS+

Client Type - Select standalone for either a standalone or a server.

Remote Install - Select Disable, even if you are going to install Solaris software over the network.

9. **Go to Step 11.**
   Step 10 is required only for adding dataless clients for network installation.

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



Host Manager: Add Host

Client Type: dataless

Host Name:

IP Address:

Ethernet Address:

Timezone Region: United States

Timezone: Central

File Server: Other...

OS Release: sparc sun4c Solaris 2.3

Remote Install: Enable Disable

Media Server: sinister

Naming Service: NIS+

OS Release - Select the OS Release you want installed on the dataless client. The proper OS support must reside on the install server. This support is normally set up when a server is installed. It can also be specified with Software Manager (`swmtool`) after the server has been installed.

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

Remote Install - Select Disable, even if you are going to install Solaris software over the network.

11. **Make sure the install server has been set up.**
    See "How to Create an Install Server" on page 25 for instructions.

12. **Determine your next step based on whether the standalone system or server is on the same subnet as the install server.**

If Standalone System or Server Is	Then
On same subnet as install server	Go to Step 13.
Not on same subnet as install server	Go to Step 16.

13. **On the install server, change the directory to the mounted Solaris CD or the directory where the Solaris CD image has been copied.**

14. **Use the `add_install_client` command to add information about the system to be installed to the install server's configuration files.**

```
# ./add_install_client [-c server:jumpstart_dir_path] host_name  architecture
```

In this command,

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

Note: This option is unnecessary if you have completed the procedure "How to Enable All Systems to Access the JumpStart Directory" on page 74.

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.

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.

architecture Is the kernel architecture of the system to be installed. Sample valid values include sun4e, sun4d, sun4c, sun4m, and i86pc. (For a detailed list of kernel architectures, See Appendix C, "Kernel Architectures.")

15. **Go to Task Complete on page 40.**
    Step 16 through Step 18 are required only if you are using a boot server.

16. **On the boot server, log in as root.**

17. **Change the directory to the boot directory.**
    The boot directory is created when you set up a system as a boot server, as described in "How to Create a Boot Server on a Subnet" on page 28.

```
# cd /boot_dir_path
```

In this command,

/boot_dir_path Is the absolute path to the directory that contains the kernel architecture information necessary for booting a system.

For example, the following command changes to a sample boot directory named `boot_dir`.

```
cd /boot_dir
```

18. 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 architecture
```

In this command,

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

Note: This option is unnecessary if you have completed the procedure “How to Enable All Systems to Access the JumpStart Directory” on page 74.

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.

architecture Is the kernel architecture of the system to be installed. Sample valid values include sun4e, sun4d, sun4c, sun4m, and i86pc. (For a detailed list of kernel architectures, See Appendix C, “Kernel Architectures.”)

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 `/tftpboot` directory on the local system, which will be the boot server. The system that will be installed is named `basil`, and it is a SPARCstation™ IPC™, which has a sun4c kernel architecture:

```
./add_install_client -s install_server1:/export/install basil sun4c
```

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 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: JumpStart"
- Chapter 8, "Booting and Installing Solaris: Interactive"
- Chapter 9, "Booting and Installing Solaris: Custom JumpStart"

▼ How to Set Up Servers for Network Installation of a Diskless Client

Technically speaking, diskless clients do not get Solaris software installed over the network. They just receive file services from a file server. However, diskless clients do require boot services of a server on the network. The following procedure describes how to specify:

- Where the diskless client will get its files
- Which server will boot the diskless client

Overview – The procedure to add diskless clients for installation involves:

- Logging in to the install server
- Using the Administration Tool Host Manager to add information about the diskless client to the install server configuration files

---

Note – If you have a character-based terminal, you can use the `ttyhstmgr` to add systems for network installation. To start `ttyhstmgr`, type `ttyhstmgr` and press Return.

Follow this procedure to add a diskless client to the install server.

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 Administration Tool 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.

---

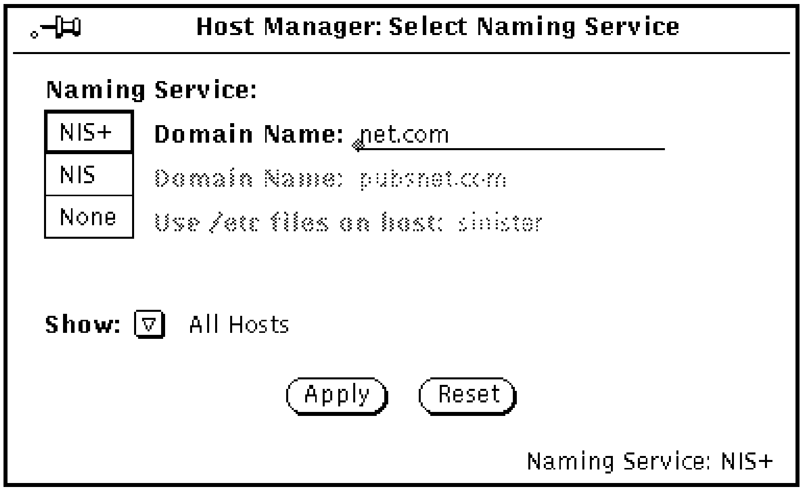
2. **Start Administration Tool with the following command.**

```
$ /usr/bin/admintool &
```

3. **After the Administration Tool main window appears, click on the Host Manager icon.**

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

If the Name Service Is	Then Select
NIS	None. Host Manager will store the information you provide in `/etc` files. You will need to update NIS maps manually.
NIS+	NIS+. Host Manager will update the NIS+ tables.
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.



Domain Name: net.com

Domain Name: pubsnet.com

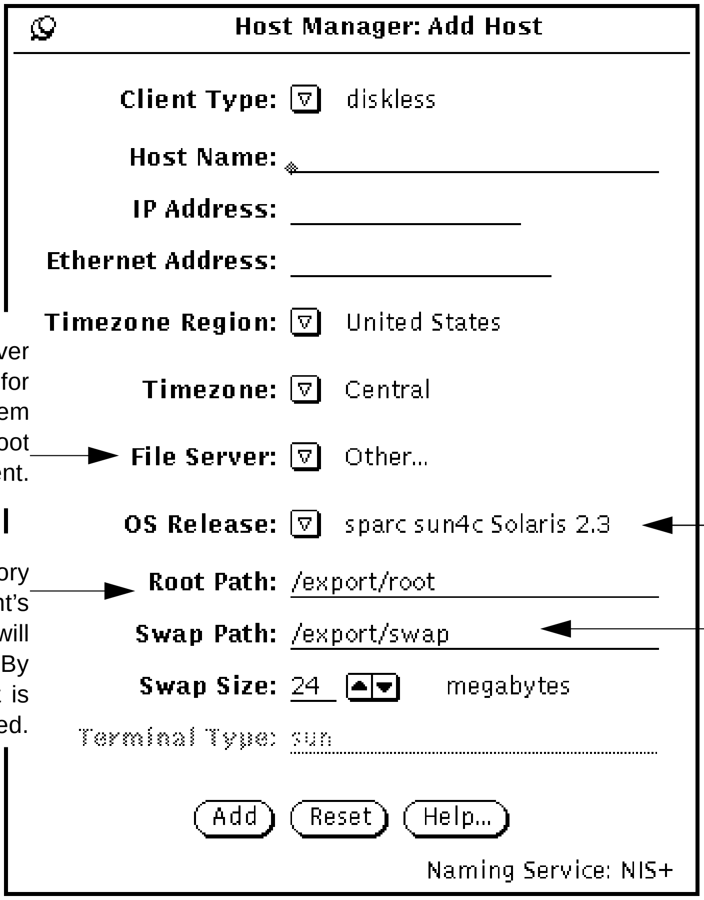
Use /etc files on host: sinister

Show: All Hosts

Naming Service: NIS+

5. **On the Host Manager screen, choose Add Host from the Edit menu command.**

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



Host Manager: Add Host

Client Type: diskless

Host Name:

IP Address:

Ethernet Address:

Timezone Region: United States

Timezone: Central

File Server: Other...

OS Release: sparc sun4c Solaris 2.3

Root Path: /export/root

Swap Path: /export/swap

Swap Size: 24 megabytes

Terminal Type: sun

Naming Service: NIS+

File Server – Specify the server that will provide the software for the diskless client. This system you specify will also be the boot server for the diskless client.

Root Path – Specify a directory where the diskless client's individual root file system will reside on the file server. By default, `/export/root` is specified.

OS Release – If the server supports multiple client architectures, select the correct one for the diskless client.

Swap Path – Specify a directory where the diskless client's individual swap space will reside on the file server. By default, `/export/swap` is specified, but you can specify any directory where there is space available.

7. On the diskless system, type the appropriate boot command from the PROM `ok` prompt. Choose from the following table:

If Diskless System Type Is	Then Boot Using
Sun-4™/3*xx*	`b le()`
Sun-4/110 Sun-4/2*xx* Sun-4/4*xx*	`b ie()`
All other Sun® systems	`boot net`

Caution – You may need to verify that the bootparams search path is correct in the `/etc/nsswitch.conf` file. If the search path is incorrect, the diskless client will report "timeout waiting for ARP/RARP" messages and the boot will fail. See the `nsswitch.conf(4)` man page for detailed information.

Task Complete

The diskless system is now set up to boot over the network. After you turn on the diskless system, it will boot over the network and the proper file systems will be mounted from the specified file server.

Preconfiguring Network Information

Preconfiguring network information provides default values for the following network-wide information:

- Date and time
- Geographic region
- Site subnet mask

Setting up network default values eliminates the need to supply this information after the system boots. After you set up this information in the name service, the Solaris installation program finds it automatically and does not prompt the user for it.

You can also preconfigure the language (called the *locale*) provided by Solaris software. See "Preconfiguring the Default Locale" on page 48 for detailed information.

Note – To preconfigure a system's host name and Ethernet address, use the Administration Tool's Host Manager as described in "How to Set Up Servers for Network Installation of a Standalone, Server, or Dataless Client" on page 31.

▼ How to Preconfigure Network Information

Overview – The procedure to preconfigure network information involves:

- Using the `vi` text editor to edit files in the `/etc` directory
- Updating the name service (NIS or NIS+) files with the information you've added to the files in the `/etc` directory

Follow this procedure to preconfigure the date and time, the geographic region, and the subnet mask used at your site. (You can do Step 2, Step 3, or Step 4, one or all, as long as you complete Step 5.)

1. **On the NIS or NIS+ name server, log in as root.**

2. **Edit the** `/etc/hosts` **file.**
   In the `/etc/hosts` file, choose an entry for the server that will supply the date and time for systems being installed over the network. At the end of this entry, add the `timehost` alias. Use the following syntax:

```
IP_address      host_name  timehost
```

Note – The `/etc/hosts` file has read-only permissions. Using the `vi` editor, you must exit the file using `:wq!` to force the file to be saved with your changes.

3. **In the** `/etc/timezone` **file, specify the domain that will supply the geographic region for systems being installed over the network.**
   To do this, add a valid time zone and the network domain name. Use the following syntax:

```
timezone  domain_name
```

   For example, the following entry in the `/etc/timezone` file configures all systems in the xx.yy.zz.COM domain into the U.S. Central time zone:

```
US/Central xx.yy.zz.COM
```

   For information on valid time zone names, see the `/usr/share/lib/zoneinfo` directory.

Note – The Administration Tool Host Manager sets the geographic region for each system rather than for all systems within a network domain.

4. **In the** `/etc/netmasks` **file, specify the subnet mask to be used by systems being installed over the network.**
   To do this, add the network number and the corresponding netmask. Use the following syntax:

```
network_address  subnet_mask
```

   For example, the following entry in the `/etc/netmasks` file sets the subnet mask for the network address 128.32.0.0:

```
128.32.0.0 255.255.255.0
```

5. **Update the NIS or NIS+ tables with the information you added to the** `/etc/hosts`**,** `/etc/timezone`**, and** `/etc/netmasks` **files.**

If Your Site Uses	Then
NIS	`/var/yp/make`
NIS+	`/usr/lib/nis/nispopulate -F -p /etc hosts timezone netmasks`

Task Complete

The date and time, geographic region, and netmask are now configured in the network. During a network installation of a system, the Solaris installation program will automatically use the default values you have configured.

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` 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 as 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 52 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.**

```
# /var/yp/make
```

Task Complete

This completes the procedure to set up a default locale for systems running the NIS name server. 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

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 *Name Services 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**
English (Solaris default)	`C`
French	`fr`
German	`de`
Italian	`it`
Latin America	`es`
Swedish	`sw`

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

This completes the procedure to set up a default locale for systems running the NIS+ name server. The default language you have specified will be used during the installation and will also be the language provided to system users.

Preparing Custom JumpStart Installations 4

How to Create a JumpStart Directory on a Diskette for x86 Systems	*page 61*
How to Create a JumpStart Directory on a Diskette for SPARC Systems	*page 67*
How to Create a JumpStart Directory on a Server	*page 71*
How to Enable All Systems to Access the JumpStart Directory	*page 74*
How to Create a Profile	*page 77*
How to Create the rules File	*page 93*
How to Use check to Validate the rules File	*page 104*

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.

Reasons to Choose a Custom JumpStart Installation

You should choose custom JumpStart installations when:

- You have to install the Solaris software on many systems.
- You have to install the Solaris software differently on particular groups of systems.

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

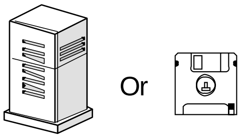
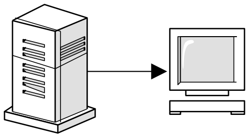
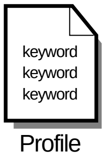
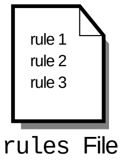
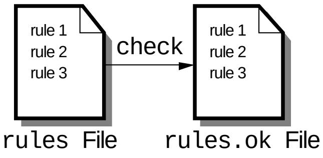
- You need to install the Solaris software on 100 new systems.
- The engineering group owns 70 out of the 100 new systems, and its systems must be installed as standalone systems with the developer software group.
- The marketing group owns 30 out of the 100 new systems, and its systems must be installed as dataless 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 61. If you are going to use a server for custom JumpStart installations, see “Creating a JumpStart Directory on a Server” on page 71.
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 74 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 76 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 92 for detailed information.
Using `check` to validate the `rules` file	 rule 1 rule 2 rule 3 check rule 1 rule 2 rule 3 rules File 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* use the `check` script to create the `rules.ok` file. See “Using check to Validate the rules File” on page 103 for detailed information.

What Happens During a Custom JumpStart Installation

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

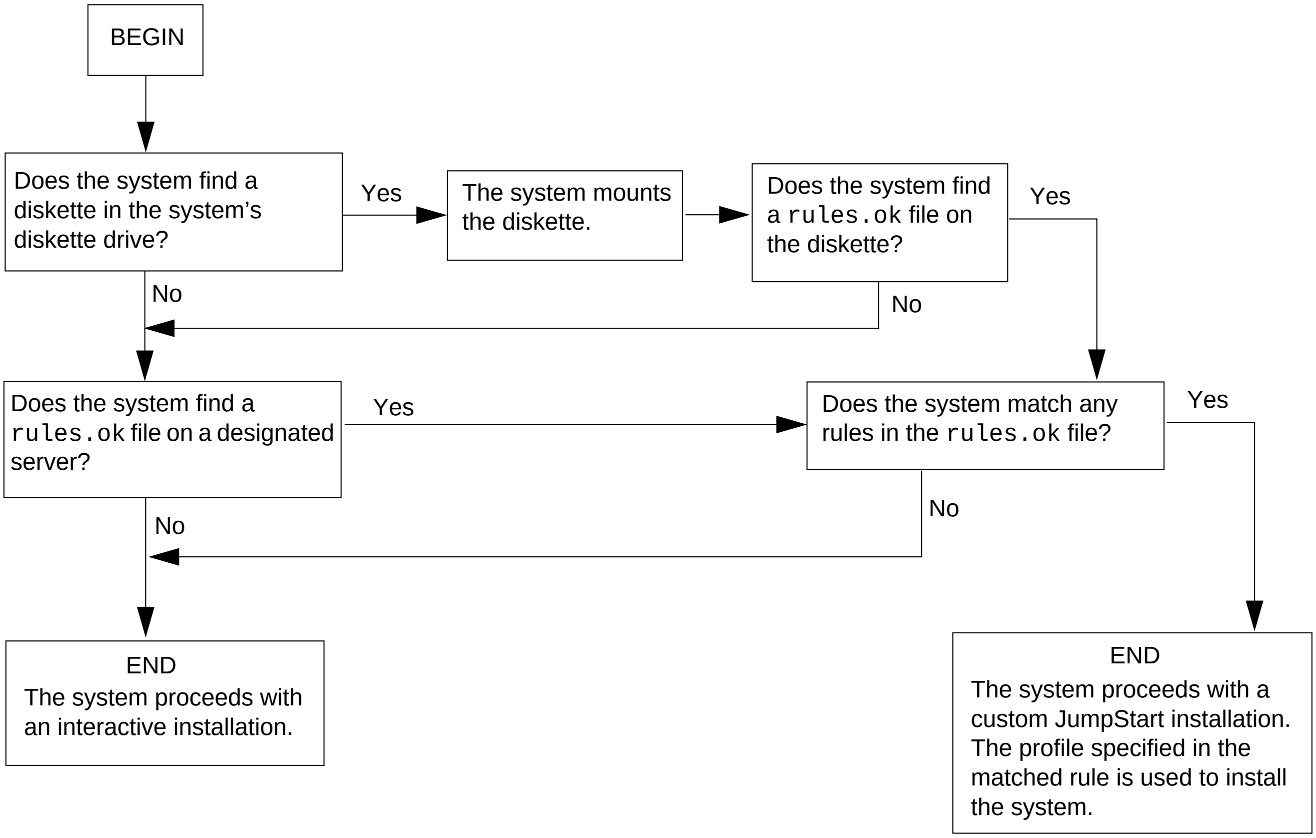
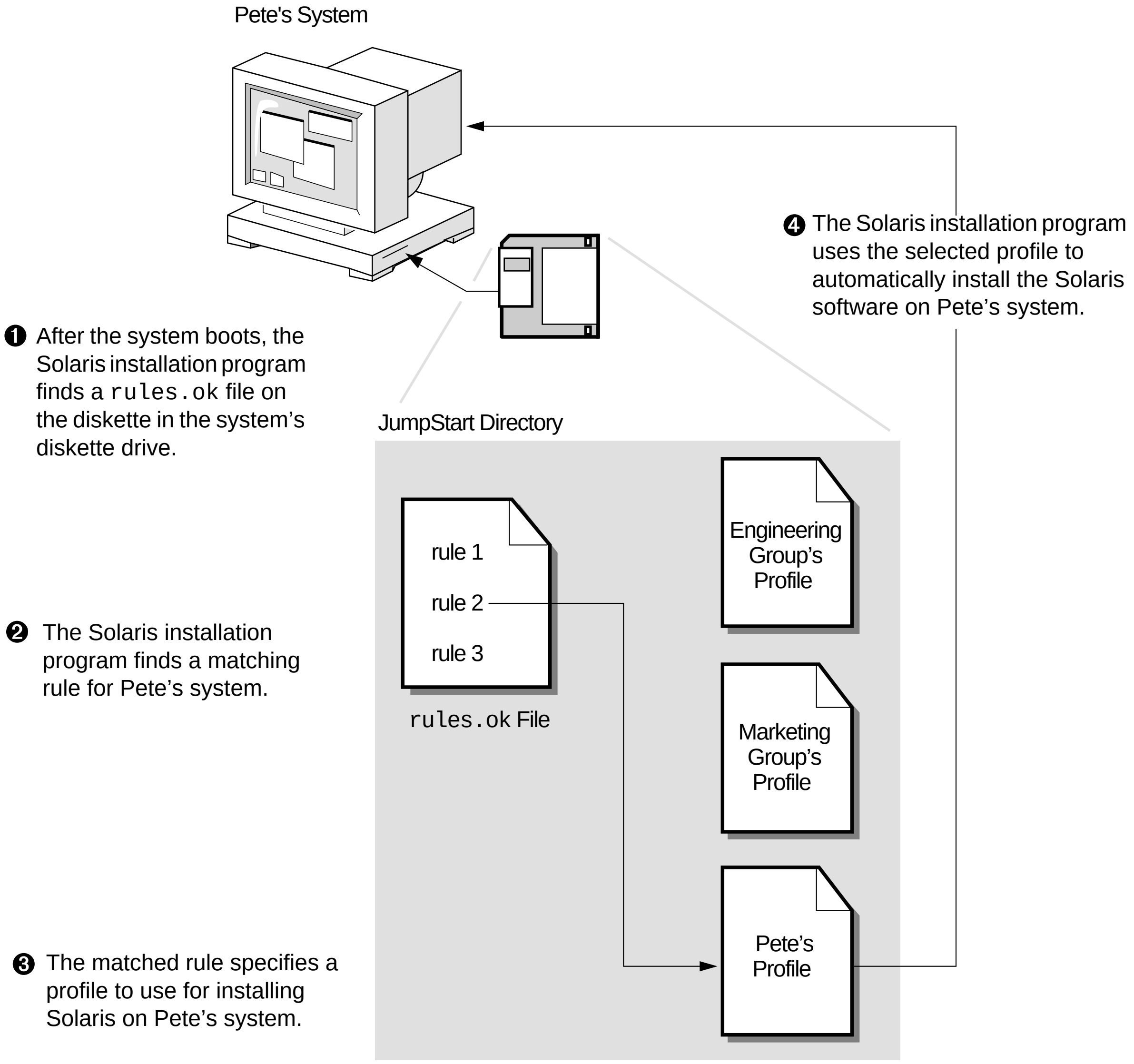

BEGIN
Does the system find a diskette in the system's diskette drive?
Yes
No
The system mounts the diskette.
Does the system find a rules.ok file on the diskette?
Yes
No
Does the system find a rules.ok file on a designated server?
Yes
No
Does the system match any rules in the rules.ok file?
Yes
No
END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

Figure 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.



❶ After the system boots, the Solaris installation program finds a `rules.ok` file on the diskette in the system's diskette drive.

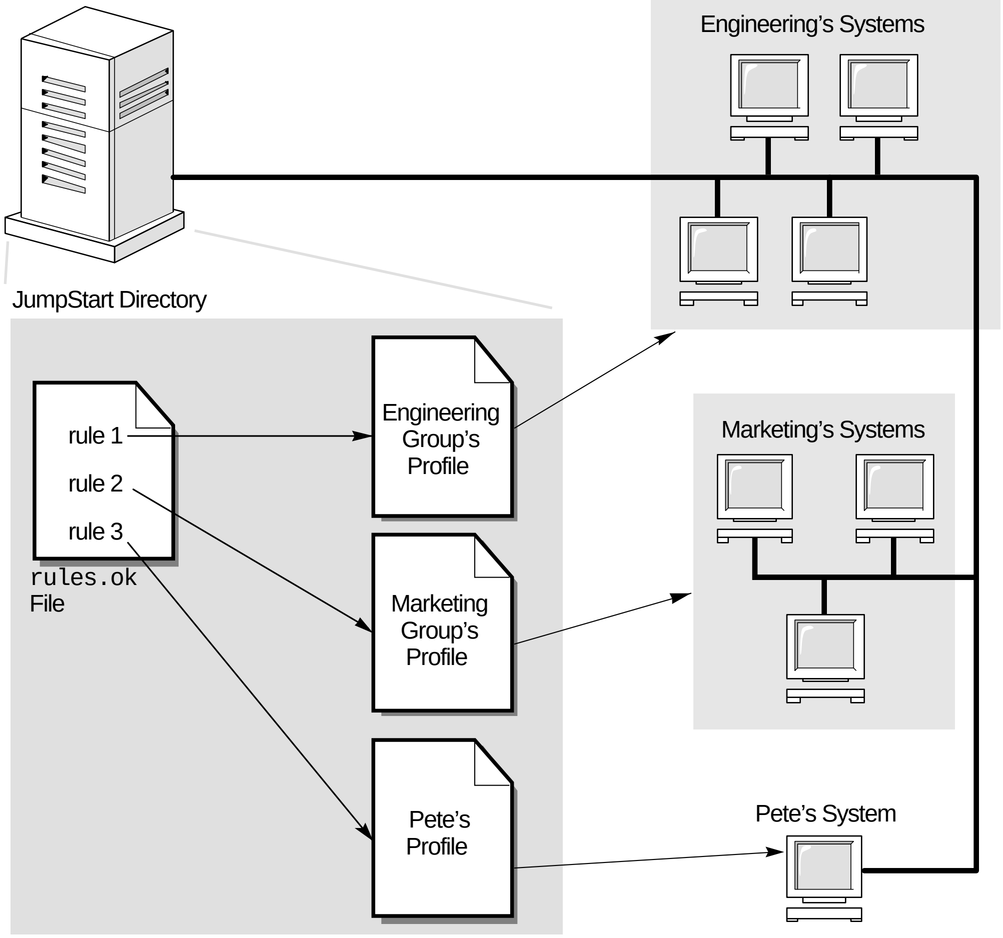
❷ The Solaris installation program finds a matching rule for Pete's system.

❸ The matched rule specifies a profile to use for installing Solaris on Pete's system.

❹ The Solaris installation program uses the selected profile to automatically install the Solaris software on Pete's system.

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.



❶ After a system boots, the Solaris installation program finds a `rules.ok` file on designated server.

❷ Solaris installation program finds a matching rule for the specific system.

❸ The matched rule specifies a profile to use for installing the Solaris software on a that system.

❹ The Solaris installation program uses the selected profile to automatically install the Solaris software on the system.

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. So, the following pages describe how to create a JumpStart directory on a diskette for an x86 system and for a SPARC system.

▼ 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 77.

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

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

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

Follow this procedure to create a JumpStart directory on a diskette for SPARC Systems.

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

2. **Insert a 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. **If the diskette already has a UFS file system on it, go to Step 16.**
   To find out if the diskette has a UFS file system on it, check the `/etc/mnttab` file for an entry similar to this:

```
/floppy/unnamed_floppy    ufs
```

6. **Format the diskette:**

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

```
# fdformat -U
```

7. **Create a UFS file system on the diskette:**

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

8. **Eject the diskette:**

```
# eject floppy
```

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

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

```
# volcheck
```

11. **Go to Step 16.**
    Step 12 through Step 15 are used if the system is not running Volume Management.

12. **If the diskette already has a UFS file system on it, go to Step 15.**
    If the `mount` command fails in Step 15, the diskette does not have a UFS file system on it.

13. **Format the diskette:**

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

```
# fdformat /dev/rdiskette
```

14. Create a file system on the diskette:

```
# newfs /dev/rdiskette
```

15. Mount the diskette:

```
# mount -F ufs /dev/diskette jumpstart_dir_path
```

In this command,

jumpstart_dir_path Is the absolute directory path where the diskette is mounted.

For example, the following command would mount a SPARC diskette on the `/mnt` directory:

```
mount -F ufs /dev/rdiskette /mnt
```

Note – If the `mount` command fails, go back to Step 13 to format the diskette.

16. 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 17.
Solaris CD image on the local disk	**1)** Change the directory to the Solaris CD image on the local disk. For example: `cd /export/install` **2)** Go to Step 18.

17. **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`

18. **Copy the custom 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.

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 77.

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 77.

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
```

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 75.

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 61 or page 71, 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 78 provides some examples of profiles.
   - Table 4-2 on page 81 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 114 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 93.

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 81 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.

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 kernel architecture than it uses. If you do not specify `client_arch`, any diskless client must have the same architecture as the server. You must specify `client_arch` once for each client architecture. Valid values for *karch_value* are sun4, sun4e, sun4d, sun4c, sun4m, or i86pc. (See Appendix C, "Kernel Architectures" for a detailed list of the kernel architectures of the various systems.) **Restriction:** `client_arch` 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 (SPARC only).**

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*. Use Software Manager on an installed system to view detailed information about clusters and their names. **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 (SPARC only).**

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 102). • `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 (SPARC only).**

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

size - Choose one of the following:

- *DDD* - An fdisk partition of size *DDD* (in Mbytes) is created on the specified disk. *DDD* must be 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 (SPARC only).**

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 and /usr/kvm file systems. The following lines are 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`
	`filesys sherlock:/export/exec/kvm/Solaris_2.4_sparc.sun4c/usr/kvm - /usr/kvm`
	server: - The name of the server where the remote file system resides (followed by a colon).
	path - The remote file system's mount point name, for example, /usr or /export/home.
	server_address - The IP address of the server specified in *server:path*. If you don't have a name service running on the network, this value can be used to populate the /etc/hosts file with the server's host name and IP address. If you don't want to specify the server's IP address (if you have a name service running on the network), you must specify a minus sign (-).
	mount_pt_name - The name of the mount point that the remote file system will be mounted on.
	mount_options - One or more mount options (-o option of the mount(1M) command) that are added to the /etc/vfstab entry for the specified *mount_pt_name*.
	Note: If you need to specify more than one mount option, the mount options must be separated by commas and no spaces. For example: ro,quota

† **Profile keywords that can be used for upgrading (SPARC only).**

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 (SPARC only).**

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 (SPARC only) on the system. **Restriction:** `install_type` must be the first profile keyword in every profile.

† **Profile keywords that can be used for upgrading (SPARC only).**

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 — `sw` 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 (SPARC only).**

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 Software Manager 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 (SPARC only).**

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 (SPARC only).**

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 kernel architecture 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 61 or page 71, 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 98). 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 96 provides some examples of rules.
   - Table 4-5 on page 98 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 kernel architecture (`karch`), 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 98 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 98 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.
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 108 for detailed information on how to create begin scripts.

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

Field	Description
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 121 and “Creating Derived Profiles With Begin Scripts” on page 108.
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 110 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 104.

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 installation program matches a rule keyword and value with a known system, it installs the Solaris software specified by the profile listed in the profile field.

```
   # rule keywords and rule values             begin script   profile      finish script
   # -----------------------------             ------------   -------      -------------
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 i86pc                                     setup      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.

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. This rule also provides an example of rule wrap, which is defined on page 97.

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 allow 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.

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`	*arch_value*		Matches a system's architecture type.
	Architecture	*arch_value*	If you have a system already installed, the `arch` command or the `uname -p` command reports the system's architecture type.
	SPARC	sparc	
	x86	i386	
`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 102. *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`.

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 102. *version* - A version name, such as `Solaris_2.2`, 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`	*karch_value* Valid values are sun4, sun4e, sun4d, sun4c, sun4m, or i86pc. (See Appendix C, "Kernel Architectures" for a detailed list of the kernel architectures of the various systems.)	Matches a system's kernel architecture type. If you have a system already installed, the `arch -k` command or the `uname -m` command reports the system's kernel architecture type.
`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,4-80	
	SPARCstation Voyager	SUNW,S240	
	x86	i86pc	
`network`	*network_num*		Matches a system's network number, which the 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).

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

Rule Keyword	Rule Values	Description
`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]
```

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 2.4.

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

- 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 9, “Booting and Installing Solaris: Custom JumpStart.”

Using Optional Custom JumpStart Features 5

How to Use pfinstall to Test a Profile	*page 115*
How to Create a Disk Configuration File for a SPARC System	*page 117*
How to Create a Multiple Disk Configuration File for a SPARC System	*page 119*

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 actually 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 9, "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
```

▼ How to Create a Disk Configuration File for a SPARC System

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.

Overview – The procedure to create a disk configuration file for a SPARC system involves:

- Locating a SPARC system with a disk that you want to test a profile against.
- Using the `prtvtoc(1M)` command to create the disk configuration file.

Follow this procedure to create a disk configuration file.

1. **Locate a 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 `prtvtoc` to create the disk configuration file:**

```
$ prtvtoc /dev/rdsk/device_name > disk_config
```

In this command,

`/dev/rdsk/`*device_name*

Is the device name of the system's disk. *device_name* must be in the form `c`*w*`t`*x*`d`*y*`s2` or `c`*x*`d`*y*`s2`.

Note: Slice 2 must be specified in *device_name.*

disk_config Is the disk configuration file name.

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

```
$ cp disk_config jumpstart_dir_path
```

Task Complete

You have completed creating a disk configuration file. The following page provides an example of creating a disk configuration file.

The following example creates a disk configuration file, `104_test`, on a system with a 104-Mbyte disk, whose device name is c0t3d0s2.

```
$ prtvtoc /dev/rdsk/c0t3d0s2 > 104_test
```

In this example, the `104_test` file contains the following information:

```
# cat 104_test
* /dev/rdsk/c0t3d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*      35 sectors/track
*       6 tracks/cylinder
*     210 sectors/cylinder
*    1019 cylinders
*     974 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags   Sector     Count    Sector  Mount Directory
       0      2    00          0     16170     16169
       1      3    00      16170     28140     44309
       2      5    00          0    204540    204539
       6      4    01      44310    160230    204539
```

▼ How to Create a Multiple Disk Configuration File for a SPARC System

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

Overview – The procedure to create a multiple disk configuration file for a SPARC system involves:

- Concatenating two or more disk configuration files into one file.
- Changing the target numbers of the disks (if needed).

The following procedure creates a disk configuration file to test a profile on two 104-Mbyte disks:

1. **Concatenate the `104_test` file with itself and save the output to another file:**

```
$ cat 104_test 104_test > dual_104_test
```

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

For example, the `dual_104_test` file is shown as follows:

```
    # cat dual_104_test
❶   * /dev/rdsk/c0t3d0s2 partition map
    *
    * Dimensions:
    *     512 bytes/sector
    *      35 sectors/track
    *       6 tracks/cylinder
    *     210 sectors/cylinder
    *    1019 cylinders
    *     974 accessible cylinders
    *
    * Flags:
    *   1: unmountable
    *  10: read-only
    *
    *                          First     Sector    Last
    * Partition  Tag  Flags   Sector    Count    Sector Mount Directory
           0      2    00          0     16170     16169
           1      3    00      16170     28140     44309
           2      5    00          0    204540    204539
           6      4    01      44310    160230    204539
❷   * /dev/rdsk/c0t0d0s2 partition map
    *
    * Dimensions:
    *     512 bytes/sector
    *      35 sectors/track
    *       6 tracks/cylinder
    *     210 sectors/cylinder
    *    1019 cylinders
    *     974 accessible cylinders
    *
    * Flags:
    *   1: unmountable
    *  10: read-only
    *
    *                          First     Sector    Last
    * Partition  Tag  Flags   Sector    Count    Sector Mount Directory
           0      2    00          0     16170     16169
           1      3    00      16170     28140     44309
           2      5    00          0    204540    204539
           6      4    01      44310    160230    204539
```

This is what was done to the `dual_104_test` file:

❶ The first disk device name `was` not changed.

❷ The second disk device name was changed from `/dev/rdsk/c0t3d0s2` to `/dev/rdsk/c0t0d0s2`. This gives each disk a different target.

Task Complete

You have completed creating a multiple disk configuration file.

Using a Site-Specific Installation Program

Through the use of begin and finish scripts, sites with unusual 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 - merges the new version of Solaris software with existing files on the system's disk. It saves as many local modifications as possible.
- Initial - overwrites the system's disk with the new version of Solaris software.

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

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 2.1 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 2.1 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.

Backing Up Your System

Always back up an existing system before installing a new version of Solaris software, even if you choose the upgrade option. 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 *File System Administration*.

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.

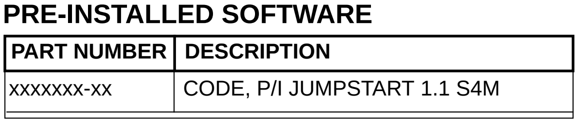
Booting and Installing Solaris: JumpStart 7

This chapter describes how to perform a JumpStart installation.

1 Make sure you have a new system with the JumpStart software.

If the New System Is a	Then
SPARCclassic SPARCstation IPX SPARCstation LX SPARCstation 5 SPARCstation 10 SPARCstation 10 SX SPARCstation 20 SPARCstation Voyager	You can use the JumpStart method of installing the Solaris software described in this chapter.
None of above	You must use the interactive method or custom JumpStart method; go back to “Choose a method for installing Solaris software.” on page 11

The Customer Information sheet, which is attached to the outside of the system’s packing box, will have the following entry if the JumpStart software is installed on the system.



PRE-INSTALLED SOFTWARE

PART NUMBER	DESCRIPTION
xxxxxxx-xx	CODE, P/I JUMPSTART 1.1 S4M

Also, the “JumpStart for SPARC Systems” document will be included with the system’s hardware documentation if the JumpStart software is installed on the system.

2 Make sure you have reviewed the task map for a JumpStart installation (shown in Figure 2-2 on page 13).

3 Make sure the Solaris software that will automatically be installed meets your needs.

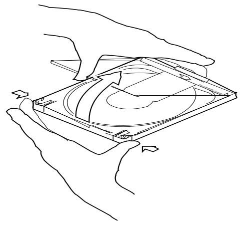
If the System Is a	And the Size of Its Formatted, Internal Disk Is Between	Then the following Solaris software is installed
SPARCstation Voyager	200 - 9999 Mbytes	Core software group, including the following SUNW packages: audio, bnur, bnuu, lpr, lpu, tltk, xilrt, xildg, xiler, xilow, libms, owrqd, admfw, inst, admap, loc, doc, apppr, apppu, pppk
SPARCstation 10 SX SPARCstation 20	320 - 430 Mbytes	Developer software group and the SUNWCsx cluster
SPARCstation 10 SX SPARCstation 20	431 - 9999 Mbytes	Entire distribution software group
SPARCstation 5	N/A	End user software group and the SUNWCtcx cluster
SPARCclassic SPARCstation IPX SPARCstation LX SPARCstation 10	98 - 108 Mbytes	Core software group, including the following SUNW packages: admr, admap, admfw, bcp, esu, inst, ipc, lpr, lpu, too
SPARCclassic SPARCstation IPX SPARCstation LX SPARCstation 10	109 - 9999 Mbytes	End user software group

4 If you are using the system’s local CD-ROM drive to install the Solaris software, prepare the CD-ROM drive.

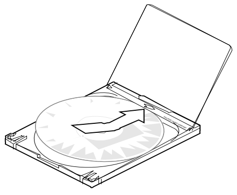
The following instructions cover the most common types of CD-ROM drives. If your CD-ROM drive is not one of the following types, see your hardware manual for instructions.

CD-ROM Drive - Caddy Version

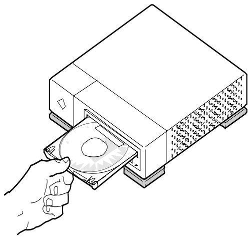
❶ To open the caddy, pinch the corners while lifting the lid at the center lip.



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

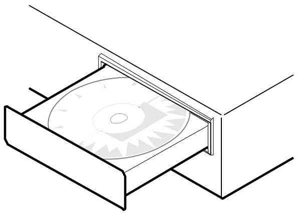


❸ With the arrow on the caddy pointing toward the CD-ROM drive, insert the caddy.



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.

5 Turn on the components in the order recommended in your hardware guide.

If System Begins to Boot	If System Displays the > or `ok` Prompt
You will see messages displayed on the screen indicating hardware and system components are being checked. The booting phase can last for several minutes. **Note**: During the booting phase, OpenWindows is started for localization reasons, or for the graphical user interface; however, OpenWindows is not fully installed until Solaris software is installed.	Your system does not have the JumpStart software. Go back to page 11 and choose another method of installing Solaris software.

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

You're done for awhile; installing the Solaris software can take between 15 minutes and 2 hours. After a successful installation, the system will reboot and you're ready for Step 8.

8 Start the OpenWindows software to display the desktop:

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

Task Complete

For post-installation information, see Chapter 11, "Where to Go After Installing Solaris."

Booting and Installing Solaris: Interactive 8

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 to Solaris 2.4, add or delete software if desired, then start the upgrade.

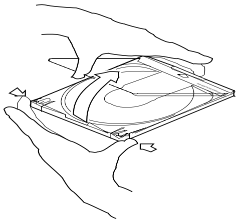
The procedure in this chapter should be done on the system that is being installed.

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

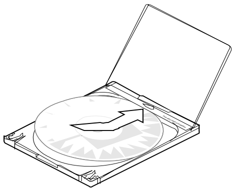
The following instructions cover the most common types of CD drives. If your CD drive is not one of the following types, see your hardware manual for instructions.

CD-ROM Drive - Caddy Version

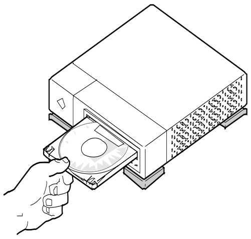
❶ To open the caddy, pinch the corners while lifting the lid at the center lip.



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

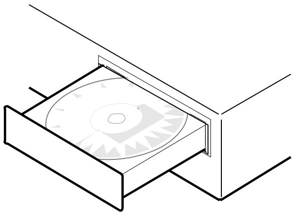


❸ With the arrow on the caddy pointing toward the CD-ROM drive, insert the caddy.



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.

2 Make sure you've reviewed the task map for an interactive installation (Figure 2-3 on page 14).

3 Follow these 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. **Caution:** If the system starts booting, press L1-A or Stop-A. **2)** Go to Step 4.

On	**1)** If the system is running Solaris, enter the following commands:
	`$ su root`
	`# halt`
	2) Go to Step 4.

4 If the screen displays the > prompt instead of the ok prompt, then enter n and press Return.

The screen should now display the `ok` prompt.

Note – This step is not required for Sun-4 systems, because they do not have the `ok` prompt.

5 Boot the system using the appropriate boot command:

If You Are Booting	And The System You Are Booting Is	Then Enter
From a server on the network	Sun-4/3*nn*	`b le()`
	Sun-4/1*nn* Sun-4/2*nn* Sun-4/4*nn*	`b ie()`
	All other Sun systems	`boot net`
From the system's local CD-ROM drive	Sun-4/1*nn* Sun-4/2*nn* Sun-4/3*nn* Sun-4/4*nn*	`b sd(0,30,1)`
	SPARCstation 1 (4/60) SPARCstation 1+ (4/65) SPARCstation SLC (4/20) SPARCstation IPC (4/40)	`boot sd(0,6,2)`
	SPARCengine® 1E	`boot sd(0,6,5)`
	All other Sun systems	`boot cdrom`

6 Wait for booting to complete.

After you type the boot command, the system goes through a booting phase where hardware and system components are checked. This lasts for several minutes. The following screen provides an example of what you should see.

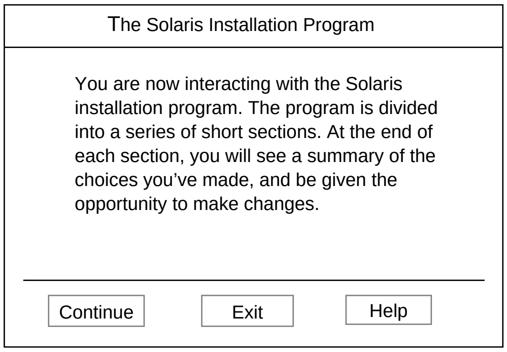
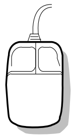
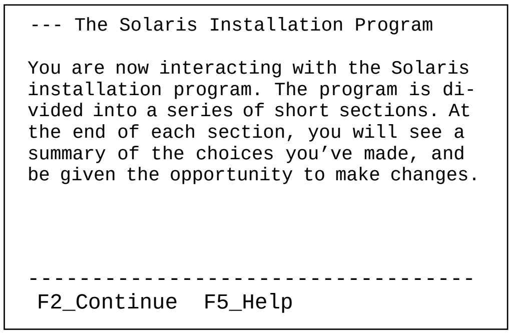
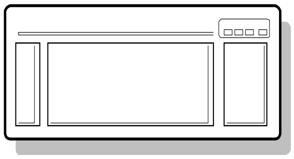
During the booting phase, OpenWindows may be started for localization reasons, or for the graphical user interface; however, OpenWindows is not fully installed until the Solaris software is installed.

```
Type b (boot), c (continue), or n (new command mode)
>n
Type help for more information
ok boot sd(0,6,2)
Booting from: sd(0,6,2)
SunOS Release 5.4 Version [UNIX(R) System V Release]
Copyright (c) 1983-1994, Sun Microsystems, Inc.
WARNING: clock gained 35 days -- CHECK AND RESET THE DATE!
Configuring the /devices directory
Configuring the /dev directory
Starting OpenWindows...
```

7 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, “Work Sheets for the Solaris Installation Program.” There is also a time zone map on page 182 to help you set your system’s time clock. After the Solaris software is successfully installed, you’re ready for Step 8.

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** 

8 If you used the upgrade option, see Chapter 10, “Completing an Upgrade,” for information needed to complete an upgrade.

You must reboot the system after an upgrade.

9 Start the OpenWindows software to display the desktop:

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

Task Complete

For post-installation information, see Chapter 11, “Where to Go After Installing Solaris.”

Booting and Installing Solaris: Custom JumpStart 9

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 Solaris 2.4 software after you boot the system.

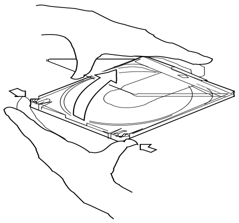
The procedure in this chapter should be done on the system that is being installed.

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

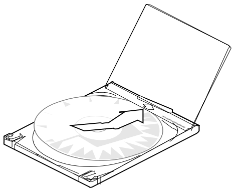
The following instructions cover the most common types of CD drives. If your CD drive is not one of the following types, see your hardware manual for instructions.

CD-ROM Drive - Caddy Version

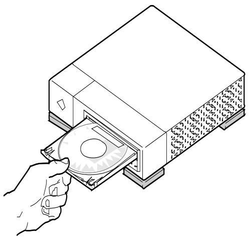
❶ To open the caddy, pinch the corners while lifting the lid at the center lip.



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

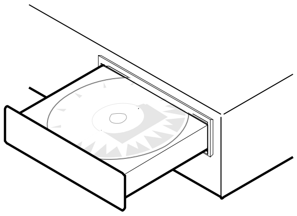


❸ With the arrow on the caddy pointing toward the CD-ROM drive, insert the caddy.



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.

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

3 If you are using a diskette to perform a custom JumpStart installation, insert a diskette that contains a JumpStart directory into the system's diskette drive.

4 Follow the instructions before you boot the system:

If the system has the JumpStart software installed (on new or re-preinstalled systems), turn on the system and go to Step 7.

If The System Is	Then
Off	**1)** Turn on the system components in the order recommended in the hardware guide. **Caution:** If the system starts booting, press L1-A or Stop-A. **2)** Go to Step 5.

On	**1)** If the system is running Solaris, enter the following commands:
	`$ su root`
	`# halt`
	2) Go to Step 5.

5 If the screen displays the > prompt instead of the `ok` prompt, then enter `n` and press Return.

The screen should now display the `ok` prompt.

Note – This step is not required for Sun-4 systems, because they do not have the `ok` prompt.

6 Boot the system using the appropriate boot command:

If You Are Booting	And the System You Are Booting Is	Then Enter[1]
From a server on the network	Sun-4/3*nn*	`b le() - install`
	Sun-4/1*nn* Sun-4/2*nn* Sun-4/4*nn*	`b ie() - install`
	All other Sun systems	`boot net - install`
From the system's local CD-ROM drive	Sun-4/1*nn* Sun-4/2*nn* Sun-4/3*nn* Sun-4/4*nn*	`b sd(0,30,1) - install`
	SPARCstation 1 (4/60) SPARCstation 1+ (4/65) SPARCstation SLC (4/20) SPARCstation IPC (4/40)	`boot sd(0,6,2) - install`
	SPARCengine 1E	`boot sd(0,6,5) - install`
	All other Sun systems	`boot cdrom - install`

[1] A space is required between the minus sign and install.

7 Wait for booting to be completed.

After you type the boot command, the system will go through a booting phase where various hardware and system components are checked. The following screen provides an example of what you should see:

```
Type b (boot), c (continue), or n (new command mode)
>n
Type help for more information
ok boot net - install
Booting from: le(0,0,0) - install
2bc00 hostname: sherlock
domainname: england.com
root server: london
root directory:
/cdrom/solaris_2_4_sparc/s0/export/exec/kvm/sparc.sun4c.Solaris_2.4
SunOS Release 5.4 Version [UNIX(R) System V Release]
Copyright (c) 1983-1994, Sun Microsystems, Inc.
Configuring the /devices directory
Configuring the /dev directory
Searching for JumpStart directory...using london:/jumpstart
Starting OpenWindows...
```

Note – The booting phase will last for a few minutes.

8 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.

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

11 Start the OpenWindows software to display the desktop:

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

Task Complete

For post-installation information, see Chapter 11, “Where to Go After Installing Solaris.”

Completing an Upgrade 10

Overview

This chapter helps you determine if a system was upgraded successfully.

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 file:

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

Table 10-1 provides the entry descriptions of the unpreserved local modifications that may be found in the `upgrade_cleanup` file.

Table 10-1 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.

Where to Go After Installing Solaris 11

Table 11-1 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 2.4 Introduction*.

Note – Some SPARC systems require that you add additional software before Solaris software is completely installed. Check the *SPARC Hardware Platform Guide* to see if your vendor requires any additional software to run Solaris; if yes, see the *Software and AnswerBook Packages Administration Guide* for instructions on adding software.

Table 11-1 Where to Go After Installing Solaris Software

Information Needed	**Manual Title**
Adding and removing Solaris, third-party, or unbundled software	*Software and AnswerBook Packages Administration Guide*
Setting up mail accounts	*User Accounts, Printers, and Mail Administration*
Setting up printers	*User Accounts, Printers, and Mail Administration*
Setting up user accounts	*User Accounts, Printers, and Mail Administration*

Table 11-1 Where to Go After Installing Solaris Software *(Continued)*

Information Needed	**Manual Title**
Performing common administration tasks	*Common Administration Tasks*
Using system administration tools	*Administration Application Reference Manual*
Increasing your system's performance	*Security, Performance, and Accounting Administration*
Adding and Maintaining Peripherals	*Peripherals Administration*
Setting up system security	*Security, Performance, and Accounting Administration*
Administering file systems	*File System Administration*
Installing AnswerBooks	*Software and AnswerBook Packages Administration Guide* **Note:** The End User AnswerBook is the only AnswerBook installed with the Solaris software. The End User AnswerBook is part of the entire distribution software group.

Work Sheets for the Solaris Installation Program A

Purpose of the Work Sheets

The work sheets in this appendix list information that may need to be supplied during the Solaris installation program. There is a work sheet for the initial installation option and the upgrade option.

Uses of the Work Sheets

The work sheets can be used before installing Solaris software, or to record your responses as 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 work sheet 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 work sheet 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

Work Sheet for the Initial Installation Option

The following work sheet 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
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 name service not yet established? Example: Other, DCE	
Domain name	Domain in which the system resides. Example:chandy.West.Arp.COM	
Name server	Specify a name server for the system or have software try to find one?	

Information You May Need When Installing Solaris	Description/Example	Work Space
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, server, or dataless client?	
Diskless clients	Number of diskless clients the server will support. Example: 5 (default) Number of megabytes of swap space to assign each diskless client. Example: 24 Mbytes (default)	
Diskless client architectures	All the kernel architectures for diskless clients that are *different* from the server’s architecture (see Appendix C, “Kernel Architectures).	

Information You May Need When Installing Solaris	Description/Example	Work Space
Dataless clients	Host names and IP addresses of the server(s) where dataless clients will mount their `/usr` and `/usr/kvm` 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	
Software	Which software group to install? • Core • End User System Support • Developer System Support • Entire Distribution • Entire Distribution Plus OEM Support 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	

Information You May Need When Installing Solaris	Description/Example	Work Space
Auto-layout file systems?	Should file systems be laid out on disks automatically or manually? **Note**: Manual layout requires advanced system administration skills.	
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	

Work Sheet for the Upgrade Option

The following work sheet lists the important information you may need to supply when upgrading a system from Solaris 2.1 or later.

Information You May Need When Upgrading Solaris	Description/Example	Work Space
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 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	

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.	

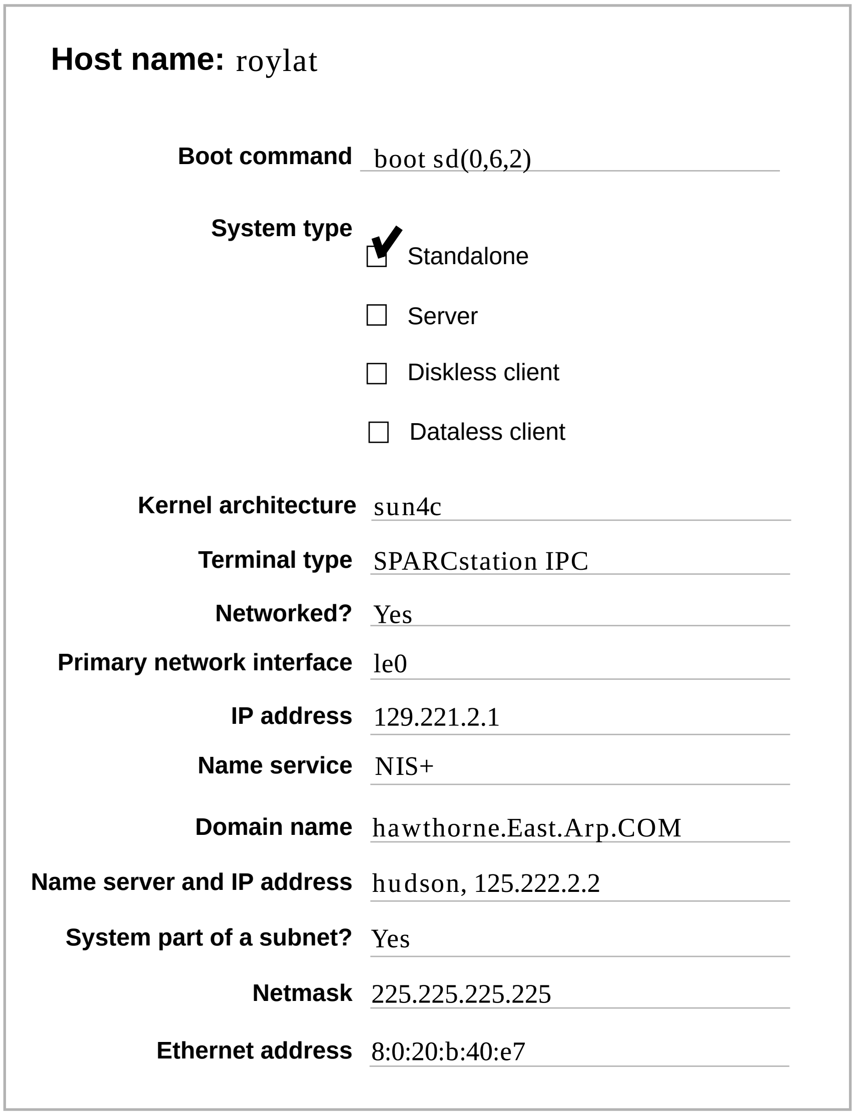
System Identification Label B

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

Boot command boot sd(0,6,2)

System type

☑ Standalone

□ Server

□ Diskless client

□ Dataless client

Kernel architecture sun4c

Terminal type SPARCstation IPC

Networked? Yes

Primary network interface le0

IP address 129.221.2.1

Name service NIS+

Domain name hawthorne.East.Arp.COM

Name server and IP address hudson, 125.222.2.2

System part of a subnet? Yes

Netmask 225.225.225.225

Ethernet address 8:0:20:b:40:e7

B

System Identification Label

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

Host name:

Boot command ______________________

System type

□ Standalone

□ Server

□ Diskless client

□ Dataless client

Kernel architecture ______________________

Terminal type ______________________

Networked? ______________________

Primary network interface ______________________

IP address ______________________

Name service ______________________

Domain name ______________________

Name server ______________________

System part of a subnet? ______________________

Netmask ______________________

Ethernet address ______________________

Kernel Architectures

C

Table C-1 shows the kernel architectures for various hardware platforms. You may need this information when preparing a system for installation of the Solaris software.

Table C-1 Architecture Types

System	Kernel Architecture
x86	i86pc
Sun-4/110 Sun-4/2*xx* Sun-4/3*xx* Sun-4/4*xx*	sun4
SPARCstation 1 SPARCstation 1+ SPARCstation SLC SPARCstation IPC SPARCstation ELC SPARCstation IPX SPARCstation 2	sun4c
SPARCserver 1000 SPARCcenter 2000	sun4d
SPARCengine 1E	sun4e

Table C-1 Architecture Types *(Continued)*

System	Kernel Architecture
SPARCstation 10 SPARCserver 6*xx* SPARCstation LX SPARCclassic SPARCstation 10 SX SPARCstation 20 SPARCstation 5 SPARCstation Voyager	sun4m
Other SPARC systems	Refer to the *SPARC Hardware Platform Guide* for kernel architecture 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.

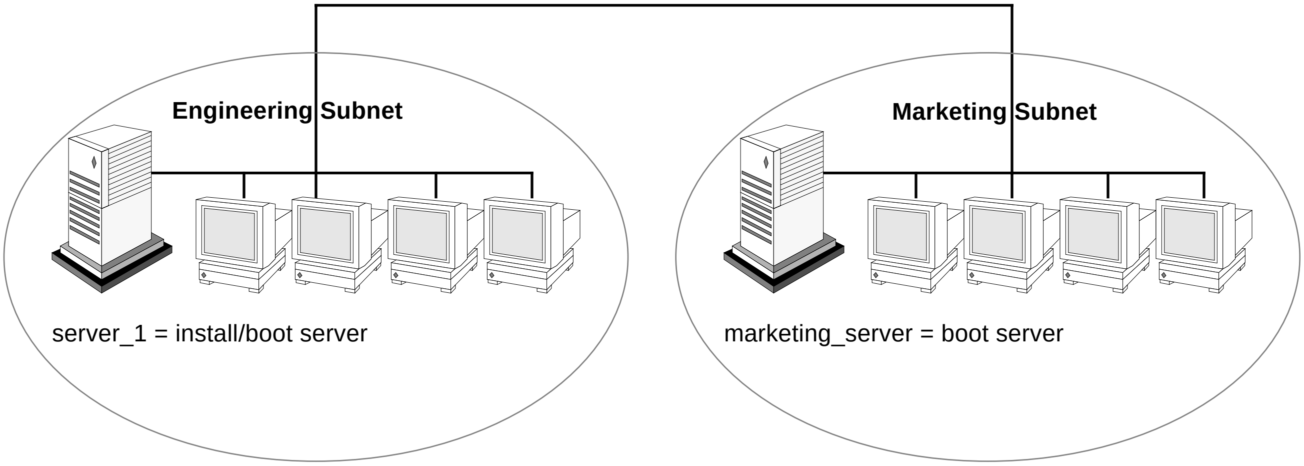

Engineering Subnet
Marketing Subnet
server_1 = install/boot server
marketing_server = boot server

Figure D-1 Sample Site Setup

At this fictitious site:

- The engineering group is on its own subnet. This group uses 32-Mbyte Sun IPX systems for software development.
- The marketing group is on its own subnet. This group uses 16-Mbyte Sun ELC 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`.
- The network information is preconfigured (as described in "Preconfiguring Network Information" on page 45).

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 software 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 software 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 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 98 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 sun4c
# ./add_install_client -c server_1:/jumpstart host_eng2 sun4c
    .
    .
    .
```

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.

`sun4c` Specifies the architecture of the systems that will use `server_1` as an install server. (This is the proper kernel architecture for Sun IPX 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 CD to the marketing server.

```
# cd /cdrom/cdrom0/s0
# ./setup_install_server -b /marketing/boot-dir sun4c
```

In the `setup_install_server` command,

`-b` Specifies that `setup_install_server` will copy the boot information from the Solaris CD to the directory named `/marketing/boot-dir`.

`sun4c` Specifies the architecture of the systems that will use this boot server. (This is the proper kernel architecture for Sun ELC 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 sun4c
# ./add_install_client -s server_1:/export/install -c server_1:/jumpstart host_mkt2 sun4c
    .
    .
    .
```

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.

`sun4c` Specifies the architecture of the systems that will use this boot server. (This is the proper kernel architecture for Sun ELC systems.)

9 Boot the systems and install Solaris software.

The administrator boots the engineering systems by using the following `boot` command at the `ok` (PROM) prompt of each system.

```
ok boot net - install
```

Troubleshooting *E*

This appendix describes problems you may encounter when installing Solaris software. This appendix also suggests possible solutions for some of the problems.

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

The file just loaded does not appear to be executable	*page 175*
Error: Unknown client “host_name”	*page 174*
le0: No carrier - transceiver cable problem	*page 174*
No network boot server. Unable to install the system. See installation instructions.	*page 175*
prom_panic: Could not mount filesystem	*page 175*
Timeout waiting for ARP/RARP packet...	*page 176*
WARNING: clock gained xxx days -- CHECK AND RESET DATE!	*page 176*

Specific Installation Errors

```
Error: Unknown client “host_name”
```

Reason Error Occurred	How to Fix the Problem
The *host_name* argument in the `add_install_client` command must be a host in the name service.	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.

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

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

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

Reason Error Occurred	How to Fix the Problem
This error occurs on a system that you are attempting to install using custom JumpStart. The system could not find a local Solaris CD or an install server on the network.	The system administrator must set up an install server to be used for custom JumpStart installation (see “Creating an Install Server” on page 24). Otherwise, you must do an interactive installation, as documented in Chapter 8, “Booting and Installing Solaris: Interactive.”

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

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

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

Reason Error Occurred	How to Fix the Problem
This message cannot find the proper media for booting.	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. See the table for boot commands on page 136.

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

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

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

Reason Error Occurred	How to Fix the Problem
This is an informational message.	Ignore the message and continue with the installation.

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*

Problem	How to Fix the Problem

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

Problem	How to Fix the Problem
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 `suninstall` or other programs that need more memory. If a disk with a swap partition is available, the system administrator could avoid this problem by using the `swap(1M)` command to add `swap` space to the system before invoking `setup_cd.` When booting from the CD during disaster recovery, it is possible to boot the system in single user mode from the CD. For example: `boot cdrom -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.

Time Zones *F*

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.

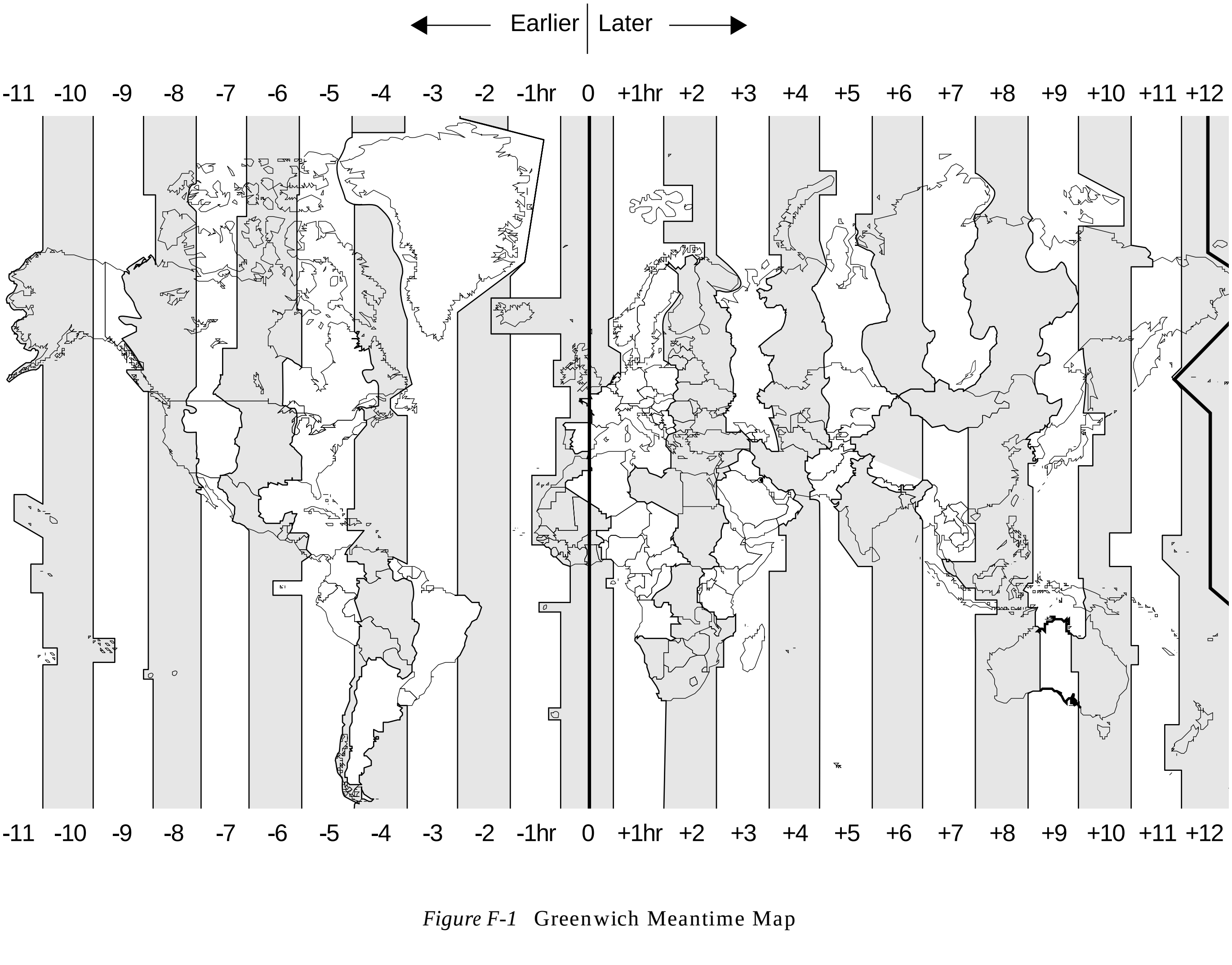


Figure F-1 Greenwich Meantime Map

Glossary G

architecture type

The instruction set that a system uses to run the Solaris software.

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*.

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 a server for its `/usr` and `/usr/kvm` file systems.

derived profile

A profile that is dynamically created by a begin script during a custom JumpStart installation.

developer system support

A software group that contains the End User System Support software group plus the libraries, include files, man pages, and programming tools for developing software.

disk configuration file

A file that represents a structure of a disk (for example, bytes/sector, flags, slices). Disk configuration files enable you to use `pfinstall` from a single system to test profiles on different sized disks.

diskless client

A networked system that does not have its own disk, so it relies completely on a 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 a file server.

domain

A part of the Internet naming hierarchy. It represents a group of systems on a local network that share administrative files.

domain name

The identification of a group of systems on a local network. A domain name consists of a sequence of component names separated by periods (for example: tundra.mpk.ca.us). As you read a domain name from left to right, the component names identify more general (and usually remote) areas of administrative authority.

end user system support

A software group that contains the core software group plus the recommended software for an end user, including OpenWindows and DeskSet software.

entire distribution

A software group that contains the entire Solaris release.

entire distribution plus OEM support

A software group that contains the entire Solaris release, plus additional hardware support for OEMs (SPARC only). This software group is recommended when installing Solaris software on servers.

/etc

A directory that contains critical system configuration files and maintenance commands.

/export

A file system on a 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 a file 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.

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).

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.

JumpStart installation

A type of installation in which the Solaris software is automatically installed on a system by using factory-installed JumpStart software.

kernel architecture

The type of kernel that must be run on a particular system because of its set of hardware components.

locale

A specific language associated with a region or territory.

media server

See *install server.*

mount

The process of making a remote or local file system accessible by executing the `mount` command. To mount a file system, you need a mount point on the local system and the name of the file system to be mounted (for example, `/usr`).

mount point

A directory on a system where you can mount a file system that exists on the local or a remote system.

name server

A server that provides a name service to systems on a network.

name service

A distributed network database that contains key system information about all the systems on a network, so the systems can communicate with each other. With a name service, the system information can be maintained, managed, and accessed on a network-wide basis. Sun supports the following name services: NIS (formerly YP) and NIS+. Without a name service, each system has to maintain its own copy of the system information (in the local `/etc` files).

network installation

A way to install software over the network—from a system with a CD-ROM drive to a system without a CD-ROM drive. Network installations require a *name server* and an *install server.*

networked systems

A group of systems (called hosts) connected through hardware and software, so they can communicate and share information; referred to as a local area network (LAN). One or more servers are usually needed when systems are networked.

NIS

Network Information Service. A type of name service that is standard on SunOS 3.x, 4.x, and Solaris 1.x systems.

NIS+

Network Information Service, Plus. The replacement for NIS that provides automatic information updating and adds security features such as authorization and authentication. NIS+ is the standard on Solaris 2.x systems.

non-networked systems

Systems that are not connected to a network or do not rely on other systems.

/opt

A file system that contains the mount points for third-party and unbundled software.

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.

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*.

rules.ok file

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.

server

A system that provides services to systems on a network. To serve diskless clients, a 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, a server must provide the `/usr` and `/usr/kvm` file systems.

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, entire distribution, or entire distribution plus OEM support (SPARC only).

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 a 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. The valid system types are standalone system, dataless client, and diskless client.

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 (SPARC only) if at least one disk on your system has a Solaris 2.x root file system on slice 0. An upgrade 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.

Index

A

add_install_client
- adding system to boot server 39
- adding system to install server 37
- command syntax 37, 39

adding diskless client 41

adding systems to boot server 39

adding systems to install server 37

administration tool
- adding diskless clients 41

alternative installation program 121

arch rule keyword 98

auto-install, *See* custom JumpStart

automating installation
- *See* custom JumpStart

B

begin script
- creating derived profiles with 108
- example 109
- required permissions 108
- use of 108

boot commands
- custom JumpStart 142
- diskless clients 44
- interactive installation 136

boot server
- adding systems to 39
- defined 18

booting from wrong server 177

C

check script 103, 104

client_arch profile keyword 81

client_swap profile keyword 81

cluster profile keyword 82

common installation problems 173

common problems
- booting from wrong server 177
- I/O interrupt messages 23
- OpenWindows doesn’t start 178
- system does not boot 177

copying boot software with setup_install_server 171

copying Solaris CD to disk 26

creating a rules file 93

creating profiles 76

custom JumpStart
- boot commands 142
- compared to interactive and JumpStart 11
- creating a rules file 93
- defined 55

- from a server 71
- overview of set-up tasks 57
- process described 58, 59, 60
- rules file syntax 93
- sharing the JumpStart directory 71, 167

D

dataless client
- adding for network installation 35
- adding x86 support on server 30
- defined 7

derived profile, defined 108

dfstab file, sharing the JumpStart directory 71, 167

disaster recovery procedure 179

disk configuration files 117

disk requirements for copying Solaris CD to disk 26, 29

disk space requirements 9

diskette
- formatting 63, 64, 68
- mounting 69
- using for custom JumpStart 61

diskless client
- adding for network installation 41
- adding x86 support on server 30
- booting 44
- defined 7, 41

disksize rule keyword 98

DNS, note about use 32

domainname rule keyword 98

dontuse profile keyword 83

E

error messages 173

F

fdisk profile keyword 83, 84

file server, defined 18

filesys profile keyword 85, 86, 87

finish script
- adding files with 111
- customizing environment with 112
- examples 112
- required permissions 110
- setting root password with 112
- use of 110
- using set_root_pw file 113

floppy, *See* diskette

formatting a diskette 63, 64, 68

H

host manager
- adding diskless clients with 41
- adding standalone, server, or dataless clients with 32

hostname rule keyword 99

I

I/O interrupt messages, recovery 23

initial installation, defined 123

inserting Solaris CD in drive 130, 134, 140

install server, defined 18

install_type profile keyword 87

installation methods 11

interactive installation 136
- compared to JumpStart and custom JumpStart 11
- decision flowchart 14

J

JumpStart
- Solaris software installed during 129
- systems supporting 128

JumpStart directory
- defined 71
- required permissions 61, 71
- sharing from a server 71

JumpStart installation
- compared to interactive and custom JumpStart 11
- decision flowchart 13

K

karch rule keyword 99
kernel architectures 163

L

languages, valid values 52
locale
  defined 48
  updating NIS+ table 52
  valid values 52
locale profile keyword 88
locale.org_dir, creating NIS+ table 51
localization, languages supported 52

M

matching order in rules file 96
memsize rule keyword 99
methods for installing Solaris software 11
model name rule keyword 100
mounting a diskette 69
mounting Solaris CD manually 25, 28, 72

N

name server, defined 18
naming profiles 77
network installation
  adding diskless clients 41
  adding standalone systems 31
  defined 17
  preconfiguring date and time 45
  preconfiguring geographic region 45
  preconfiguring network information 20
  preconfiguring subnet mask 45
  recommendations 24
  requirements 18
network rule keyword 100
networked systems, defined 7
NIS
  preconfiguring default locale 48
  pushing maps 47
NIS+
  preconfiguring default locale 51
  preconfiguring network information 47
nispopulate, preconfiguring network information 47
nistbladm
  creating locale.org_dir table 51
  updating locale.org_dir table 52
non-networked systems, defined 7
num_clients profile keyword 88

O

OpenWindows, starting 138, 144
output files from upgrade installation 145
overview of custom JumpStart 57

P

package profile keyword 89
partitioning profile keyword 90
PASSWD variable 113
pfinstall
  command syntax 114
  examples 115, 116
  testing profiles with 115
  use of 114
  using disk configuration file with 117
populating NIS+ tables 47
preconfiguring date and time for network installation 45
preconfiguring geographic region for network installation 45
preconfiguring network information 20
preconfiguring subnet mask for network installation 45
preserving symbolic links for upgrade 125
profile
  keywords defined 81
  values defined 81
profile keywords

- client_arch 81
- client_swap 81
- cluster 82
- dontuse 83
- fdisk 83, 84
- filesys 85, 86, 87
- install_type 87
- locale 88
- num_clients 88
- package 89
- partitioning 90
- system_type 90
- usedisk 90

profiles
- creating 76
- examples 78
- information about editing 77, 93
- naming 77
- requirements for 76
- testing 115

prtvtoc
- command syntax 117
- creating disk configuration files 117
- examples 118

R

recovery procedure 179

requirements for a profile 76

resetting display after I/O interrupts 23

resolv.conf, note about use 32

restoring the root file system 179

root password, setting with finish script 112

rootdisk 102

rule keywords
- any 98
- arch 98
- disksize 98
- domainname 98
- hostname 99
- installed 99
- karch 99
- memsize 99
- model 100
- network 100
- totaldisk 101

rules file
- comment fields 97
- creating 93
- defined 92
- example rules 96
- matching order 96
- name requirements 97
- required permissions 97
- rule keywords 98
- rule matching order 97
- rule values 98
- rule wrap 97
- syntax defined 93
- validating 103

S

server
- adding other architecture support on 30
- defined 7

setting system clock 181

setup_install_server
- copying Solaris to disk 26
- setting up a boot server 29, 171

sharing JumpStart directory 167

SI_CONFIG_DIR variable 111

SI_PROFILE variable 109

SI_SYS_STATE variable 113

Solaris CD, inserting in drive 130, 134, 140

Solaris software
- character-based interface 137
- core system support 9
- developer system support 9
- disk space requirements 9
- end user system support 9
- entire distribution 9
- graphical interface 137
- groups of software to install 9
- methods for installing 11

standalone system

- adding for network installation 31
- defined 7

starting OpenWindows 138, 144

starting tftpd 177

SUNWCall software group 9, 82

SUNWCprog software group 9, 82

SUNWCreq software group 9, 82

SUNWCuser software group 9, 82

SUNWCXall software group 9, 82

system boot problem 177

system types 7

system types and kernel architectures 163

system_type profile keyword 90

systems with JumpStart software 128

T

testing profiles on multiple disks 119

testing profiles with disk configuration files 117

testing profiles with pfinstall 114

tftpd, starting 177

time zones supported by Solaris 181

totaldisk rule keyword 101

U

upgrade installation
- continuing after a failed installation 179
- controlling file systems mounted 126
- defined 123
- determining availability 124
- output files 145
- preserving local modifications 125
- preserving symbolic links 125

upgrade_cleanup file 145

upgrade_log file 145

usedisk profile keyword 90

V

validating a rules file 103

validating profiles 115

variables
- PASSWD 113
- SI_CONFIG_DIR 111
- SI_PROFILE 109
- SI_SYS_STATE 113

W

wrapping lines in rules file 97